

Ontwikkelen van een nieuw Support Feedback Systeem. Mailplus B.V.

Auteur: Jos Verhaar (20052930)
Bedrijf: Mailplus B.V.
Opdrachtgever: Paul Nederlof
Bedrijfsmentor: Niels Noorlander
Examinatoren: Arie Toet & Gerard Mijnares
Datum: 9 januari 2017
Instelling: Haagse Hogeschool
Stageperiode: mei 2016-januari 2017

Voorwoord

Bij deze afstudeeropdracht probeer ik tegemoet te komen aan de wens van meerdere MailPlus klanten om ook gebruik te mogen maken van het feedback systeem die de afdeling support van Mailplus B.V. gebruikt. Deze wens van onze klanten viel samen met mijn afstuderen aan de Haagse Hogeschool. Zodoende valt de wens van mijn opdrachtgever en het afstudeeropdracht vanuit Haagse Hogeschool samen.

Deze opdracht is met name interessant voor Mailplus B.V. Naar aanleiding van dit afstudeeronderzoek kan dit nieuwe product geïmplementeerd worden en ook verkocht worden aan klanten van Mailplus B.V.

Graag neem ik deze mogelijkheid om vanaf deze plaats mijn bedrijfsmentor Niels Noorlander en mijn afstudeerbegeleider dhr. Toet te bedanken voor hun steun en feedback die ik gedurende de uitvoering van deze afstudeeropdracht heb mogen ontvangen.

Speciale dank gaat uit naar mijn vrouw voor alle steun die zij mij gegeven heeft en het vele geduld dat zij met mij heeft gehad.

Leiden, 9 januari 2017

Jos Verhaar

Inhoudsopgave

Voorwoord.....	3
1. Inleiding.....	8
1.1 Doel van het document	8
1.2 Leeswijzer	8
2. Het bedrijf	9
2.1 Organisatiebeschrijving.....	9
2.2 Jouw werkomgeving.....	10
2.3 Betrokkenen afstudeeropdracht.....	10
3. Opdrachtdefinitie	11
3.1 Opdrachtoomschrijving.....	11
3.1.1 Wat is het Support Feedback Systeem?.....	11
3.1.2 Huidige implementatie.....	11
3.1.3 Wat wil MailPlus B.V. bereiken?.....	15
3.1.4 Waarom voldoet de huidige implementatie niet?	15
3.2 Doelstelling.....	15
3.3 Gewenste resultaat.....	16
3.4 Op te leveren producten	16
4. Methoden en technieken	17
4.1 Proces/ Algemeen	17
4.1.1 Scrum	17
4.1.2 Implementatie Scrum.....	17
4.1.3 Concurrentieonderzoek.....	18
4.2 Ontwerpen	18
4.2.1 Requirements.....	18
4.2.2 MoSCoW.....	19
4.2.3 Database.....	19
4.2.4 3-Lagen model.....	19
4.3 Implementatie	19
4.3.1 Spring Web MVC Framework	19
4.3.2 Jersey JAX-RS.....	20
4.3.3 Jackson.....	20
4.3.4 OAuth.....	20
4.4 Testen	21
4.4.1 TestGoal	21
4.4.2 JUnit	21
4.4.3 Hamcrest.....	22
4.4.4 Mockito.....	22
4.4.5 HSQLDB.....	22
5. Sprint 0	23
5.1 Inlezen en inventarisatie huidige implementatie.....	23
5.2 Concurrentieonderzoek.....	25
5.3 Requirements	26
5.3.1 Requirements huidige implementatie.....	26
5.3.2 Input belanghebbende.....	27
5.3.3 Prioritering requirements.....	27
5.3.4 Database ontwerpkeuze: Meerdere gebruikers	30

5.4	Keuze en motivatie: Ontwerpkeuze om meerdere gebruikers te bedienen	31
5.5	Reflectie sprint 0	31
6.	Sprint 1	33
6.1	Opzet ontwikkelomgeving.....	33
6.2	Uitwerken requirements UR12 en UR32.....	33
6.3	Sprint retrospective.....	37
6.4	Reflectie sprint 1	37
7.	Sprint 2	38
7.1	JUnit, Hamcrest en HSQLDB.....	39
7.1.1	UR24 <i>Via de Rest-API is het mogelijk om de statistieken op te vragen van het Feedback Systeem</i>	40
7.2	Herschrijven van FeedbackController naar IncidentController	40
7.3	Sprint retrospective.....	40
7.4	Reflectie sprint 2	41
8.	Sprint 3	42
8.1	Uitwerking UR-38.....	42
8.1.1	Database.....	42
8.1.2	Codebase.....	43
8.1.2.1	Dependency Injection binnen Spring	43
8.1.3	Nieuwe controller	44
8.1.4	Defence in depth.....	45
8.2	Sprint retrospective.....	46
8.3	Reflectie sprint 3	46
9.	Sprint 4	47
9.1	UR9.....	47
	<i>Het dient mogelijk te zijn om behandelingen (operators) te beheren (toevoegen, bewerken en verwijderen) via het SFS.</i>	47
9.2	UR27	49
	<i>De dag-, maand- en jaarcijfers worden absoluut, in percentage en in (afgeronde) percentagepunt getoond</i>	49
9.3	UR29.....	49
	<i>De mogelijkheid tot het tonen van grafieken met daarin het percentage good/ok/bad in vergelijking met de vorige periode</i>	49
9.4	UR30	51
	<i>Bij de beoordelingen op het dagoverzicht heb je de mogelijkheid tot het zien bij welk feedback verzoek deze hoort (naam beoordelaar, bedrijfsnaam en callnummer)</i>	51
9.5	Aanpassingen buiten de sprint backlog.....	52
9.5.1	Introductie Universally Unique Identifier.....	52
9.5.2	Toevoeging kwartalen als timeframe.....	52
9.5.3	E-mail validatie naar validator klasse	52
9.6	Sprint retrospective.....	52
9.7	Reflectie sprint 4	53
10.	Sprint 5	54
10.1	Opstellen testplan	54
10.1.1	testbasis.....	54
10.1.2	teststrategie.....	54
10.1.3	testrisicoanalyse	54
10.1.4	bepalen testtechnieken.....	55
10.1.5	bepalen logische testgevallen	56
10.1.6	bepalen fysieke testgevallen.....	56
10.1.7	Incident-Update en Incident-Toevoegen	56

10.2 Geautomatiseerd testen in JUnit.....	56
10.2.1 JUnit.....	56
10.2.2 Mockito	57
10.2.3 Hamcrest.....	58
10.2.4 HSQLDB.....	58
10.3. Uitwerking Operator-Opvragen.....	59
10.3.1 Logische testgevallen.....	59
10.3.2 Fysieke testgevallen	60
10.3.3 Implementatie fysieke testgevallen.....	65
10.3.4 Uitvoering van de testgevallen.....	66
10.4. Sprint retrospective	68
10.5. Reflectie sprint 5	69
11. Procesevaluatie.....	70
11.1 Proces.....	70
11.2 Methoden en Technieken	70
11.3 Begeleiding	70
11.4 Leerpunten.....	71
12. Product evaluatie	72
12.1 Support Feedback Systeem v2	72
12.2 OAuth 1.0a.....	72
12.3 Aanbevelingen.....	72
12.3.1 Frontend.....	73
12.3.2 Backend.....	73
12.3.3 TopDesk integratie	73
13. Beroepstaken.....	74
13.1 Beroepstaak 2.2	74
<i>Ontwerpen, bouwen en bevragen van een database niveau: 4 (zelfstandig, complex).....</i>	<i>74</i>
13.2 Beroepstaak 3.2	74
<i>Ontwerpen systeemdeel niveau: 4 (zelfstandig, complex)</i>	<i>74</i>
13.3 Beroepstaak 3.3	74
<i>Bouwen applicatie niveau: 4 (zelfstandig, complex).....</i>	<i>74</i>
13.4 Beroepstaak 3.4	75
<i>Initiëren en plannen van het testproces: 3 (zelfstandig, lastig)</i>	<i>75</i>
13.5 Beroepstaak 3.5.....	75
<i>Uitvoeren van en rapporteren over het testprocesniveau: 3 (zelfstandig, lastig).....</i>	<i>75</i>
Literatuurlijst.....	76
Verklarende woordenlijst	78
Bijlage I : Klassendiagram huidige implementatie	81
Bijlage II: Sequentiediagrammen key-features huidige implementatie	87
Bijlage III: Huidige opzet	93
RRM huidige database.....	93
Bijlage IV: Het concurrentieonderzoek.....	94
Wat is het doel van het onderzoek?	94
Waarom wordt dit onderzocht?.....	94
Hoe wordt dit onderzocht?	94
Hoe worden concurrenten gevonden?.....	95
Gebruikte zoekwoorden.....	95
Gevonden artikelen en bedrijven	95

Verfijnen van concurrenten.....	96
Analyse methode.....	96
Bevindingen.....	100
Conclusie.....	100
Bijlage V: Requirements.....	102
Business Requirements.....	102
User requirements gebaseerd op geïdentificeerde belanghebbende.....	102
Technische beperkingen.....	107
User requirements SMART gemaakt.....	107
Geprioriteerde Requirements.....	110
Bijlage VI: Input voor requirements	115
Inbreng Paul Nederlof.....	115
Eigen wensen:.....	115
Inbreng Sander van 't Hof.....	116
Inbreng Jos verhaar.....	117
Bijlage VII: Use Case beschrijvingen	118
01 Use Case: Verzend feedback verzoek via e-mail.....	118
01.1 Use Case: Zet mail om naar incident.....	118
01.2 Use Case: Verwerk incident.....	118
01.2.1 Use Case: Valideer incident.....	119
01.2.2 Use Case: Creëer nieuw feedback record.....	120
01.2.3 Use Case: Verzend feedback mail	120
02 Use Case: Verzend feedback verzoek via Rest-API call	121
02.1 Use Case: Valideer Rest-API request.....	121
Bijlage VIII: Sequentie Diagrammen huidige implementatie.....	123
Bijlage IX: Testplan	131
Bijlage X: Rest-API documentatie.....	255
Bijlage XI: Afstudeerplan.....	271
Bijlage XII: Beoordeling.....	276
Bijlage XIII: Plan van aanpak	280

1. Inleiding

1.1 Doel van het document

Dit verslag is tijdens het uitvoeren van de afstudeeropdracht geschreven. Het verslag is geschreven met als doel om geïnteresseerden te informeren over het proces van de afstudeeropdracht.

In dit verslag wordt het proces beschreven dat doorlopen is om het Support Feedback Systeem (hierna: SFS) te ontwikkelen. Het gehele proces en de bijbehorende keuzes die gedurende de ontwikkeling zijn gemaakt worden beschreven in dit verslag.

1.2 Leeswijzer

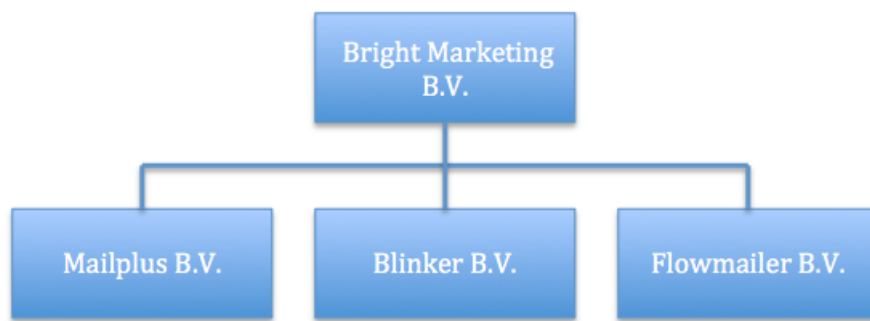
Door het gehele document heen worden termen gebruikt die regelmatig worden gebruikt binnen het bedrijf of van technische aard zijn. Om de leesbaarheid van het verslag te waarborgen is er een begrippenlijst opgesteld om deze termen te verduidelijken. Hier wordt dan in de tekst naar verwezen. Voor de rest wordt aangenomen dat de lezer basiskennis bezit van software ontwikkeling.

De afstudeeropdracht is uitgevoerd bij Mailplus B.V. In hoofdstuk twee wordt een organisatiebeschrijving gegeven. Alsmede een omschrijving van de werkomgeving en tot slot een overzicht van alle betrokkenen bij deze afstudeeropdracht. Vervolgens wordt in hoofdstuk drie een uitgebreide opdrachtdefinitie gegeven. In hoofdstuk vier worden de methoden en technieken, die tijdens het onderzoek zijn gebruikt, beschreven. Na deze hoofdstukken volgen de sprint hoofdstukken. In elk hoofdstuk wordt per sprint beschreven wat er gedaan is, hoe het gedaan is en welke keuzes zijn gemaakt. Aan het einde van elke sprint wordt er gereflecteerd op het proces. Tot slot zijn in de laatste hoofdstukken de procesevaluatie en productevaluatie opgenomen. Aan het einde van het document zijn de bijlagen opgenomen.

2. Het bedrijf

2.1 Organisatiebeschrijving

Mailplus B.V. is een E-mail Service Provider (hierna: ESP). Het bedrijf levert een e-mail marketingtool met de naam MailPlus als Software As A Service (SaaS) oplossing. Dit betekent dat Mailplus B.V. de hard- en software levert en beheert die nodig is om MailPlus te gebruiken. Gebruikers hoeven hiervoor niets te installeren en kunnen gebruik maken van MailPlus door via een webbrowser in te loggen met een gebruikersnaam en wachtwoord. Mailplus bestaat sinds 2004 en is een onderdeel van Bright Marketing Holding. Hieronder vallen eveneens Blinker B.V. en Flowmailer B.V. Binnen de Bright Marketing Holding zijn ongeveer 80 mensen werkzaam, waarvan 36 bij Mailplus B.V. In onderstaand organogram is de onderlinge verhouding inzichtelijk gemaakt van de verschillende B.V.'s.

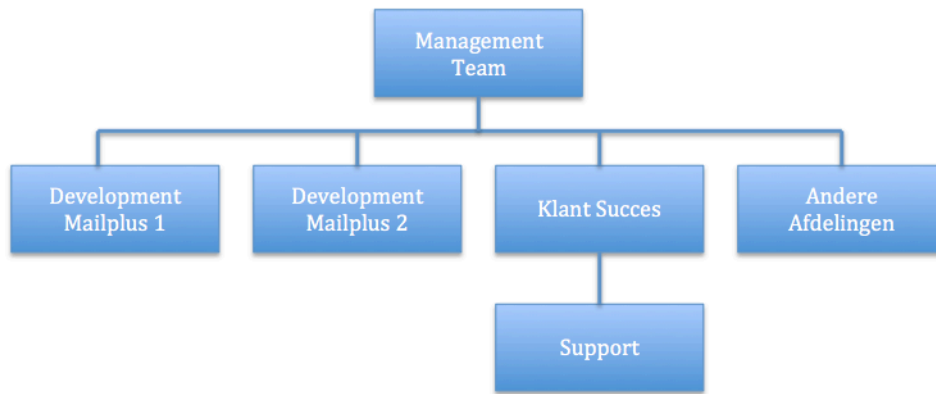


Figuur 1: Organogram Bright Marketing Holding

De afdeling support behandelt jaarlijks 20.000 support aanvragen van gebruikers. Ook collega's vallen onder de gebruikers. De afdeling support bestaat uit 8 personen. Een deel hiervan werkt parttime. De support aanvragen zijn verschillend van aard. Het kunnen eenvoudige vragen zijn over bijvoorbeeld e-mail templates, of meer ingewikkelde vragen over onze koppelingen. Hierbij kun je denken aan vragen over de werking en integratie van de REST-API en/of SOAP-API van MailPlus. Een support aanvraag doorloopt altijd een standaard proces. De vraag komt bij de afdeling binnen via de telefoon en/of via de e-mail. Deze vragen worden geregistreerd in TopDesk en behandeld. Nadat het antwoord is teruggekoppeld en verwerkt in TopDesk, wordt er een feedback mail verzonden naar de aanvrager.

De opdracht wordt uitgevoerd bij één van de twee product development teams. Deze teams bevinden zich direct onder het management team. Elk team is opgebouwd uit software developers en product managers. De product managers brengen klantwensen in kaart en beschrijven nieuwe features die vanuit de business units worden aangedragen. In samenwerking met de developers worden die wensen en features uitgewerkt en geïmplementeerd. Op dit moment werkt het team waar de opdracht uitgevoerd zal worden met name aan het implementeren van klantwensen. Het andere team richt zich op integraties tussen MailPlus en CRM/Webshop systemen. Beide teams werken volgens de Agile ontwikkelmethode Scrum.

In onderstaand organogram (figuur 2) is zichtbaar gemaakt hoe de support afdeling en beide development afdelingen zich verhouden binnen Mailplus B.V.



Figuur 2: Organogram Support ten opzichte van de Development teams.

2.2 Jouw werkomgeving

Het primaire doel van het MailPlus pakket is het versturen van marketing uitingen via e-mail. Hierbij is te denken aan nieuwsbrieven en serviceberichten. De software biedt verder de mogelijkheid tot het inrichten van uitgebreide campagnes. Bij deze campagnes kun je denken aan verlatenwinkelwagen campagne, verjaardag campagne, profiel verrijking campagne, enzovoort. Verder is het mogelijk om pagina's aan te maken waarop je formulieren, survey's, polls, en dergelijke kunt plaatsen. Mailplus B.V. verzorgt hiervan de verkoop, het onderhoud en uitbreiden van de software, het operationeel houden van de software en het geven van support aan de gebruikers. In de hoedanigheid van ESP zijn alle servers in eigenbeheer, maar extern gehuisvest.

2.3 Betrokkenen afstudeeropdracht

Bij de afstudeeropdracht zijn een aantal medewerkers van Mailplus B.V. betrokken. In dit document zal worden gerefereerd naar de rollen.

Naam	Email	Rol
Jos Verhaar	josverhaar@gmail.com jos@mailplus.nl	Afstudeerder
Paul Nederlof	paulne@mailplus.nl	Opdrachtgever
Niels Noorlander	niels@mailplus.nl	Bedrijfsmentor en product owner

Tabel 1: Betrokkenen afstudeeropdracht

3. Opdrachtdefinitie

3.1 Opdrachtoomschrijving

3.1.1 Wat is het Support Feedback Systeem?

Het SFS is een geautomatiseerd systeem dat intern is ontwikkeld. Het doel van het systeem is het doorlopend meten van de klanttevredenheid. Het SFS wordt alleen intern gebruikt door de supportafdeling van Mailplus B.V. en haar gebruikers. Klanten geven feedback over de hulp die geleverd is door een support medewerker. De medewerkers van support worden beoordeeld op hun functioneren naar aanleiding van de statistieken die uit dit systeem worden gehaald.

3.1.2 Huidige implementatie

De huidige implementatie van het SFS is web based. De applicatie is geschreven in Java met toevoeging van verschillende modules. Deze modules worden verder beschreven in hoofdstuk 4. Het SFS bestaat momenteel uit een backend waarin het mogelijk is om feedback records (incidenten) inclusief beoordeling op te slaan en waarmee beperkte statistieken kunnen worden opgevraagd over de tevredenheid van de beoordelaars. Naast de backend is er ook een front-end. De front-end communiceert naar de beoordelaar de gegeven beoordeling en geeft de mogelijkheid tot het wijzigen van deze beoordeling en/of het geven van een toelichting bij de gegeven beoordeling.

Wanneer een support medewerker de vraag van een gebruiker heeft afgehandeld, wordt er door TopDesk een e-mail gegenereerd met een JSON-object. Deze e-mail is de trigger voor het SFS.

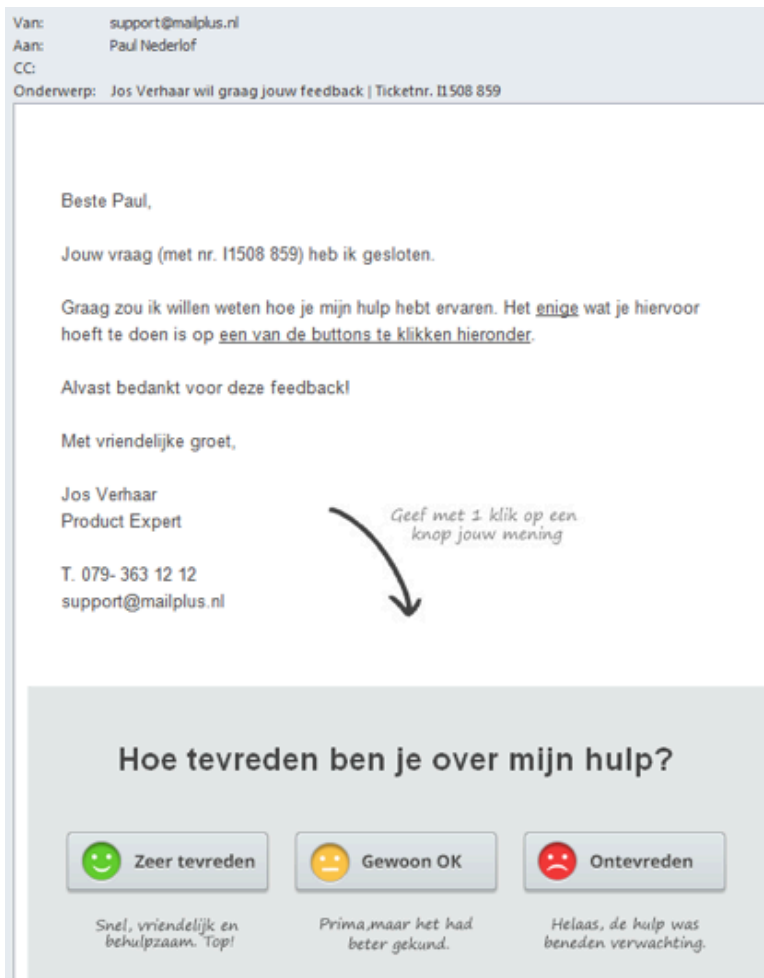


The screenshot shows an email client window titled "E-mail versturen voor I1508 964" with a subtitle "Feedback support". The interface includes tabs for "Algemeen", "Extra", and "Bijlagen". Under the "Algemeen" tab, the "Adres" section shows the sender as "niels@office.blinker.nl" and the recipient as "topdesk@services.mailplus.nl". There is a checkbox for "Kopie naar afzender". The "Bericht" section shows the subject as "Feedback support" and a checkbox for "HTML e-mail". The body of the email contains a JSON object:

```
{
  "id": "I1508 964",
  "requestor": {
    "firstname": "Paul",
    "email": "paulne@mailplus.nl",
    "lastname": "Nederlof",
    "operator": {
      "account": "Medewerkers",
      "lastname": "van Delft",
      "number": "8",
      "email": "ralph@mailplus.nl"
    }
  }
}
```

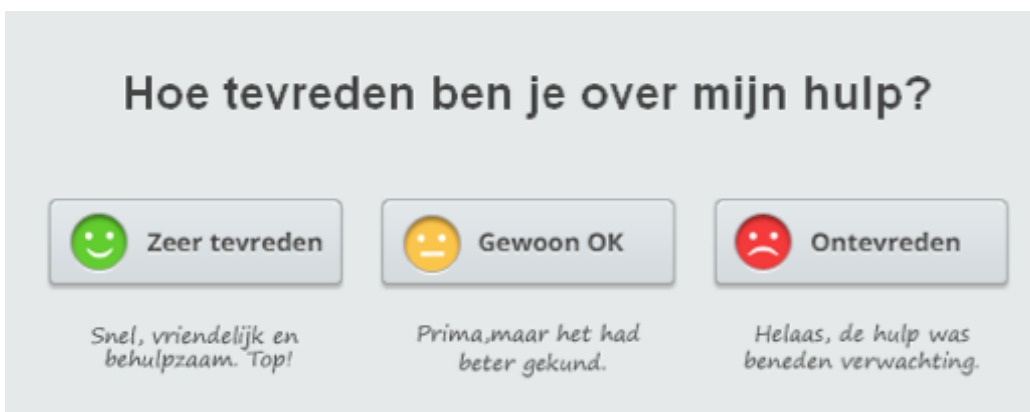
Afbeelding 1: Mail met JSON-object dat vanuit TopDesk wordt gestuurd

Het SFS leest de e-mail in en verwerkt deze tot een andere e-mail, het feedbackverzoek, welke naar de gebruiker wordt verstuurd.



Afbeelding 2: Voorbeeld van de feedbackmail

Vanuit het feedbackverzoek kan de gebruiker zijn waardering geven. Het enige dat de gebruiker hiervoor hoeft te doen is te klikken op een van de smiley-iconen onder in de mail.



Afbeelding 3: Close-up smiley's in de feedbackmail

Na het klikken heeft de gebruiker zijn waardering gegeven en wordt de gebruiker naar een dankpagina geleid. Op deze pagina is er de mogelijkheid tot het geven van een schriftelijke reactie.



Fijn dat je zeer tevreden bent.

[Beoordeling wijzigen](#)

Geef een toelichting op je beoordeling:

[Toelichting opslaan](#)

Afbeelding 4: Landingspagina met mogelijkheid tot geven toelichting

Een support medewerker verstuurt, na het sluiten van een support aanvraag, een feedback e-mail. De klant kan via deze mail de geleverde support beoordelen. Feedback kan gegeven worden door op één van de drie smiley-iconen te klikken. Deze smiley-iconen staan ieder voor een andere gradatie in tevredenheid. 'Ontevreden' (rood) staat voor de meest negatieve beoordeling, 'gewoon ok' (geel) staat voor voldoende support en 'zeer tevreden' (groen) staat voor zeer goede support. Desgewenst is er de mogelijkheid tot het toevoegen van een schriftelijke reactie. De gegeven feedback wordt via een scherm getoond op de afdeling. De huidige functionaliteiten van het SFS omvatten het genereren van unieke links voor het kunnen geven van feedback. Daarnaast het genereren van een e-mail aan de klant met een vaste indeling en opmaak, het opslaan van ontvangen feedback en het opvragen van beperkte statistieken uit het systeem.

Het huidige SFS maakt onderscheidt tussen de volgende statistieken. Customer Satisfaction, Live Scores, Operator Scores en Remarks. De customer satisfaction geeft de totale tevredenheid van de klanten.

Score klanttevredenheid



Aantal feedback verzonden: 12

Aantal gereageerd: 7

Afbeelding 5: Customer Satisfaction

De Live Scores geven per gegeven feedback de score aan met kleuren. Deze kleuren komen overeen met de kleuren in de customer satisfaction.

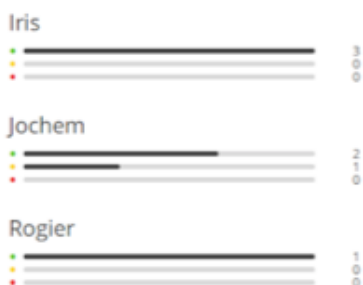
Score live



Afbeelding 6: Live Score

De Operator Score geeft de tevredenheid aan per Score per behandelaar (operator).

Score behandelaars




Afbeelding 7: Operator Score

De Remarks zijn de schriftelijke toelichtingen die zijn gegeven door klanten bij een bepaald feedback verzoek.

Laatste opmerkingen

 snelle oplossing! Dank!

Karin Jansen-Bosveld | Haskoning DHV | Rogier

 Deskundig geholpen

Tine Warle | Koning en Hartman | Iris

 Snel en goed geholpen, top. Dankjewel!

Vivianne Voeten | Axians ICT B.V. | Jochem

Afbeelding 8: Remark

3.1.3 Wat wil MailPlus B.V. bereiken?

Meerdere klanten hebben interesse getoond in het inzetten van het SFS voor hun eigen organisatie. Naar aanleiding hiervan is besloten om van het SFS een product te maken, zodat klanten in staat zijn om op eenvoudige wijze de tevredenheid van hun eigen klanten doorlopend te kunnen meten. Doordat het SFS niet wordt gezien als een marketing tool wordt er met dit product een andere type gebruiker aangesproken. Hierdoor is het niet wenselijk om MailPlus data te vermengen met de data van het SFS. Om deze reden is ervoor gekozen om het SFS apart op te zetten. Klanten koppelen hun eigen ICT omgeving aan het SFS, zowel voor het kunnen verzenden van feedback verzoeken als voor het tonen van statistieken. Zelf dienen ze een dashboard te ontwikkelen om de statistieken te presenteren.

3.1.4 Waarom voldoet de huidige implementatie niet?

Op dit moment kan het SFS niet één-op-één worden gebruikt door meerdere gebruikers. De huidige implementatie is ontwikkeld voor intern gebruik door de support afdeling van Mailplus B.V. Ook de functionaliteiten van het SFS zijn hierop gericht. Tijdens de ontwikkeling van het SFS is er geen rekening gehouden met meerdere gebruikers. Dit betekent bijvoorbeeld dat het database ontwerp enkel bedacht is voor de huidige vorm van feedback vragen (de drie smiley's), dat de gegevens die je uit het systeem kunt opvragen beperkt zijn en dat de configuratie gedaan moet worden in de database. Het bericht dat verzonden wordt aan de klant is niet aan te passen en alleen beschikbaar in het Nederlands. Momenteel wordt er dan ook geen feedback gevraagd aan de Engels sprekende klanten van Mailplus B.V. Een groot probleem van het koppelen van meerdere gebruikers aan het huidige SFS is dat de gegeven feedback door elkaar gaat. Dit betekent dat iedere gebruiker, iedere feedback kan zien. Deze vermenging van gegevens is niet gewenst. Uit deze voorbeelden blijkt dat de huidige opzet niet voldoet om deze één-op-één in te kunnen zetten door de klanten van Mailplus B.V. Om dit te realiseren dient er een nieuwe opzet ontwikkeld te worden van het gehele SFS systeem. Hierbij wordt rekening gehouden met wensen vanuit nieuwe en bestaande gebruikers. Ook zal er worden gekeken naar wat vergelijkbare pakketten momenteel voor functionaliteit aanbieden.

3.2 Doelstelling

Het doel van de afstudeeropdracht is het ontwikkelen van een nieuw SFS waarmee gebruikers hun klanttevredenheid doorlopend kunnen meten. Via een REST-API koppeling is het mogelijk om incidenten, behandelaren en gebruikers toe te voegen, statistieken uit het systeem op te vragen en deze inzichtelijk te maken met een zelf te ontwikkelen dashboard.

3.3 Gewenste resultaat

Na het uitvoeren van de afstudeeropdracht heeft Mailplus B.V. een nieuw SFS. Het nieuwe SFS biedt klanten van Mailplus B.V. de mogelijkheid hun klanttevredenheid te registreren. Ook is het mogelijk om het systeem intern op meerdere afdelingen in te zetten. Het nieuwe SFS bestaat uit een backend met dataopslag en een REST-API waarmee gecommuniceerd kan worden. De front-end bestaat uit een webpagina waar gebruikers de feedback kunnen versturen en een reactie op kunnen plaatsen. Er zal een eenvoudige applicatie voor presentatie doeleinden worden ontwikkeld. De nieuwe applicatie wordt bij Mailplus B.V. gehost. Hierdoor hoeven gebruikers deze applicatie niet zelf te hosten. Door middel van een koppeling met hun eigen systeem kunnen ze gebruik maken van het nieuwe SFS.

3.4 Op te leveren producten

Tijdens het uitvoeren van deze afstudeeropdracht worden de volgende (tussen)producten opgeleverd:

- Plan van Aanpak: Uitleg wat de opdracht inhoudt, inclusief een gedetailleerde planning.
- Klassen- en Database diagram huidige implementatie: Om de huidige implementatie beter te begrijpen wordt hiervan een klassen- en database diagram opgesteld.
- Uitwerking onderzoek naar gelijksoortige systemen (concurrentie onderzoek). Het onderzoeksrapport van het gedane onderzoek naar gelijksoortige systemen op de markt.
- Geïnterpreteerde (niet)functionele eisen.
- Database ontwerp ter ondersteuning van de nieuwe applicatie.
- Detail software ontwerp van de nieuwe applicatie.
- Test ontwerp, -plan en -rapportage voor de nieuwe applicatie.
- Documentatie van de REST-API.
- Het afstudeerverslag.

4. Methoden en technieken

In dit hoofdstuk worden de methoden en technieken besproken die tijdens deze afstudeeropdracht gebruikt zijn.

4.1 Proces/ Algemeen

4.1.1 Scrum

Op de development afdeling van Mailplus B.V. wordt gewerkt met Scrum. Om deze reden is ook tijdens deze opdracht gewerkt met de methodologie Scrum. Scrum is de manier van werken die de effectiviteit van teams verhoogt. Het voordeel van Scrum is dat projecten worden opgedeeld in korte sprints, waarin tussenproducten worden opgeleverd. Hierdoor is het zichtbaar waaraan gewerkt wordt en wat de voortgang is. Regelmatig overleg en feedback zorgt er tevens voor dat er een effectieve en intensieve samenwerking is wat resulteert in een effectievere aanpak van de opdracht (de Boer, Bruggink, Bruns et al., 2015).

Vanwege bovenstaande voordelen is daarom ook gekozen om Scrum te gebruiken tijdens deze afstudeeropdracht. Om het overzicht en de structuur te bewaren wordt door van Loon, Barendsen & Hoppenbrouwers (2015) aangeraden om gebruik te maken van een Scrum board. Dit wordt op zowel de development afdeling als tijdens deze opdracht digitaal bijgehouden middels Jira. Op dit Scrumboard worden de volgende drie kolommen bijgehouden; to do, in progress en done. Een sprint duurt twee weken. Na deze 2 weken wordt teruggekeken hoe de sprint is gegaan, de sprint retrospective, en welke producten er zijn opgeleverd. Wanneer blijkt dat een user story niet (geheel) is afgerond, zal deze terug worden geplaatst in de product backlog zodat deze met een volgende sprint kan worden meegenomen. Tevens wordt er een planning en indeling gemaakt voor de volgende sprint. Bij deze besprekingen zijn de student en de bedrijfsmentor aanwezig.

4.1.2 Implementatie Scrum

De werkwijze van Scrum is te uitgebreid om binnen de afstudeeropdracht één op één te kunnen implementeren. Daarom is Scrum aangepast op een aantal punten zodat het beter werkbaar is bij het aantal personen en de omvang van de opdracht.

4.1.2.1 Sprint 0

De eerste sprint waaraan wordt gewerkt levert geen shippable product op en kan daarom niet als een echte sprint worden gezien. Tijdens deze sprint zal de nadruk meer liggen op het inlezen van de huidige implementatie, het concurrentie onderzoek, het verkrijgen van requirements en de keuze en motivatie van het gekozen database model.

4.1.2.2 Planning

Scrum werkt met user story points die de moeilijkheid aangeven van een user story. In een sprint kunnen een bepaald aantal user story points worden uitgevoerd. Hoe hoger het aantal punten dat gegeven wordt, hoe ingewikkelder de implementatie wordt geschat en dus hoe meer tijd ervoor dient te worden gereserveerd. Door het gebruik van de verschillende, veelal voor de student nieuwe, technieken wordt het toekennen van deze punten vermoeilijkt. Ook zal de student deze technieken beter gaan beheersen naarmate de opdracht vordert. Hierdoor zal de waarde van een user story point veranderen en dienen deze gedurende de

afstudeeropdracht opnieuw gedefinieerd te worden. Hierdoor is ervoor gekozen om geen user story points te gebruiken tijdens de uitvoering van de afstudeeropdracht.

4.1.2.3 Rollen binnen Scrum

Binnen Scrum worden drie rollen gedefinieerd: Product Owner, Development Team en Scrum Master. Tijdens de uitvoering van de afstudeeropdracht zal de bedrijfsmentor de rol van product owner op zich nemen. Omdat de rol van scrum master niet gecombineerd dient te worden met de rol van product owner zal de student deze rol op zich nemen. Verder zal de student het enige lid zijn van het development team. Normaliter is een development team binnen Scrum groter, ongeveer drie tot zes personen. Ook worden er normaal gesproken niet meerdere rollen door dezelfde persoon uitgevoerd, maar door het beperkte aantal personen die werken aan deze opdracht is dit niet mogelijk.

4.1.2.4 Stand-ups

Normaal gesproken worden er dagelijks korte bijeenkomsten gehouden met alle leden uit een Scrum team. Deze bijeenkomsten worden Daily Scrums of Stand-up genoemd. Tijdens deze 15 minuten durende bijeenkomsten wordt er door de leden van het Scrum team drie vragen beantwoord:

- Wat heb ik gisteren gedaan?;
- Wat ga ik vandaag doen? En
- Zie ik problemen (uitdagingen), heb ik hulp nodig?

Mede door het kleinschalige team is ervoor gekozen om geen dagelijkse Stand-up in te plannen. Wel wordt er wekelijks overleg gepleegd tussen de student en de bedrijfsmentor om eventuele uitdagingen te melden en te bespreken. Tijdens deze afspraak wordt ook de retrospective uitgevoerd.

4.1.3 Concurrentieonderzoek

Voor het concurrentieonderzoek wordt gebruik gemaakt van twee methoden:

4.1.3.1 Sneeuwbal-effect

Een gevonden bron kan zelf ook referenties hebben naar andere bronnen. Door het volgen van deze referenties vind je steeds meer informatie. Dit wordt het sneeuwbal-effect genoemd. Het kan natuurlijk ook zijn dat er door het lezen van een bron, andere zoekwoorden en/of termen worden gevonden die weer gebruikt kunnen worden in de zoekopdrachten.

4.1.3.2 Horizontaal vergelijken

De methode horizontaal vergelijken kan worden gebruikt om ongestructureerde data te analyseren. Deze methode wordt vaak gebruikt om antwoorden van een interview te analyseren. Het concurrentieonderzoek resulteert ook in ongestructureerde data hierdoor kan deze methode goed worden ingezet.

4.2 Ontwerpen

4.2.1 Requirements

Voor het bepalen van de requirements voor het SFS wordt gebruikt gemaakt van de methode requirement development (De Swart, 2010). Deze methode bestaat uit vier processtappen. De eerste stap is het positioneren van het systeem binnen het businessdomein. De input voor deze

processtap is de probleembeschrijving zoals deze eerder is beschreven. Deze stap levert de business requirements en geïdentificeerde belanghebbende op.

Als tweede wordt de gewenste oplossing gedefinieerd. Tijdens dit proces worden de gebruikers- en software requirements, de systeemafbakening en de technische beperkingen nader bepaald.

Tijdens de derde stap worden de requirements gedetailleerd en hierdoor worden de specificaties van gebruikers- en software requirements duidelijk.

De vierde en laatste stap is het beheren van de requirements.

4.2.2 MoSCoW

Als prioriteringsmethode voor de requirements wordt gebruik gemaakt van de MoSCoW-analyse.

4.2.3 Database

4.2.3.1 Huidige implementatie

Om de huidige implementatie inzichtelijk te maken zal er van de huidige database een relational representation model (RRM) worden gemaakt. Aanwezige constraints worden weergegeven met behulp van object definition language (ODL).

4.2.3.2 Nieuwe implementatie

De implementatie van de nieuwe database zal per sprint, waar dit nodig is, worden uitgebreid. Het gegevensmodel zal op voorhand worden gemaakt. Hiervoor zal het RRM van de huidige implementatie als basis dienen en worden uitgebreid. Met behulp van ODL zullen de tabellen worden gespecificeerd. Om tot het nieuwe database ontwerp te komen zal het gegevensmodel worden genormaliseerd tot en met de derde normaalvorm. Doordat het nieuwe SFS meerdere gebruikers dient te bedienen zal er in het ontwerp rekening worden gehouden met concurrent gebruikers, performance- en beveiligingsaspecten.

4.2.3.3 TomCat Cluster

Het SFS zal gaan draaien op een TomCat Cluster met een MySQL database voor de gegevensopslag.

4.2.4 3-Lagen model

De huidige implementatie van het SFS maakt gebruik van het 3-lagen model. Hierdoor worden de verschillende verantwoordelijkheden binnen de applicatie onderverdeeld in 3 lagen. Het gebruik van het 3-lagen model zal worden gehandhaafd.

4.3 Implementatie

4.3.1 Spring Web MVC Framework

Spring Framework is een Java framework dat een alternatief biedt voor de standaard manier van ontwikkelen middels API's. Dit ontwerp deelt complexe toepassing op in drie eenheden, te weten 'model' 'view' en 'controller'. Deze onderscheiding in drie verschillende eenheden bevordert de leesbaarheid en de herbruikbaarheid van de code (Johnson, Hoeller, Sampaeanu et al., 2004). Hierdoor wordt de efficiëntie vergroot en kan de ontwikkelaar zich richten op het oplossen van het business probleem in plaats van technische problemen. Er is gebruik gemaakt van versie 3.2.5.

4.3.2 Jersey JAX-RS

JAX-RS is een in Java geschreven API voor het implementeren van REST-full Web Services. Het maakt veel gebruik van annotaties om het proces van ontwikkelen en uitrollen te vereenvoudigen.

Jersey is een uitbreiding op de JAX-RS toolkit met extra mogelijkheden om eenvoudig REST-full services te ontwikkelen. Ook zijn er mogelijkheden tot het uitbreiden van Jersey zodat het aangepast kan worden naar eigen wensen. (Oracle Corporation, 2016). Er is gebruik gemaakt van versie 2.6.

4.3.3 Jackson

Jackson is een lichte veelzijdige Java library die ontwikkeld is met het oog op snelheid, correctheid en eenvoudige implementatie door ontwikkelaars (FasterXML, 2013). Met deze library is het onder andere mogelijk om objecten om te zetten van en naar JSON. Ondanks dat Jackson meerdere implementaties biedt, wordt tijdens het afstuderen Jackson gebruikt voor JSON. Er is gebruik gemaakt van versie 2.2.2.

4.3.4 OAuth

OAuth is open protocol om op een eenvoudige en standaard manier veilige autorisatie te garanderen voor web-, mobiele- en desktop applicaties (OAuth.net, z.d.). Het OAuth protocol biedt websites en applicaties (Consumers) beveiligde toegang tot een web service (Service Provider) middels een API, zonder dat Users hun logingegevens van de Service Provider beschikbaar hoeven te stellen aan de Consumer. OAuth biedt een vrij te implementeren en algemene manier voor API authenticatie (OAuth.net, 2009).

4.3.4.1 OAuth 1.0a vs OAuth 2.0

Er zijn een aantal grote verschillen tussen OAuth 1a en OAuth 2.0. Bij OAuth 1a is de beveiliging niet afhankelijk van de transportlaag (HTTPS/ TLS). Doordat dit bij OAuth 2.0 wel het geval is, kan dit een beveiligingsrisico vormen. Dit kan gebeuren doordat een library die wordt gebruikt een kwetsbaarheid bevat, wat kan leiden tot een man-in-the-middle (MiTM) aanval. OAuth 2.0 maakt gebruik van 'Bearer-tokens'. Dit zijn random strings die worden meegestuurd met een request voor authenticatie. Deze 'Bearer-tokens' kunnen worden gestolen en/of gekopieerd en vervolgens door derden worden gebruikt. Wanneer dit gebeurt zijn alle OAuth 2.0 communicaties kwetsbaar. De access tokens bij OAuth 2.0 hebben een korte levensduur (geldigheid). Dit betekent ook dat de access token lifecycle bijgehouden dient te worden. Wat leidt tot extra implementaties.

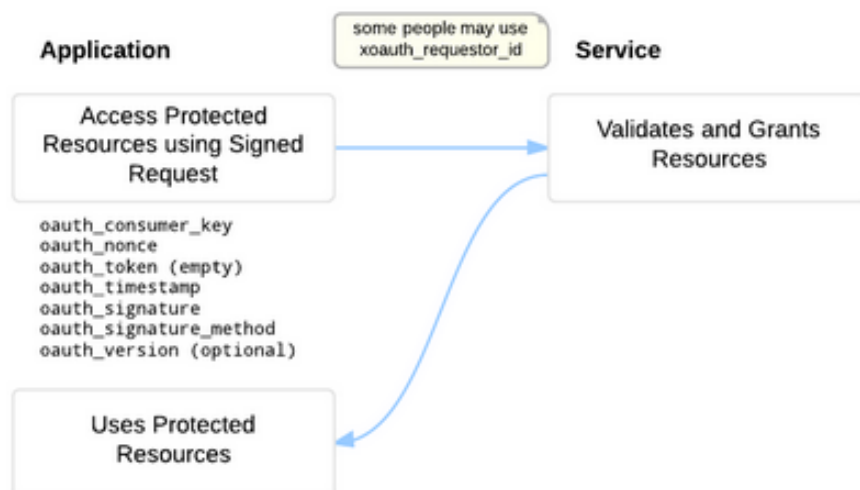
Mede door deze verschillen en omdat OAuth 1.0a One Legged reeds wordt gebruikt binnen MailPlus is er gekozen om nu ook OAuth 1.0a One Legged te gebruiken.

4.3.4.2 OAuth v1.0a One Legged

Om de SFS API te kunnen gebruiken dient de cliënt zich te authenticeren via OAuth. De versie van OAuth die wordt gebruikt is OAuth 1.0a One Legged. Het authenticeren via OAuth gebeurt in een aantal stappen (Authenticating with OAuth, 2009):

- De Consumer vergaart een niet geautoriseerde Request Token;
- De User autoriseert de Request Token;
- De Consumer wisselt de Request Token in voor een Access Token.

Nadat de authenticatie stappen zijn doorlopen kan de Consumer requests maken met de API. In de implementatie zoals deze in Mailplus is toegepast wordt ieder request opnieuw ondertekend en dus ook gevalideerd. Hierdoor wordt er geen gebruik gemaakt van request- en accesstokens. Hieronder staat dit schematisch weergegeven.



Abbeelding 9: Werking OAuth 1.0a one legged bron: <http://oauthbible.com/>

OAuth is een manier van autorisatie en geen manier van data encryptie. Om ervoor te zorgen dat de data die uitgewisseld wordt tussen de Service Provider (het SFS) en de Consumer beveiligd is, wordt er gebruikt gemaakt van HTTPS.

4.3.5 EclipseLink

EclipseLink JPA biedt ontwikkelaars een gestandaardiseerde Object-Relational persistence oplossing met de ondersteuning voor vele features (The Eclipse Foundation, 2015). EclipseLink wordt binnen de afstudeeropdracht ingezet als Object Relational Mapper (ORM). Er is gebruik gemaakt van versie 2.5.1.

4.3.6 Velocity

Velocity is een Java-based template engine. Velocity maakt het mogelijk om door middel van een simpele, maar krachtige template language gebruik te maken van objecten uit Java code. Velocity past goed binnen MVC en scheidt Java code (Model en Controller) van de webpagina's (View). Hierdoor worden webpagina's beter te onderhouden. Velocity is een goed alternatief voor JSPs of PHP (The Apache Software Foundation, 2016). Er is gebruik gemaakt van versie 1.7.

4.4 Testen

4.4.1 TestGoal

Tijdens het afstuderen wordt er gebruik gemaakt van resultaat gedreven testen volgens de TestGoal-aanpak. De TestGoal-aanpak (Grood, 2008) beschrijft niet de gebruikte testmethodes, maar ondersteunt de tester in het benutten van zijn expertise en kennis en in het hanteren van de testprincipes.

4.4.2 JUnit

JUnit is een geautomatiseerd test framework. Het is een simpel, open source framework waarmee test kunnen worden geschreven en herhaaldelijk kunnen worden uitgevoerd. Het is een onderdeel van de xUnit architectuur (JUnit, z.d.). Tijdens de uitvoering van het project wordt gebruik gemaakt van JUnit versie 4.7.0.

4.4.3 Hamcrest

Hamcrest is een uitbereiding op JUnit. Hamcrest maakt gebruik van de `assertThat` methode aangevuld met matcher expressies om de test te valideren. Hamcrest heeft tot doel om tests zo leesbaar mogelijk te maken. Dit geldt ook voor de error messages, deze zijn duidelijker doordat de expected- en actual waarden worden getoond in de error (Purpose of the Hamcrest matcher framework, Lars Vogel, 2016). Er is gebruik gemaakt van versie 1.3.

4.4.4 Mockito

Mockito is een mocking framework voor Java. Hiermee kan op eenvoudige wijze de klasse die getest wordt geïsoleerd worden van alle afhankelijkheden. Door het gebruik van mocking kan er met zekerheid worden gesteld dat eventuele fouten voortkomen uit de klasse die wordt getest en niet uit een andere klasse. Met Mockito is het mogelijk om methodes te stubben, hierdoor kun je per methode aangeven hoe deze wordt aangeroepen en welk response je terug wilt krijgen. Er is gebruik gemaakt van versie 2.3.4.

4.4.5 HSQLDB

HyperSQL DataBase (HSQLDB) is relationele database software geschreven in Java. Het biedt een kleine, multithread en transactionele database engine die zowel in-memory als disk-based kan worden ingezet. Ook is het mogelijk om HSQLDB embedded of als server te draaien. Verder ondersteunt HSQLDB een breed aanbod aan SQL standaarden (The HSQL Development Group, mei 2016). HSQLDB wordt binnen het afstudeertraject gebruikt als Fake-object voor de database. Hierdoor blijft de test data gescheiden van de live data en is deze voor iedere test gelijk. Binnen de test omgeving zal HSQLDB runtime als embedded database worden ingezet. Voor iedere test case wordt er een nieuwe database opgezet, met nieuwe test data erin. Er is gebruik gemaakt van versie 2.3.2.

5. Sprint 0

Tijdens de initiële sprint van het afstudeertraject zijn er een aantal activiteiten uitgevoerd om de huidige implementatie te doorgronden, concurrenten bekeken er zijn de requirements verzameld die als basis dienen voor het nieuwe SFS systeem. Als laatste zijn er ontwerp keuzes gemaakt met betrekking tot het database ontwerp zodat deze meerdere gebruikers kan toestaan.

5.1 Inlezen en inventarisatie huidige implementatie

Aan de start van het afstudeertraject is met de bedrijfsmentor de huidige implementatie besproken. Hij heeft de globale werking en de architectuur van het systeem uitgelegd en heeft toegang gegeven tot verschillende bronnen. In deze bronnen staan de huidige requirements en keuzes die gemaakt zijn tijdens het ontwikkelen van de eerste versie van het SFS. Tevens is er toegang verkregen tot de broncode van de applicatie en van de huidige feedback monitor.

Om een beter inzicht te krijgen in de huidige implementatie zijn deze documenten uitvoerig bestudeerd. De beschrijving van de gemaakte keuzes en de requirements zijn zorgvuldig gelezen en later opgenomen in de requirements van het nieuwe systeem. De broncode van de huidige feedback monitor is bekeken om een beter beeld van het geheel te ontwikkelen. Deze code is verder niet geanalyseerd omdat het ontwikkelen van een dashboard buiten de scope van de opdracht valt.

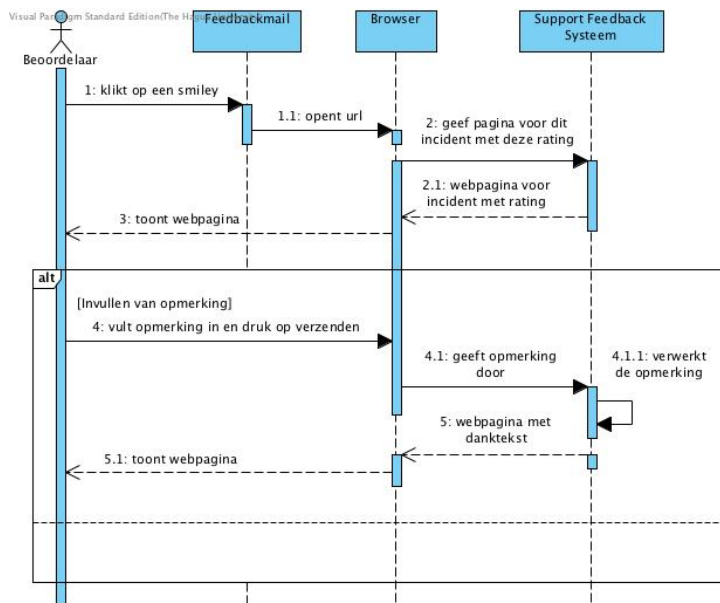
De broncode van de huidige implementatie is vervolgens omgezet naar een klassendiagram¹ met behulp Visual Paradigm. Hiervoor is gekozen om een helicopterview te creëren van de huidige implementatie en om te dienen als basis voor het nieuwe ontwerp. Van de volgende belangrijke processen uit het SFS zijn sequentiediagrammen gemaakt:

- Het omzetten van het e-mail bericht naar een feedback verzoek, het creëren en het verzenden van de feedback mail².

Het proces van het verwerken van een ontvangen beoordeling die gegeven is door een beoordelaar is schematisch weergegeven in onderstaande afbeelding.

¹ Bijlage I: Klassendiagram huidige implementatie

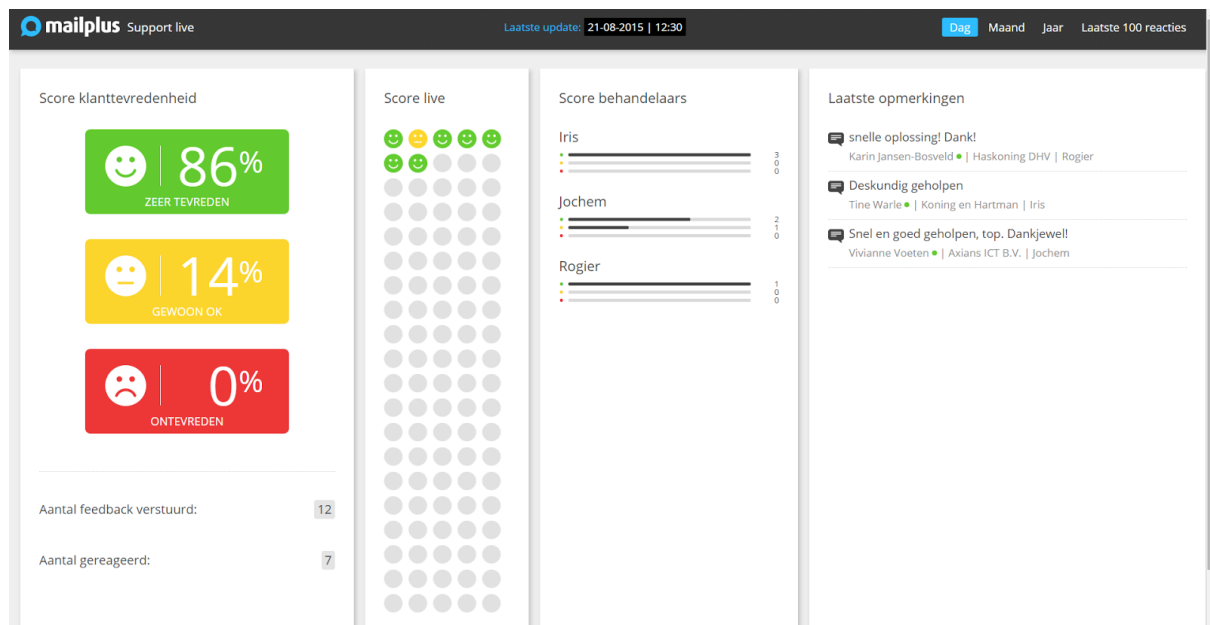
² Bijlage II: Sequentiediagrammen key-features huidige implementatie



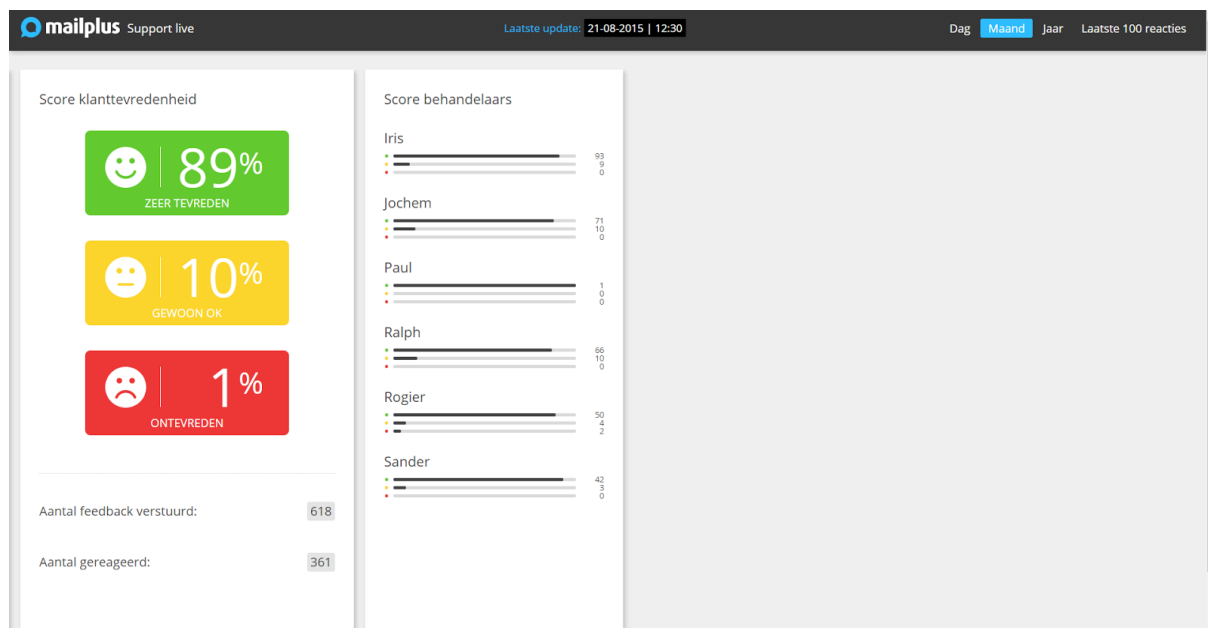
Afbeelding 10: Schematische weergaven geven van feedback

Door de hierboven genoemde processen via de genoemde methoden in kaart te brengen is de algehele werking en architectuur van het huidige systeem in kaart gebracht en visueel gemaakt.

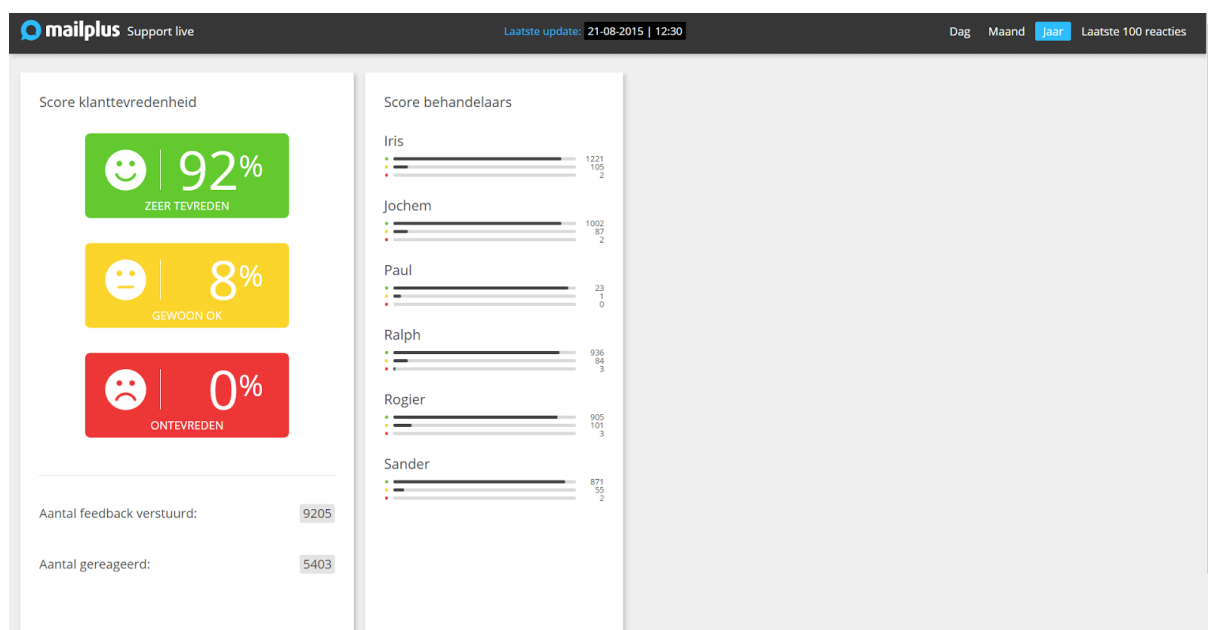
Naast de broncode bestaat de huidige implementatie uit een MySQL database. In de database wordt de gegeven feedback opgeslagen zodat deze data uitgelezen en getoond kan worden in de vorm van een dashboard. Het dashboard bestaat uit een aantal schermen waarop statistieken worden weergegeven.



Afbeelding 11: Huidig dashboard toont dag scherm



Afbeelding 12: Huidig dashboard toont maand scherm



Afbeelding 13: Huidig dashboard toont jaar scherm

De huidige database bestaat uit twee tabellen. Door deze kleinschalige en overzichtelijke opzet is er voor gekozen om hier alleen een relationeel representatiemodel van te maken, zie bijlage III³. Het gemaakte relationeel representatiemodel dient als basis voor de opzet van de database voor de nieuwe implementatie.

5.2 Concurrentieonderzoek

Er is, na het inventariseren van de huidige implementatie, een goed beeld ontstaan van het te ontwikkelen systeem. Vervolgens is er gekeken naar vergelijkbare, concurrerende systemen die op de markt zijn. Het doel van het concurrentieonderzoek is als volgt beschreven: “Het

³ Bijlage III: RRM huidige implementatie

doel van het concurrentieonderzoek is het achterhalen welke functionaliteiten andere systemen bieden voor het uitvragen en analyseren van feedback. In het bijzonder feedback over de tevredenheid van klanten kort na een geleverde dienst.”⁴

Het onderzoeken van vergelijkbare systemen kan een goede input leveren voor functionaliteiten van het te ontwikkelen feedback systeem. Het onderzoek is uitgevoerd in de vorm van een literatuur onderzoek. Aan de hand van de huidige implementatie zijn functionaliteiten onttrokken die de basis zijn geweest voor het opstellen van een vijftal vragen. Aan de hand van deze vragen zijn de concurrenten onderzocht.

De onderzochte concurrenten zijn gevonden door middel van verschillende zoekwoorden en het gebruik van de zoekmachine Google. De zoekwoorden bestaan uit voor de hand liggende steekwoorden die gebruikt zouden kunnen worden door iemand die opzoek is naar een applicatie om feedback te verzamelen. De gebruikte zoekwoorden hebben een grote lijst met bedrijven opgeleverd. De lijst met bedrijven was zo uitgebreid dat deze op basis van opgestelde criteria is uitgedund tot vijf concurrenten welke nader zijn onderzocht op basis van de vijftal opgestelde vragen.

Per concurrent zijn de vijf vragen beantwoord en per vraag genoteerd in tabelvorm. De opgeleverde antwoorden zijn te categoriseren als ongestructureerde data en zijn volgens de analyse methode horizontaal vergelijken met elkaar vergeleken. Deze methode houdt in dat de antwoorden per concurrent met elkaar worden vergeleken en de bevindingen worden genoteerd in de laatste kolom. Hierdoor zijn de verschillen en overeenkomsten overzichtelijk gemaakt voor gebruik in de conclusie.

Uit het onderzoek is onder andere naar voren gekomen dat ieder systeem toegang heeft tot een uitgebreid dashboard waar statistieken worden getoond. Het ontwikkelen van zo een dashboard (feedback monitor) valt buiten de scope van deze opdracht. Er wordt echter wel, via een rest-api, de mogelijkheid geboden om statistieken op te halen uit het systeem. Door middel van deze rest-api kan een eigen feedback monitor worden ontwikkeld. Een ander opvallend punt dat uit het onderzoek is gebleken is dat er, door verschillende systemen, de mogelijkheid wordt geboden om het versturen van de feedback mails te beperken. Dit is een feature die is meegenomen in de requirements voor de nieuwe implementatie.

5.3 Requirements

Na de stappen die hierboven zijn genoemd is er een goed beeld ontstaan van het huidige SFS en concurrenten. De vervolg stap is het inventariseren van de requirements voor de nieuwe implementatie.

5.3.1 Requirements huidige implementatie

Voor het inlezen van de huidige implementatie is er toegang verkregen tot de requirements en gemaakte keuzes van de huidige implementatie. De requirements die eerder zijn opgesteld zijn ook in de nieuwe requirements opgenomen. Hiervoor is gekozen omdat bepaalde requirements nog steeds van belang zijn en om deze te waarborgen binnen Jira. Deze requirements zijn in te zien in Bijlage V⁵ met als belanghebbende ‘Product management’.

⁴ Bijlage IV: Het Concurrentieonderzoek

⁵ Bijlage V: Requirements

5.3.2 Input belanghebbende

Het vaststellen van de behoeften en eisen van de belanghebbende is op verschillende manieren gebeurd. Dit is per belanghebbende op een andere wijze aangepakt. Sinds de huidige versie van het SFS in gebruik is hebben verschillende klanten interesse getoond. De opdrachtgever heeft in deze periode, vanaf het begin van het gebruik tot nu, gesprekken gevoerd met deze potentiële klanten en hun wensen en ideeën bijgehouden. Het document met zijn aantekeningen heeft hij aangeboden voor het gebruik bij de afstudeeropdracht. De behoeften van de geïnteresseerde klanten waren reeds geïnventariseerd in de periode voorafgaand aan deze afstudeeropdracht. De wensen en eisen van product management waren ook reeds bekend en aangeleverd. Deze zijn verwerkt zoals hierboven bij “Requirements huidige implementatie” staat beschreven. Aan de medewerkers van de afdeling support is gevraagd naar inbreng voor het nieuwe systeem. Zij zijn immers de personen die dagelijks werken met het SFS. Er is aan hen gevraagd of ze in de huidige implementatie bepaalde functionaliteiten misten en/of ze verbeteringen konden aandragen voor het nieuwe systeem. De verzamelde input van de verschillende belanghebbenden zijn opgenomen in bijlage VI⁶.

5.3.3 Prioritering requirements

Als prioriteringstechniek is er gekozen om gebruik te maken van de MoSCoW-analyse. Er zijn veel verschillende technieken om requirements te prioriteren. Er is gekozen voor de MoSCoW-analyse omdat deze methode van prioriteren een minder subjectieve onderverdeling met zich meebrengt dan wanneer er bijvoorbeeld wordt gewerkt met de gradaties laag, midden, hoog. Of met de methode ‘tevredenheids- en ontevredenheidsratio’s’ waar de belanghebbenden kunnen aangeven in hoeverre ze tevreden zullen zijn met de implementatie van een bepaalde requirement en hoe ontevreden wanneer deze niet wordt geïmplementeerd. Met deze methodes zie je dat deze vooral gericht zijn op de toegevoegde waarde van een bepaald requirement ten opzichte van een andere. Hierdoor zijn deze indelingen redelijk subjectief.

Nummer	Omschrijving	Prioritering	Belanghebbende
UR12	Een Feedback verzoek bestaat minimaal uit: <ul style="list-style-type: none">• Gegevens van te beoordelen hulpvraag (uniek ID en korte omschrijving);• Gegevens behandelaar (Voornaam, achternaam, e-mail en nummer); Gegevens beoordelaar (Voornaam, achternaam, e-mail en bedrijfsnaam)	M	Teammanager support
UR24	Via de Rest-API is het mogelijk om de statistieken op te vragen van het SFS	M	Geïnteresseerde klanten
UR31	Het SFS kan een Feedback verzoek genereren	M	

Tabel 2: Voorbeelden requirements na MoSCoW-analyse: prioriteit Must have

Het nieuwe SFS heeft minimale input nodig om een incident aan te kunnen maken in de database. Hiervoor zijn de genoemde gegevens in UR12 voor nodig. Een grote toevoeging aan het nieuwe SFS ten opzichten van de huidige versie, is het uitbreiden van de Rest-API

⁶ Bijlage VI: Input voor requirements

calls. Hierdoor is het opvragen van statistieken een belangrijk requirement en heeft UR24 deze prioritering gekregen. Het SFS zal niet werken als beoordelaars geen feedback verzoek toegestuurd krijgen. Zonder zo een verzoek kunnen zij geen beoordeling geven. Hierdoor is UR31 van belang voor de bruikbaarheid van het SFS.

Nummer	Omschrijving	Prioritering	Belanghebbende
UR08	De dag-, maand- en jaar overzichten geven de cijfers van de huidige periode en die van de vorige periode (dag/ maand/ jaar).	S	Teammanager Support
UR17	Het SFS kan overweg met verschillende talen (NL en EN) { Mogelijk door het aanpasbaar maken van de teksten in de front-end}	S	Geïnteresseerde klanten
UR30	Bij de beoordelingen op het dag overzicht heb je de mogelijkheid tot het zien bij welk feedback verzoek deze hoort (naam beoordelaar, bedrijfsnaam en callnummer).	S	Medewerker support

Tabel 3: Voorbeelden requirements na MoSCoW-analyse: prioriteit Should have

Hoewel het geven van statistieken over de huidige- en vorige periode (UR08) een belangrijk requirement is, is het niet zo dat het SFS niet werkt wanneer het SFS alleen de statistieken van de huidige periode terug geeft. Hetzelfde geldt voor de requirements UR17 en UR30. Hierdoor hebben deze requirements niet de prioriteit 'Must have' gekregen. Het resultaat van requirement UR08 is tweeledig. Het beschrijft de wens om de periode statistieken per dag/ maand/ jaar niet alleen voor de huidige periode te geven, maar ook voor de voorgaande periode. Het is niet mogelijk om één van beide te implementeren. UR17 is belangrijk voor het nieuwe SFS, omdat momenteel Engels sprekende klanten geen mogelijkheid hebben tot het geven van feedback. Dit geeft de ernst aan van dit requirement, een (groeïend) gedeelte van de klanten kan geen beoordeling geven omdat ze geen Nederlands spreken. UR30 resulteert erin dat medewerkers op een eenvoudige manier kunnen achterhalen wie welke beoordeling heeft gegeven en voor welk ticket (support vraag) deze is. Zonder deze functionaliteit zal het systeem goed werken. Ook is er voor dit requirement een workaround, namelijk het opzoeken in de database. Echter is deze workaround niet voor iedereen toegankelijk, door rechten, en zeker ook niet wenselijk. Deze genoemde redenen hebben ervoor gezorgd dat de requirements uit het voorbeeld de prioritering 'Could have' hebben gekregen boven 'Should have' en 'Won't have this time'.

Nummer	Omschrijving	Prioritering	Belanghebbende
UR10	De tekst van een Feedback verzoek kan worden aangepast.	C	Teammanager support
UR13	Het is mogelijk om een Feedback verzoek te beperken voor een bepaalde beoordelaar (blacklist/ afmeldingen bijhouden (helemaal niet sturen), maximaal aantal per tijdseenheid {dag, week, maand})	C	Teammanager support
UR14	Wanneer een aanvraag niet leidt tot een Feedback verzoek wordt de reden hiervan opgeslagen in de database bij de bijbehorende aanvraag. (bijvoorbeeld door blacklist, missen van e-mail adres)	C	

Tabel 4: Voorbeelden requirements na MoSCoW-analyse: prioriteit Could have

De nieuwe implementatie van het SFS zal zonder problemen werken wanneer de tekst van een feedback verzoek (UR10) niet kan worden aangepast, echter is het een goede functionaliteit om toe te voegen. Zeker wanneer er meerdere gebruikers gebruik gaan maken van het nieuwe SFS. Ook UR13 heeft zeker een toegevoegde waarde aan de functionaliteiten van het nieuwe SFS, maar is niet van belang voor de werking van het systeem en kan ook in een latere release worden geïmplementeerd. UR14 hangt nauw samen met UR13 en heeft daarom dezelfde prioriteit gekregen. Deze requirements hebben niet de prioriteit ‘Won’t have this time’ gekregen doordat ze wensen vertegenwoordigen van meerdere belanghebbende.

Nummer	Omschrijving	Prioriteit	Belanghebbende
UR15	Het verzendmoment van het feedback verzoek is aanpasbaar: <ul style="list-style-type: none"> • op een vast tijdstip • met een vertraging van x-aantal minuten 	W	Teammanager support
UR16	Het systeem kan met extra velden omgaan. M.a.w. toe te voegen in het systeem, vullen en verwerken in de mail.	W	Geïnteresseerde klanten
UR20	Rapportage vanuit het systeem of in een tool waar de standaardrapportage bijgehouden wordt. Opties zoals download naar pdf, rapportage mailen naar een e-mailadres.	W	Geïnteresseerde klanten

Tabel 5: Voorbeelden requirements na MoSCoW-analyse: prioriteit Won’t have this time

De opzet voor het nieuwe SFS is voornamelijk dat deze werkt via een Rest-API koppeling. Hierdoor kan een gebruiker het eigen systeem eenvoudig koppelen met het SFS. Mede door deze opzet kan de gebruiker zelf bepalen wat het gewenste verzendmoment is (UR15). Het SFS hoeft deze functionaliteit niet perse aan te bieden. Het doel van het SFS is het versturen van standaard mails naar beoordelaars met de mogelijkheid tot het geven van een beoordeling. De gegevens in deze mails hebben betrekking op het incident. Verder wordt het mogelijk om de tekst van deze mails naar wens aan te passen. Hierdoor heeft UR16 de prioritering ‘Won’t have this time’ gekregen. Voor UR20 geldt dat dit requirement door de klant kan worden ingebouwd in het eigen systeem.

5.3.4 Database ontwerpkeuze: Meerdere gebruikers

Een van de grootste eisen aan het nieuwe systeem is dat het door meerdere klanten (accounts) gebruikt kan worden. Echter dient de opzet zo te zijn dat er geen datavermenging plaats kan vinden. Hiermee wordt bedoeld dat de data van gebruiker A alleen ingezien, bewerkt en opgevraagd kan worden door gebruiker A en niet door anderen.

Om dit te bereiken zijn er een aantal oplossingen mogelijk. Deze oplossingen zijn niet uitputtend, maar dit zijn de mogelijkheden die in overleg met de bedrijfsmentor zijn bedacht. Hierna worden de bevonden opties besproken en per optie de voor- en nadelen genoemd. Aan de hand van deze punten wordt er beargumenteerd een keuze gemaakt voor één van de opties.

5.3.4.1 Voor iedere gebruiker een aparte database

Een van de mogelijke oplossingen is om de data voor ieder account op te slaan in een eigen database. Aan de hand van de OAuth consumer key wordt dan bepaald met welke database connectie gemaakt dient te worden om het api-request te verwerken. Het grote voordeel hiervan is dat de data per gebruiker gescheiden is. Datavermenging is hierdoor vrijwel uitgesloten. Nadelen aan deze opzet zijn onder andere dat er per api-request een connectie opgezet dient te worden met de juiste database.

Het opzetten van een database connectie kost enkele seconden, tegen enkele milliseconden om een query uit te voeren (api-request). Doordat iedere gebruiker een eigen database heeft, dient de database connectie steeds opnieuw te worden opgezet. Dit levert een connectie overhead op. Binnen programmeren komt het vaker voor dat er objecten nodig zijn die kostbaar zijn om te creëren, zoals het maken van een database connectie.

Door gebruik te maken van het creational design pattern: 'Object Pool' zou dit probleem op een algemene wijze aangepakt kunnen worden. Doordat er steeds connectie gemaakt dient te worden met een andere database, vermoeilijkt deze oplossing het gebruik van het 'Object Pool Pattern'.

5.3.4.2 Voor iedere gebruiker aparte tabellen in één database

Een andere mogelijke oplossing is om ieder account eigen tabellen te geven in één database. Hierdoor heb je het voordeel dat er gebruik gemaakt kan worden van het 'Object Pool Pattern', omdat er nu de connectie met de enkele database uitgewisseld kan worden tussen meerdere api-requests. Ook met deze oplossing is alle data van de verschillende accounts gescheiden.

Echter betekent deze opzet dat de queries complexer worden doordat de kolomnamen variabel zijn. Dit levert een nadelig effect op met het gebruik van annotaties binnen de Java Persistence API (JPA). Deze zijn bedoeld om databasetabellen te mappen aan Java Klassen. Ook betekent dit dat het aantal tabellen in de database evenredig zullen toenemen naar mate het aantal gebruikers van het systeem groeit.

5.3.4.3 Één database met een unieke identifier voor ieder account in de tabellen

Het probleem dat meerdere accounts gebruik moeten kunnen maken van het systeem, is nog op een andere manier op te lossen. Deze laatste methode maakt gebruik van één database met de bijbehorende tabellen. In de tabellen wordt een unieke identifier meegegeven die aangeeft van welk account de data is. Dit levert als voordeel op dat het aantal tabellen beperkt blijft en dat de queries niet veel complexer worden. Wanneer queries relatief eenvoudig blijven komt dit de testbaarheid ten goede.

Doordat er gebruik gemaakt wordt van een enkele database is het ook hier mogelijk om gebruik te maken van het 'Object Pool Pattern', wat de snelheid van het systeem ten goede komt. Ook kunnen de annotaties van de JPA gebruikt worden.

Doordat de data van de verschillende accounts in dezelfde tabel staat is data vermenging een mogelijk probleem. Wanneer er een fout wordt gemaakt in de code, kan het gebeuren dat account A niet (alleen) de data terug krijgt van zichzelf, maar ook de data van account N. Echter is dit probleem goed te ondervangen door het systeem uitvoerig te testen. Doordat alle data van de accounts in één database worden opgeslagen, kan deze database dusdanig groot worden dat dit problemen kan opleveren met de schaalbaarheid. Volgens de MySQL handleiding houdt schaalbaarheid het volgende in: “Scalability refers to the ability to spread both the database and the load of your application queries across multiple servers” (Chapter 17 High Availability and Scalability, Oracle Corporation and/of its affiliates (2016).

5.4 Keuze en motivatie: Ontwerpkeuze om meerdere gebruikers te bedienen

Hierboven zijn drie mogelijke oplossingen aangedragen om meerdere gebruikers mogelijk te maken binnen het databaseontwerp. Per oplossing zijn zowel voor- als nadelen genoemd.

Na het in overweging nemen van deze voor- en nadelen is ervoor gekozen om te gaan voor een enkele database met een unieke identifier voor iedere gebruiker. De reden dat hiervoor is gekozen heeft voornamelijk te maken met de voordelen van deze aanpak en dat de verwachte nadelen goed te ondervangen zijn. Doordat deze oplossing de mogelijkheid biedt tot het gebruik van het ‘Object Pool Pattern’ zal de connectie overhead worden beperkt. Het aantal tabellen blijft overzichtelijk en de queries hoeven door deze oplossing niet onnodig complex te worden. Ook de mogelijkheid tot het gebruiken van JPA annotaties geniet de voorkeur. Zoals aangegeven zijn de nadelen goed te ondervangen. Het probleem van datavermenging dat zou kunnen ontstaan, is door middel van goed testen, terug te brengen tot een nihil risico. Dat deze opzet problemen kan opleveren met de schaalbaarheid van het systeem wordt voor nu geaccepteerd.

Een mogelijk probleem is het aantal rijen dat een MySQL tabel aankan. Het aantal rijen dat een tabel aankan is onder andere afhankelijk van de grootte van de gebruikte harde schijven, het gebruikte besturingssysteem, de gebruikte storage engine en het gebruikte data type voor de primary key. De eventuele beperkingen van het besturingssysteem en de opslag capaciteit van de harde schijven laten we buiten beschouwing. De database storage engine InnoDB heeft een maximale grootte van 64TB per tabel (15.8.8 Limits on InnoDB Tables, onbekend). Geschat wordt dat dit voldoende ruimte biedt.

De keuze voor het datatype van de primary key, kan het aantal rijen beperken. Voor alsnog is er gekozen voor een integer, welke maximaal 2.147.483.647 mogelijkheden kan opslaan. Uit het document ‘Requirements Paul Nederlof.docx’ blijkt dat er 20 tot 25 klanten interesse hebben getoond in het Support Feedback Systeem. In 2015 heeft de support afdeling van MailPlus B.V. afgerond 15.000 feedbackverzoeken geregistreerd in het systeem. In het eerste kwartaal van 2016 zijn er door deze afdeling 8400 verzoeken tot feedback verstuurd. Dit zou impliceren dat er 16.800 verzoeken worden verstuurd in 2016.

Wanneer we rekening houden met 25 gebruikers die elk 20.000 verzoeken per jaar registreren in het systeem, resulteert dit in 600.000 feedbackverzoeken per jaar. Hieruit kunnen we concluderen dat deze oplossing voorlopig (2.1 miljard mogelijkheden/ 600.000 verzoeken per jaar = 3.500 jaar) genoeg ruimte biedt.

5.5 Reflectie sprint 0

Deze sprint heeft veel duidelijk gemaakt voor mij over de werking van de huidige implementatie en hoe het nieuwe SFS eruit dient te komen zien. Er is veel tijd gestoken in het

uitwerken van de huidige implementatie, maar ik ben ervan overtuigd dat dit werk zijn vruchten gaat afwerpen tijdens deze afstudeeropdracht.
Sprint 0 heeft geen echt shippable product opgeleverd, zoals eerder is aangegeven in hoofdstuk 4.1.2.

6. Sprint 1

Tijdens sprint 1 van het afstudeerproject zijn de volgende activiteiten uitgevoerd:

- Opzetten ontwikkelomgeving
- Uitwerken requirements UR12 en UR32
- Sprint retrospective
- Reflectie sprint 1

6.1 Opzet ontwikkelomgeving

Aan het begin van de eerste sprint is samen met de bedrijfsmentor een ontwikkelomgeving ingericht. Hiervoor is de benodigde software geïnstalleerd, de broncode van de huidige implementatie vanuit het versiebeheer systeem (Git) verkregen en toegang tot Jira verleend. Nadat dit alles was ingesteld zijn alle requirements in Jira gezet met de juiste prioritering. Vervolgens is de eerste sprint aangemaakt met twee requirements.

6.2 Uitwerken requirements UR12 en UR32

In de eerste sprint worden twee user stories uitgewerkt die te maken hebben met het genereren van een feedbackverzoek. Requirement UR12 voegt een korte omschrijving toe aan het feedbackverzoek en de feedbackmail die naar de beoordelaar wordt verzonden. Klanten stellen regelmatig meerdere vragen per dag, door het toevoegen van een korte omschrijving is het voor hen duidelijker om welke hulpvraag het feedbackverzoek gaat. UR32 maakt het mogelijk om via de Rest-API een feedbackverzoek te genereren.

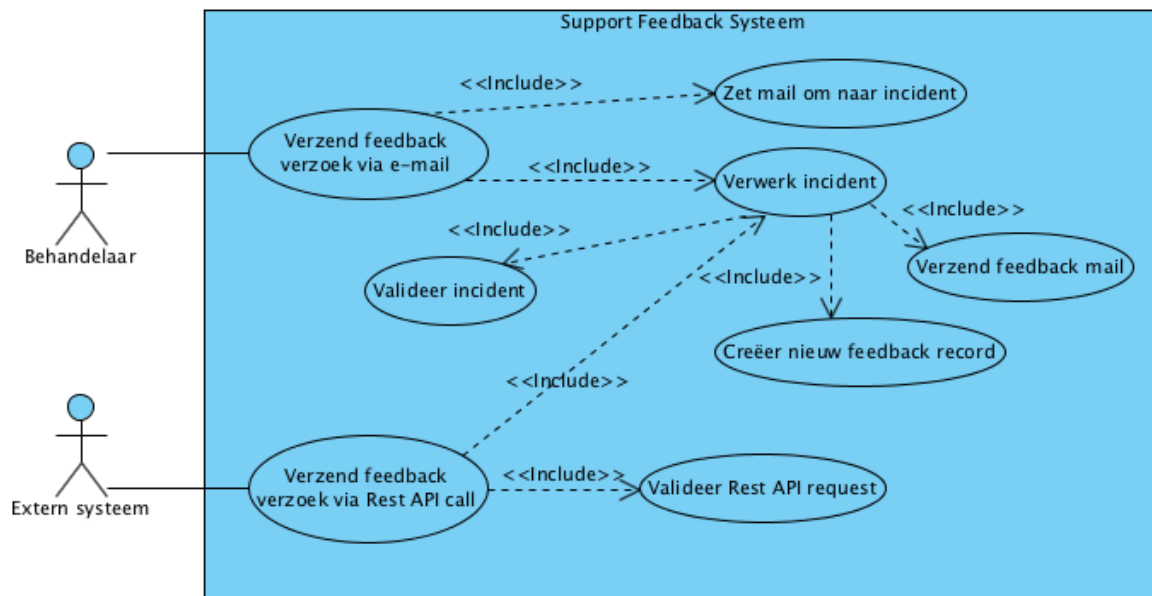
De keuze is gevallen op deze twee user stories, omdat dit core functionaliteit is van het systeem en de gehele codebase wordt geraakt. Ook is ingeschat dat deze user stories een goed beeld kunnen geven van de zwaarte van een sprint, zodat de volgende sprints goed kunnen worden ingepland.

Nummer	Omschrijving	Prioritering	Belanghebbende
UR12	Een Feedback verzoek bestaat minimaal uit: <ul style="list-style-type: none">• Gegevens van te beoordelen hulpvraag (uniek ID en korte omschrijving);• Gegevens behandelaar (Voornaam, achternaam, e-mail en nummer);• Gegevens beoordelaar (Voornaam, achternaam, e-mail en bedrijfsnaam)	M	Teammanager support
UR32	Het SFS kan op de volgende manieren een Feedback verzoek genereren: per mail en via een REST-API call (voor het gebruik met Social Media).	M	

Tabel 6: Geselecteerde user stories voor sprint 1

Iedere user story is uitgeschreven in use-case beschrijvingen, zie bijlage VII. Vervolgens zijn deze use-case beschrijvingen omgezet naar sequentiediagrammen per use-case, zie bijlage

VIII. Alle use-cases zijn opgenomen in een use-case diagram. Hieronder is het use-case diagram van sprint 1 getoond.



Figuur 3. Use-case diagram sprint 1

Na het beschrijven en uitwerken van de user stories is het klassendiagram uitgebreid met de benodigde variabelen, klassen en methoden.

Tijdens het programmeren van UR-12 zijn er geen problemen ondervonden. Echter tijdens het programmeren van UR-32 zijn er problemen ondervonden met het omzetten van Java-objecten naar JSON door middel van het Jackson Framework. Jackson leverde problemen met het omzetten van een Java klasse naar JSON omdat deze klasse een innerklasse bevatte. Het response object FeedbackResponse leverde de problemen op met Jackson. In onderstaande afbeelding staat welke data erin FeedbackResponse staat.

```

@JsonInclude(Include.NON_NULL)
public class FeedbackResponse {

    private String ticketNumber;
    private String shortDescription;
    private Operator operator;
    private Requestor requestor;
    private FeedbackLink feedbackLinks;

    public class FeedbackLink {
        private String good;
        private String ok;
        private String bad;

        public void setGoodUrl(String url) {
            good = url;
        }

        public String getGoodUrl() {
            return good;
        }

        public void setOkUrl(String url) {
            ok = url;
        }

        public String getOkUrl() {
            return ok;
        }

        public void setBadUrl(String url) {
            bad = url;
        }

        public String getBadUrl() {
            return bad;
        }
    }

    public String getTicketNumber() {
        return ticketNumber;
    }
}

```

Afbeelding 14: Screenshot FeedbackResponse

Zoals te zien is, zijn er in deze klasse afhankelijkheden met de klasse FeedbackLink. Omdat de klasse FeedbackResponse voor meerdere doeleinde wordt gebruikt, zijn niet alle variabelen altijd gevuld. Wanneer dit het geval is, dienen deze variabelen niet getoond te worden in de response.

Met Jackson is het mogelijk om gebruik te maken van annotaties die ervoor zorgen dat variabelen die leeg zijn of een null waarde hebben niet worden getoond, echter werken deze annotaties niet goed wanneer innerklassen worden gebruikt. Een zoektocht op internet gaf een mogelijke oplossing: maak de innerklassen static. Jackson heeft problemen met het initiëren van de innerklasse doordat er geen constructor aanwezig is (Saloranta, 2010). Het resultaat is dat de variabelen, ondanks de annotatie “@JsonInclude(Include.NOT_NULL)” leeg worden getoond of met de waarde NULL, zoals is te zien in onderstaande afbeelding.

```

1 {
2   "ticketNumber": "I2016 007",
3   "shortDescription": "Feedback verzoek via Rest API call.",
4   "operator": {
5     "firstname": "Jos",
6     "lastname": "Verhaar",
7     "number": 2
8   },
9   "requestor": {
10    "firstname": "Rest API",
11    "lastname": "feedback/create",
12    "account": "Postman",
13    "email": ""
14  },
15  "feedbackLinks": {
16    "goodUrl": null,
17    "okUrl": null,
18    "badUrl": null
19  }
20 }

```

Afbeelding 15: Voorbeeld response met lege en NULL waardes

Dit probleem is opgelost door de implementatie anders te doen. Er is ervoor gekozen om de innerklasse als los object op te nemen en in de FeedbackResponse klasse op te nemen als Map. Een voordeel van deze oplossing is dat het op eenvoudige wijze mogelijk is om meer/ andere waarden toe te voegen aan het response object. Bijvoorbeeld wanneer er in een latere versie van het SFS gekozen wordt om meerdere Ratings (Good/Ok/Bad) toe te staan. Dit kan gedaan worden door het toevoegen van een nieuw key/value-pair aan de variabele.

User story UR32 is tijdens de sprint onderverdeeld in meerdere user stories:

- SFS-47 Genereren Incident in database via Rest-api. Hierbij wordt geen feedbackmail verzonden;
- SFS-48 Feedbackmail verzenden via Rest-api op basis van feedbackEncId;
- SFS-49 Opvragen gegevens van een feedbackmail via Rest-api. Hierdoor is het mogelijk om het feedback verzoek op een andere manier te verzenden dan via het SFS.

Omdat het SFS ook een mogelijkheid dient te bieden om het feedback verzoek (de feedbackmail) via een ander systeem te versturen, zoals bijvoorbeeld sociale media, is het niet wenselijk dat er direct een feedbackmail wordt verzonden door het SFS wanneer er via de Rest-api een incident wordt aangemaakt (SFS-47). Hierdoor zijn twee extra Rest-api calls nodig. Een voor het versturen van de feedbackmail via het SFS (SFS-48) en een Rest-api call om de gegevens, waaronder de feedback links, op te halen zodat deze gebruikt kunnen worden in een ander systeem. Voor het scheiden van deze twee functionaliteiten is gebruik gemaakt van het design pattern 'Seperation of Conerns'. Door beide functionaliteiten op te nemen in een eigen methode is iedere methode verantwoordelijk voor één taak. Ook bevordert dit de testbaarheid van de code.

Er is nagedacht om sociale media, zoals Facebook en Twitter direct te integreren in het SFS. Wanneer een extern systeem gebruik wil maken van een Facebook- en/of Twitter account dient het externe systeem 'vertrouwd' te worden door het sociale media account. Dit houdt in dat het SFS berichten mag verzenden namens het sociale media account. Dit brengt echter een beveiligingsissue met zich mee.

Wanneer het SFS gehackt wordt is het mogelijk om namens bedrijven berichten te plaatsen op hun officiële sociale media accounts met alle gevolgen van dien. Mede hierdoor is er in overleg met de bedrijfsmentor voor gekozen om deze integratie niet toe te passen, maar te kiezen voor de mogelijkheid om de gegevens van een feedbackmail (inclusief de feedback links) op te vragen via de Rest-api.

6.3 Sprint retrospective

De eerste sprint is afgerond met een aantal zaken die terug zijn gegaan naar de product backlog. Helaas door de uitloop van het programmeren was er geen tijd voor het opstellen, het uitvoeren en rapporteren van de tests. Tijdens de sprint bleek dat user story UR-32, “Het Feedback Systeem kan op de volgende manieren een Feedback verzoek genereren: per e-mail en via een Rest-API call.”, verder uitgebreid dient te worden. Deze onderverdeling is noodzakelijk omdat deze scheiding (e-mail versus Rest-API) er voor zorgt dat een feedbackmail niet direct verzonden dient te worden, zodat deze via een ander systeem verstuurd kan worden. Hierdoor is user story UR-32 niet in zijn geheel afgerond. In overleg met de bedrijfsmentor is de sprint gesloten en zijn deze activiteiten opgenomen in sprint 2.

De volgende functionaliteiten zijn toegevoegd aan het shippable product:

- Toevoeging van een korte omschrijving in het feedbackverzoek en de feedbackmail;
- Mogelijkheid via de Rest-API een incident op te slaan in de database;
- Mogelijkheid via de Rest-API om een feedbackmail te verzenden op basis van het feedbackEncId;
- Mogelijkheid via de Rest-API om de gegevens van een feedbackmail op te vragen, zodat het feedbackverzoek via een ander systeem kan worden verzonden.

6.4 Reflectie sprint 1

De eerste sprint is redelijk goed verlopen. Het is erg jammer dat er niet voldoende tijd bleek te zijn om alle user stories compleet af te ronden. Hierdoor moesten er een aantal user stories terug geplaatst worden in de product backlog.

De problemen waar tegen aan is gelopen tijdens het uitvoeren van de sprint, zijn allemaal op een goede manier opgelost. De keuze voor de implementatie van de Map voor de feedbacklinks zorgt er tevens voor dat de code makkelijk uit te breiden is in de toekomst.

7. Sprint 2

In de eerste sprint zijn niet alle user-stories compleet afgerond. Voor twee user-stories stond het ontwerpen en uitvoeren van de unit-tests nog open. Deze twee zijn dan ook meegenomen in deze sprint. Om de sprint aan te vullen is UR-24 toegevoegd. Dit zijn de activiteiten die in sprint 2 zijn uitgevoerd:

- UR-12 ontwerpen en uitvoeren unittests;
- UR-32 ontwerpen en uitvoeren unittests;
- UR-24: De Rest-API geeft de mogelijkheid tot het opvragen van statistieken;
- Sprint retrospective
- Reflectie sprint 2

Nummer	Omschrijving	Prioritering	Belanghebbende
UR-12	Een Feedback verzoek bestaat minimaal uit: <ul style="list-style-type: none">• Gegevens van te beoordelen hulpvraag (uniek ID en korte omschrijving);• Gegevens behandelaar (Voornaam, achternaam, e-mail en nummer);• Gegevens beoordelaar (Voornaam, achternaam, e-mail en bedrijfsnaam)	M	Teammanager support
UR32	Het SFS kan op de volgende manieren een Feedback verzoek genereren: per mail en via een REST-API call (voor het gebruik met Social Media).	M	Medewerker Support
UR24	Via de Rest-API is het mogelijk om de statistieken op te vragen van het SFS.	M	Geïnteresseerde klanten

Tabel 7: Geselecteerde user stories sprint 2

Aan het begin van de sprint is UR-24 samengevoegd met een drietal andere user-stories. De reden dat hiervoor is gekozen is dat deze user-stories een specificering zijn van UR-24. Het gaat om:

- UR07 De Rest-API kan per periode statistieken terug geven;
- UR08 De Rest-API geeft statistieken van de huidige- en voorgaande periode weer;
- UR25 Het feedback systeem kan de laatste x-aantal reacties weergeven;

Via het huidige SFS kunnen statistieken worden opgevraagd per timeframe (dag, maand en jaar). Echter zijn deze API-calls speciaal gemaakt voor het huidige dashboard dat door de afdeling support wordt gebruikt. Hierdoor zijn deze te beperkt voor het nieuwe systeem.

Via de Rest-API dienen deze statistieken los van elkaar opgevraagd te kunnen worden.

Het gaat hier om de volgende statistieken:

- het opvragen van de percentages (good, ok en bad) per timeframe (Customer satisfaction);
- het opvragen van de gegeven scores (good, ok en bad) per timeframe (Live scores);
- het opvragen van de operatorscore (aantal en percentage per score) per timeframe (Operator scores);
- het opvragen van de opmerkingen die zijn gegeven per timeframe (Remarks);

Deze gegevens worden teruggegeven over het huidige timeframe en het vorige timeframe. Als voorbeeld:

TimeFrame	Huidig	Vorig
Dag	30 oktober 2016	29 oktober 2016
Maand	oktober 2016	september 2016
Jaar	2016	2015

Tabel 8: Voorbeelden mogelijke Time Frame waarden

- Het opvragen van de x-aantal laatst gegeven reacties.
Deze laatste reacties worden weergegeven als de customer satisfaction statistieken, percentage van good, ok en bad van de laatst aantal gegeven reacties. Deze worden van het opgegeven x-aantal reacties terug gegeven en van het zelfde aantal ervoor. Als voorbeeld: De laatste 100 reacties worden opgevraagd. In het antwoord komen de laatste 100 reacties terug en de 100 daarvoor.

7.1 JUnit, Hamcrest en HSQLDB

Als test framework is er gekozen om gebruik te maken van JUnit. Als toevoeging op JUnit wordt gebruik gemaakt van Hamcrest. Dit is een uitbreiding op JUnit om de unit tests meer leesbaar te maken.

Hamcrest wordt gezien als een derde generatie matcher framework. De eerste generaties maakte gebruik van assert(logical statement). Met deze manier van noteren was het niet eenvoudig om te zien wat er nu precies werd getest. Bij de tweede generatie werden speciale methoden geïntroduceerd voor assertions, zoals assertEquals(). Helaas leidde dit tot het gebruik van veel methoden. Hamcrest maakt gebruik van de assertThat methode aangevuld met matcher expressies om de test te valideren. Hamcrest heeft tot doel om tests zo leesbaar mogelijk te maken. Dit geldt ook voor de error messages, deze zijn duidelijker doordat de expected- en actual waarden worden getoond in de error (Purpose of the Hamcrest matcher framework, Lars Vogel, 2016). In de volgende afbeelding wordt JUnit met Hamcrest vergeleken en is duidelijk te zien dat Hamcrest meer leesbaar is dan JUnit.

```
// JUnit 4 for equals check
assertEquals(expected, actual);
// Hamcrest for equals check
assertThat(actual, is(equalTo(expected)));

// JUnit 4 for not equals check
assertFalse(expected.equals(actual));
// Hamcrest for not equals check
assertThat(actual, is(not(equalTo(expected))));
```

Afbeelding 16: Leesbaarheid Hamcrest vs JUnit (Vogel, 2016).

Als database oplossing is gekozen voor HSQLDB. Dit is een runtime Java database. Beide frameworks worden standaard binnen Mailplus B.V. gebruikt.

Het opzetten van de testomgeving heeft problemen opgeleverd met de integratie van HSQLDB. De database leek te werken, maar de test bleef falen. Er kwamen echter geen duidelijke error meldingen en/of stacktraces terug waardoor het oplossen niet eenvoudig bleek. Deze problemen werden veroorzaakt door het niet kunnen vinden van een configuratie file. Toen deze file werd ingeladen werden er bruikbare error meldingen gegeven. Na het debuggen van deze error meldingen slaagde de test.

7.1.1 UR24 Via de Rest-API is het mogelijk om de statistieken op te vragen van het Feedback Systeem

De calls die geschreven dienen te worden zijn eigenlijk niets meer dan een opsplitsing van de huidige calls die gedaan worden voor (en door) het dashboard op de support afdeling. Hierdoor lijkt het een eenvoudige opgave. Helaas bleek dit minder waar.

Voor de implementatie van deze calls in combinatie met UR-8 is er veel nieuwe code geschreven. De response objecten zijn herschreven, queries zijn herschreven en vele functies zijn toegevoegd en/of gewijzigd. Deze user-stories zijn afgerond op het testen na.

7.2 Herschrijven van FeedbackController naar IncidentController

Door het gebruik van Spring MVC voor de frontend van het SFS en JAX-RS voor de Rest-API en het toepassen van 'Separation of Concerns' is het beter om de functies die betrekking hebben op de Rest-API uit de FeedbackController te halen. Dit heeft te maken met de rol van de FeedbackController. De functie van de FeedbackController is er op gericht om webpagina's te genereren en af te handelen. Hierdoor werd het lastig om ook Rest-API verzoeken hier in te verwerken. Ook uit het oogpunt van uitbreidbaarheid voor de toekomst is dit de beste keuze. Het is bijvoorbeeld de bedoeling om OAuth te implementeren. Alleen al voor deze functionaliteit is het beter om de controller die de Rest-API calls verwerkt te scheiden van de controller die de webpagina's afhandelt.

7.3 Sprint retrospective

Helaas zijn er tijdens deze sprint een aantal user stories niet geheel afgemaakt. Dit maal is de oorzaak een probleem tijdens het inrichten van de test-suite. De oplossing voor het niet werken van de HSQLDB is door de afstudeerder gevonden. Doordat deze problemen tegen het einde van de sprint pas waren opgelost was er geen tijd om de tests te implementeren en zullen deze user-stories worden meegenomen in sprint 3.

De volgende functionaliteiten zijn toegevoegd aan het shippable product:

- Via de Rest-API kunnen verschillende statistieken per timeframe worden opgevraagd:
 - Customer Satisfaction: percentages good, ok en bad per timeframe;
 - Live Scores: per gegeven feedback de score als good, ok of bad terug;
 - Operator Scores: percentages en aantallen ontvangen feedback per score;
 - Remarks: de gegeven schriftelijke opmerkingen.
- Via de Rest-API worden de statistieken teruggegeven van de huidige- en voorgaande periode.
- Via de Rest-API kunnen de x-aantal laatste reacties worden opgevraagd:
 - Deze worden weergegevens als customer satisfaction en worden over het opgevraagde aantal teruggegeven en als offset van het zelfde aantal.

7.4 Reflectie sprint 2

Tijdens sprint 2 zijn er problemen ontstaan in de implementatie van de test-suite. Gelukkig is de oorzaak gevonden en werkt de HSQLDB nu naar behoren. Het jammer is dat de unittests nog niet geïmplementeerd kunnen worden tijdens deze sprint. Er is overleg met de bedrijfsmentor voor gekozen om de test functionaliteit te verplaatsen naar een eigen sprint. Er is hiervoor gekozen zodat er tijdens deze sprint volledige focus gelegd kan worden op het testen van de functionaliteiten. Welke sprint dit wordt is nog niet besloten. Ik ben er erg trots op dat het probleem met de HSQLDB door mijzelf is opgelost. Ik had hulp ingeroepen van de bedrijfsmentor, maar uiteindelijk heb ik zelf de oplossing gevonden.

9. Sprint 4

Gedurende sprint 4 is er aan verschillende user stories gewerkt. De activiteiten van sprint 4 zijn als volgt:

- Uitwerking UR-09
- Uitwerking UR-27
- Uitwerking UR-29
- Uitwerking UR-30
- Sprint retrospective
- Reflectie sprint 4

Nummer	Omschrijving	Prioritering	Belanghebbende
UR09	Het dient mogelijk te zijn om behandelaars (operators) te beheren (toevoegen, bewerken en verwijderen) via het SFS.	S	Teammanager support
UR27	De dag-, maand- en jaarcijfers worden absoluut, in percentage en in (afgeronde) percentagepunt getoond.	S	Medewerker support
UR29	De mogelijkheid tot het tonen van grafieken met daarin het percentage good/ok/bad in vergelijking met de vorige periode.	S	Medewerker support
UR30	Bij de beoordelingen op het dagoverzicht heb je de mogelijkheid tot het zien bij welk feedback verzoek deze hoort (naam beoordelaar, bedrijfsnaam en callnummer).	S	Medewerker support

Tabel 11: User stories sprint 4

9.1 UR9

Het dient mogelijk te zijn om behandelaars (operators) te beheren (toevoegen, bewerken en verwijderen) via het SFS.

In de beginsituatie werden de operators per feedback-record opgeslagen. Dit leidde tot veel redundante data. Om het mogelijk te maken om operators te beheren is er een nieuwe tabel toegevoegd aan de database. In de volgende afbeelding staat de databasetabel Operator weergegeven.

```

CREATE TABLE `Operator` (
  `externalId` int(11) NOT NULL,
  `userId` int(11) NOT NULL,
  `encId` varchar(255) NOT NULL,
  `firstName` varchar(255) NOT NULL,
  `lastName` varchar(255) NOT NULL,
  `email` varchar(255) NOT NULL,
  `active` tinyint(1) NOT NULL DEFAULT '1',
  PRIMARY KEY (`externalId`,`userId`),
  KEY `encId` (`encId`),
  KEY `userId` (`userId`),
  FOREIGN KEY (`userId`) REFERENCES `AccessToken` (`id`)
)

```

Afbeelding 23: Tabel Operator

Zoals te zien in afbeelding 23 worden er een aantal algemene zaken opgeslagen van een operator. Naast deze zaken wordt er een referentie, via een foreign key constraint, bijgehouden naar de gebruiker (userId). Ook wordt er een status toegewezen aan een operator. Verder zijn er twee indexes toegevoegd op de velden encId en userId. De reden hiervan is dat de tabel veel op deze waarden doorzocht zal worden, door het toevoegen van deze indexes zal de database sneller opereren. Na het toevoegen van de nieuwe tabel is ook de tabel Feedback aangepast. Hier zijn de velden: operatorNumber, operatorFirstname en operatorLastname verwijderd en is het veld operatorEncId toegevoegd. Dit laatste veld heeft een foreign key constraint naar de nieuwe tabel operator. Ook is er een index gezet op het operatorEncId om het zoeken hierop sneller te maken.

Nadat de operators zijn verplaatst naar hun eigen tabel is de codebase aangepast naar de nieuwe werking. Immers zat de informatie van operators eerst in de tabel feedback. Hiervoor zijn verschillende klassen aangepast in de service- en business laag.

Om CRUD operaties toe te kunnen passen op de operators is er nieuwe functionaliteit geschreven. Hiervoor is een nieuwe controller toegevoegd welke dient als Rest-API endpoint, nieuwe klassen toegevoegd aan de business- en service laag om deze aanvragen te faciliteren. Het is mogelijk om een delete request uit te voeren, maar er wordt geen operator verwijderd uit de database. Er is gekozen om operators niet te kunnen verwijderen, doordat hier veelal feedback-records aan gekoppeld zijn. Wanneer deze records ook zouden worden verwijderd, veranderen de statistieken en dat is geen gewenste functionaliteit. Hierdoor is ervoor gekozen dat de status van de operator op inactief wordt gezet. Het operator-record blijft bestaan en hierdoor kunnen de gekoppelde feedback-records ook blijven bestaan. Een ander voordeel van deze aanpak is dat het mogelijk is om operators 'uit te zetten' wanneer deze bijvoorbeeld een andere functie gaan bekleden.

Ook is er de mogelijkheid toegevoegd om statistieken van één operator op te kunnen vragen. Het gaat hier om de operator score (aantal en percentage van de good's, ok's en bad's) per opgegeven time-frame (dag, maand, q1, q2, q3, q4 en jaar).

Tijdens de implementatie van UR-09 vond ik het lastig om te bepalen wanneer met statistieken alle operators meegenomen dienen te worden, dus de actieven en inactieven. In overleg met de product owner is er besloten om alleen inactieven operators mee te nemen in de Rest-API calls die bedoeld zijn voor analyses en deze niet mee te nemen in de calls die zijn bedoeld voor real-time statistieken voor het gebruik in dashboards. Verder liep ik aan tegen het gebruik van verkeerde annotaties in de controller. Er werd gebruik gemaakt van de annotatie @PathVariable en @RequestBody om de variabelen uit de URL te vertalen naar een Java variabele. Deze annotaties bleken uit de package org.springframework.web.bind.annotation te komen. De JAX-RS equivalent is de annotatie @PathParam welke uit de package javax.ws.rs.PathParam komt.

9.2 UR27

De dag-, maand- en jaarcijfers worden absoluut, in percentage en in (afgeronde) percentagepunt getoond.

De user story impliceert dat de percentages in twee waarden worden teruggegeven: in gehele percentages en in procentpunten, het getal achter de komma. Voor de implementatie is gekozen om de percentages om te zetten naar een Double en deze af te ronden op 3 decimalen. De reden dat hiervoor is gekozen is dat het afronden naar hele percentages of naar 2 decimalen altijd in de client applicatie mogelijk is. Door het kiezen voor het meegeven van 3 decimalen (en dus maar 1 percentage) kun je alle kanten op. Hierdoor is deze implementatie breed inzetbaar. Met deze oplossing is het ook mogelijk om verschillen tussen percentages weer te geven. Dit kan bijvoorbeeld worden gebruikt om een stijging of daling aan te geven van de resultaten op een bepaald time-frame.

9.3 UR29

De mogelijkheid tot het tonen van grafieken met daarin het percentage good/ok/bad in vergelijking met de vorige periode.

Om deze user story een betere invulling te geven met betrekking tot de vergelijking met de vorige periode ben ik in overleg gegaan met de opdrachtgever. Hij is immers de aangewezen persoon om hier meer uitleg over te geven. Uit dit overleg is gekomen dat er met de vorige periode dezelfde periode het jaar ervoor wordt bedoeld. Als voorbeeld wanneer er statistieken voor de huidige maand (december 2016) worden opgevraagd, dienen ook dezelfde statistieken van dezelfde periode het jaar ervoor (december 2015) te worden getoond. Hierdoor wordt het mogelijk om deze twee perioden met elkaar te vergelijken in bijvoorbeeld een grafiek.

Voor de implementatie van deze functionaliteit is er gezocht naar een gestandaardiseerde oplossing voor het aanleveren van data voor het gebruik in grafieken. Helaas is deze niet gevonden. Er zijn meerdere libraries beschikbaar voor het weergeven van data in verschillende grafieken, echter heeft iedere library eigen wensen in hoe de data aangeleverd dient te worden. Hierdoor is ervoor gekozen om de data op een zo eenvoudig mogelijke manier terug te geven. In de Rest-API documentatie⁷ staan de responses beschreven.

De moeilijkheid van deze user story zat hem vooral in het aanleveren en het verkrijgen van de juiste data uit het SFS. Voor deze functionaliteit zijn aanpassingen gedaan in meerdere klassen in de business- en service laag. Gelukkig kon er gebruik worden gemaakt van eerder geschreven functionaliteit. Veel database functies voor het verkrijgen van statistieken op basis van een start- en einddatum waren reeds geschreven.

Doordat veel van deze functies reeds aanwezig waren is er gekozen om deze functies meerdere malen aan te roepen, in plaats van aparte functies te schrijven die het gewenste resultaat teruggaven. In onderstaande afbeeldingen (afbeelding 24 en 25) staat een implementatie hiervan weergegeven.

⁷ Zie Bijlage X: Rest-API documentatie

```

switch( timeframe ) {
    case DAY:
        break;
    case MONTH:
        operatorScore = feedbackService.getOperatorScoreByDate(userId, startDate, endDate);
        result.put(getDateToString(startDate, TimeFrame.MONTH), getOperatorScores(operatorScore));
        break;
    case Q1:
        for( int i=0; i < 3; i++ ) {
            cal = dateService.getNow();
            Date dateQ1 = startDate;
            cal.setTime(dateQ1);
            cal.set(Calendar.MONTH, i);
            dateQ1 = cal.getTime();
            Date endDateQ1 = calculateOffsetDateFromTimeFrame(TimeFrame.MONTH, dateQ1, 1);
            operatorScore = feedbackService.getOperatorScoreByDate(userId, dateQ1, endDateQ1);
            result.put(getDateToString(dateQ1, TimeFrame.MONTH), getOperatorScores(operatorScore));
            cal.clear();
        }
        break;
    case Q2:
        for( int i=3; i < 6; i++ ) {
            cal = dateService.getNow();
            Date dateQ2 = startDate;
            cal.setTime(dateQ2);
            cal.set(Calendar.MONTH, i);
            dateQ2 = cal.getTime();
            Date endDateQ2 = calculateOffsetDateFromTimeFrame(TimeFrame.MONTH, dateQ2, 1);
            operatorScore = feedbackService.getOperatorScoreByDate(userId, dateQ2, endDateQ2);
            result.put(getDateToString(dateQ2, TimeFrame.MONTH), getOperatorScores(operatorScore));
            cal.clear();
        }
        break;
    case Q3:
        for( int i=6; i < 9; i++ ) {
            cal = dateService.getNow();
            Date dateQ3 = startDate;
            cal.setTime(dateQ3);
            cal.set(Calendar.MONTH, i);
            dateQ3 = cal.getTime();
            Date endDateQ3 = calculateOffsetDateFromTimeFrame(TimeFrame.MONTH, dateQ3, 1);
            operatorScore = feedbackService.getOperatorScoreByDate(userId, dateQ3, endDateQ3);
            result.put(getDateToString(dateQ3, TimeFrame.MONTH), getOperatorScores(operatorScore));
            cal.clear();
        }
        break;
}

```

Afbeelding 24: Implementatie opvragen statistieken voor verschillende TimeFrames 1/2

```

    case Q4:
        for( int i=9; i < 12; i++ ) {
            cal = dateService.getNow();
            Date dateQ4 = startDate;
            cal.setTime(dateQ4);
            cal.set(Calendar.MONTH, i);
            dateQ4 = cal.getTime();
            Date endDateQ4 = calculateOffsetDateFromTimeFrame(TimeFrame.MONTH, dateQ4, 1);
            operatorScore = feedbackService.getOperatorScoreByDate(userId, dateQ4, endDateQ4);
            result.put(getDateToString(dateQ4, TimeFrame.MONTH), getOperatorScores(operatorScore));
            cal.clear();
        }
        break;
    case YEAR:
        for( int i=0; i < 12; i++ ) {
            cal = dateService.getNow();
            Date date = startDate;
            cal.setTime(date);
            cal.set(Calendar.MONTH, i);
            date = cal.getTime();
            Date dateEnd = calculateOffsetDateFromTimeFrame(TimeFrame.MONTH, date, 1);
            operatorScore = feedbackService.getOperatorScoreByDate(userId, date, dateEnd);
            result.put(getDateToString(date, TimeFrame.MONTH), getOperatorScores(operatorScore));
            cal.clear();
        }
        break;
    default:
        break;
}

```

Afbeelding 25: Implementatie opvragen statistieken voor verschillende TimeFrames 2/2

Zoals is te zien is ervoor gekozen om het timeframe DAY niet te ondersteunen. De reden hiervan is dat dagen ieder jaar een andere dag in de week zijn. Hierdoor zullen er dagen zijn die in de huidige periode een werkdag zijn en een weekenddag zijn in de vorige periode en andersom. Dit zal resulteren in extreme verschillen op deze momenten. Het timeframe MONTH geeft de waarde (in dit geval de operator score) terug van de start- en einddatum voor de gebruiker. Voor de kwartalen (Q1, Q2, Q3 en Q4) en de jaren gaat het iets anders. Deze resultaten worden in maanden terug gegeven.

Bij de kwartalen wordt er door de drie maanden van dat kwartaal gegaan met behulp van een for-loop. Vervolgens worden de juiste start- en einddatums berekend en worden de operator scores opgehaald. Daarna wordt aan de hand van de startdatum de maand opgehaald. Deze maand wordt gebruikt als key in het resultaat en als value wordt de opgehaalde operator score toegevoegd.

Bij het timeframe YEAR gebeurt hetzelfde als bij de kwartalen alleen wordt dit nu voor alle 12 maanden van een jaar gedaan.

De bovenstaande code zal voor ieder Rest-API request twee maal worden aangeroepen. De eerste maal voor de huidige periode en de tweede maal voor de vorige periode. De resultaten worden in een bovenliggende methode samengevoegd.

9.4 UR30

Bij de beoordelingen op het dagoverzicht heb je de mogelijkheid tot het zien bij welk feedback verzoek deze hoort (naam beoordelaar, bedrijfsnaam en callnummer).

In de huidige implementatie van het Dashboard worden live-scores getoond op het dagoverzicht. Dit zijn in totaal 100 smileys welke aangeven welke rating er door de beoordelaars is gegeven. De kleuren van de smileys representeren de rating. Het komt wel eens voor dat mensen een 'Ok' of 'Bad' geven zonder hier een opmerking bij te plaatsen. Medewerkers van de afdeling support willen graag weten welke beoordelaar dit is geweest en voor welk TopDesk ticket deze beoordeling geldt. Momenteel moet dit via de database, wat omslachtig is. In het nieuwe Support Feedback Systeem is deze functionaliteit uitgebreid met informatie van het incident en de beoordelaar. Hierdoor is het mogelijk om deze gegevens (bijvoorbeeld bij hover over een live-score) te tonen. Hierdoor kun je op eenvoudige wijze, zonder toegang tot de database, zien wie voor welk incident welke beoordeling heeft gegeven.

De implementatie van deze functionaliteit heeft ertoe geleid dat het IncidentResponse object is uitgebreid. Hiervoor is een nieuwe klasse LiveScore aangemaakt welke de benodigde informatie bevat. Onderstaande afbeelding is hiervan een screenshot te zien.

```
@JsonInclude(Include.NON_EMPTY)
public class LiveScore {

    private String rating;
    private String ticketNumber;
    private Requestor requestor;

    public class Requestor {
        private String firstname;
        private String lastname;
        private String account;
    }
}
```

Afbeelding 26: Screenshot LiveScore zonder de Getters en Setters

Verder zijn er voor deze implementatie een aantal aanpassingen gedaan aan de business laag.

9.5 Aanpassingen buiten de sprint backlog

Tijdens de uitvoering van de sprint zijn nog een aantal kleinere aanpassingen doorgevoerd. Deze aanpassingen worden apart genoemd omdat het belangrijke aanpassingen zijn die in het verlengde liggen van de uitgevoerde werkzaamheden binnen de huidige sprint.

9.5.1 Introductie Universally Unique Identifier

Binnen het SFS wordt op verschillende plekken gebruik gemaakt van een encrypted id (encId). Deze waarden worden onder andere gebruikt als identifier in de database. Een voorbeeld hiervan is het operatorEncId dat in de tabel Feedback wordt gebruikt. Ook het feedbackEncId zelf is een goed voorbeeld. Ook worden deze waarden naar gebruikt in de feedbacklinks en om bijvoorbeeld statistieken voor een bepaalde operator op te vragen. Voor deze externe implementaties wil je geen gebruik maken van echte database keys, maar van een uniek alternatief (encrypted id: encId). In de huidige implementatie van het SFS was voor het genereren van encId's een eigen implementatie geschreven. Dit was voor de opzet met één gebruiker een prima oplossing. Echter voor het nieuwe SFS kan dit problemen gaan opleveren. Doordat in het nieuwe SFS op meerdere plekken (ook in meerdere tabellen) gebruik wordt gemaakt van encrypted id's dient de controle op het uniek zijn aangepast te worden in de huidige opzet. Ook het gebruik van een standaard oplossing (UUID) bevordert de code. Hierdoor is er gekozen om de implementatie van de encId's door het gehele SFS om te zetten naar UUID.

9.5.2 Toevoeging kwartalen als timeframe

Doordat vele bedrijven (intern) per kwartaal rapporteren, was het van belang dat het SFS ook deze periodes ging ondersteunen. De toevoeging van de kwartalen heeft vooral bij de statistieken een flinke impact gehad en geresulteerd in extra code.

9.5.3. E-mail validatie naar validator klasse

Doordat e-mail validatie nodig is in de operator- (bij een operator kan een e-mailadres worden opgeslagen) en bij de incidentmanager (een beoordelaar heeft een e-mailadres om feedback verzoeken naar te sturen) is er gekozen om de validatie van e-mailadressen te verplaatsen naar een static validator klasse. De reden hiervan is dat deze validatie op meerdere plekken wordt gebruikt en hierdoor maar op een plek onderhouden hoeft te worden.

9.6 Sprint retrospective

Tijdens deze sprint zijn vele regels code geschreven. Zeker voor de user stories “UR9: Het dient mogelijk te zijn om behandelaars (operators) te beheren (toevoegen, bewerken en verwijderen) via het SFS.” en “UR29: De mogelijkheid tot het tonen van grafieken met daarin het percentage good/ok/bad in vergelijking met de vorige periode.”. Naast de problemen met de annotaties zijn er geen grote issues geweest tijdens de implementatie.

De volgende functionaliteiten zijn toegevoegd aan het shippable product:

- Het is mogelijk om via de Rest-API operators toe te voegen, te wijzigen, op te vragen en te verwijderen (inactief te maken);

- De statistieken worden aangegeven met een aantal en de percentages worden met drie cijfers achter de komma teruggegeven;
- De Rest-API kan statistieken teruggeven over de huidige periode en dezelfde periode het voorgaande jaar;
- De Rest-API geeft bij live scores aan welk incident nummer, bedrijfsnaam en beoordelaar hierbij hoort.

9.7 Reflectie sprint 4

Sprint 4 was een volle sprint, met mooie en uitdagende functionaliteiten. Voor de implementatie van deze functionaliteiten werd er veel aanspraak gedaan op mijn kennis van het SFS en creativiteit.

10. Sprint 5

Gedurende de vijfde sprint is er gewerkt aan het testen van het SFS. Vanwege eerdere problemen met de testopstelling is het testen opgenomen in een aparte sprint. Voor het testen van het SFS is een testplan opgesteld. De tests uit het testplan richten zich voornamelijk op de Systeemtest. De volgende activiteiten zijn uitgevoerd gedurende deze sprint:

- Opstellen testplan
 - Testbasis
 - Teststrategie
 - Testrisicoanalyse
 - Bepalen testgevallen (logisch en fysiek)
- Geautomatiseerd testen in JUnit
- Uitvoering van de testgevallen
- Retrospective
- Reflectie sprint 5

10.1 Opstellen testplan

Voor het testen van de code van het SFS is zoveel mogelijk TESTGOAL gevolgd, zo ook voor het opstellen van het testplan. Echter is deze methodiek niet volgens de letter gevolgd.

Het begroten en plannen is een voorbeeld van een deel dat niet is uitgevoerd. Hiervoor is gekozen, omdat het maken van een begroting en uitgebreide planning voor de grote en de opzet van de afstudeeropdracht overbodig is geacht. Verder is het testplan niet onderverdeeld in een master testplan en detail testplannen. Ook in deze keuze speelt de grootte van de afstudeeropdracht een rol. De afstudeeropdracht is niet gelijk aan een opdracht van een grote bedrijfskritische applicatie. Ook het aantal projectleden is dermate anders dat het maken van een master testplan en verschillende detail testplannen wordt gezien als een papieren tijger. Het kost veel werk en inspanning, maar levert weinig voordelen op. Hierdoor is er gekozen om beide testplannen in één plan op te nemen.

10.1.1 testbasis

Tijdens het opstellen van de testbasis het system under test (hierna: SUT) bepaald. Niet het gehele SFS wordt namelijk getest. Er is gekozen om alleen de Rest-API kant te testen, omdat tijdens de uitvoering van de afstudeeropdracht hier toevoegingen aan gedaan zijn. In het testplan⁸ staat een voorbeeld systeemarchitectuur weergegeven waarin ook duidelijk het SUT wordt aangeduid.

10.1.2 teststrategie

In de teststrategie wordt aangeduid hoe de uit te voeren testen bijdragen aan het beoogde resultaat. Er staat uitvoerig beschreven hoe de testen tot stand zijn gekomen. Deze stappen zijn later in het testplan dieper beschreven.

10.1.3 testrisicoanalyse

Er is gekozen om een 1-D testrisicoanalyse te maken in plaats van een 2-D risicoanalyse. Hier is voor gekozen omdat het SFS niet een bedrijfskritische applicatie is. De 2-D risicoanalyse kijkt op een andere manier naar de risico's. Een 1-D risicoanalyse neemt als basis de systeem-

⁸ Bijlage IX: Testplan

of requirements specificatie, de 2-D risicoanalyse neemt het beoogde resultaat als uitgangspunt. Hierdoor is het mogelijk dat er risico's worden gevonden die niet in de specificaties zijn genoemd.

De testboom is opgesteld door de functionaliteiten van het SFS onder te verdelen in hoofd- en subfuncties. Als basis voor de hoofdfuncties is gekozen om de controllers als basis te nemen. Doordat deze logisch zijn ingedeeld geven de controllers een duidelijke scheiding weer binnen het SFS. Iedere functie in de controllers zijn als subfunctie aangewezen. Voor deze functionaliteiten is vervolgens het relatief belang bepaald. Dit is gedaan door de student en de productowner. Beide personen zijn nauw betrokken geweest bij de uitvoering van de afstudeeropdracht en zijn daarom gekozen. Beide hadden maximaal 69 punten te verdelen. Het maximaal aantal is vastgesteld aan de hand van de volgende formule: [aantal subfuncties] * 3. Dit resulteert in de som $23 * 3 = 69$. Het verdelen van de punten kan in vier verschillende gradaties die aangeduid worden met de getallen: 9, 5, 3 en 1. Hoe hoger het getal hoe belangrijker de functie is voor de werking van het SFS.

OAuth heeft van mij negen punten gekregen, het hoogst mogelijk. Wanneer de implementatie van OAuth niet goed werkt kunnen gebruikers geen gebruik maken van de Rest-API. Hierdoor is deze functionaliteit van groot belang voor het SFS. Het toevoegen en updaten van een behandelaar (operator), het toevoegen van een gebruiker, het toevoegen, updaten, verzenden en opvragen van een incident (feedback) hebben de score vijf gekregen. Deze functionaliteiten zijn core functionaliteiten van het SFS. Wanneer deze niet naar behoren werken werkt het SFS niet goed. De andere sub functionaliteiten hebben drie of één punt gekregen. In bijlage IX is het gehele overzicht te zien.

10.1.4 bepalen testtechnieken

Na het bepalen van het relatieve belang zijn de sub functionaliteiten onderverdeeld in risicocategorieën. Nadat de risicocategorieën zijn verdeeld is bepaald met welke testtechnieken en test diepte de functies worden getest. Dit is terug te vinden in onderstaande tabel. Eerst wordt de risicocategorie genoemd, de kolommen ernaast geven de test domeinen: Invoer en Functionele afhandeling genoemd.

Risicocategorie	Invoer	Functionele afhandeling
Kritisch	Syntax: (niet) valide	AT branche coverage + CE
Hoog	Syntax: (niet) valide	AT branche coverage
Midden	Syntax: (niet) valide	AT statement coverage
Laag	Syntax: valide	

[Geselecteerde testtechnieken per test domein en risicocategorie]

Voor het bepalen welke testtechnieken worden gebruikt is als basis de tabel 12.1 genomen uit het TestGoal boek [blz 159]. In deze tabel staat een voorbeeld van een technieken matrix afgebeeld. In deze matrix staat per domein en per test sterkte testtechnieken genoemd. Het is niet efficiënt om alle testtechnieken in te gaan zetten voor iedere functie die getest gaat worden. De keuze kan bepaald worden op basis van:

- type fout dat wordt verwacht
- hoe erg is het als deze fout in de productieomgeving zit
- welke informatie is beschikbaar om de test op te baseren

De zwaarte van de gekozen technieken loopt op met de risicocategorie waarvoor deze gaan gelden. Het verschil tussen Midden en Hoog is dat de test diepte van de algoritmetest bij de categorie Hoog groter is.

10.1.5 bepalen logische testgevallen

Voor het bepalen van de logische testgevallen zijn per testtechniek en per functionaliteit de logische testgevallen opgesteld. Voor iedere testtechniek is een stappenplan doorlopen.

Voor de syntax testen zijn de volgende stappen doorlopen:

1. Bepaal voor elk te testen data-element (attribuut) de lengte en de restricties
2. Bepaal de geldige en ongeldige waarden voor elk attribuut
3. Ontwerp testgevallen op basis van deze waarden

Voor de algoritme testen zijn de volgende stappen doorlopen:

4. Vaststellen beslispunten in het procesdiagram
5. Vaststellen van de test diepte
6. Vaststellen van de paden door het procesdiagram
7. Definieer testscenario's

Voor de cause effect graphing testen zijn de volgende stappen doorlopen:

1. Bepaal condities en acties
2. Definieer beslissingstabel
3. Reduceer het aantal testgevallen
4. Definieer testgevallen

10.1.6 bepalen fysieke testgevallen

Aan de hand van de logische testgevallen zijn de fysieke testgevallen opgesteld. Deze zijn per testtechniek verder uitgewerkt volgens een vast stramien. Dit stramien zorgt ervoor dat de opzet voor ieder testgeval gelijk is en dat deze te traceren zijn tot de testboom.

10.1.7 Incident-Updaten en Incident-Toevoegen

Tijdens het schrijven van de logische test gevallen is naar voren gekomen dat het los van elkaar testen van Incident-Updaten en Incident-Toevoegen niet mogelijk is. Dit komt door de functie processIncident die in beide methoden wordt gebruikt. Deze functie dient echter in tact te blijven om de integratie met TopDesk te behouden.

10.2 Geautomatiseerd testen in JUnit

Aan de hand van de hoofdfunctionaliteit Operator wordt het geautomatiseerd testen beschreven. Voor de subfuncties zijn de volgende testgevallen gemaakt: SYN.T01, SYN.T2, SYN.T3, SYN.T11, SYN.T19, SYN.T20, AT.T03, AT.T11 en AT.T12.

10.2.1 JUnit

Als test framework is gebruik gemaakt van JUnit. JUnit maakt het mogelijk om testen te automatiseren met code. Het grote voordeel hiervan is dat de tests hergebruikt kunnen worden.

JUnit maakt gebruik van annotaties. Een van deze annotaties is `@BeforeClass`. Wanneer deze annotatie wordt gebruikt wordt de methode `setup()` eenmalig uitgevoerd bij het initialiseren van de test klasse. In onderstaande afbeelding is een voorbeeld te zien van het gebruik van `@BeforeClass`.

```

@BeforeClass
public static void setUp() throws Exception {

    setUpObjects();
    operatorService = mock(OperatorService.class);
    when(operatorService.getOperatorByEncId(operatorDao1.getEncId(), true)).thenReturn(operatorDao1);
    when(operatorService.getOperatorByEncId(operatorDao1.getEncId(), false)).thenReturn(operatorDao1);
    when(operatorService.getOperatorByEncId(operatorDao2.getEncId(), false)).thenReturn(operatorDao2);
    when(operatorService.getOperatorByEncId(operatorDao2.getEncId(), true)).thenReturn(null);
    when(operatorService.getOperatorByEncId("456-dfg-67j", false)).thenReturn(null);

    when(operatorService.getOperator(1100, 1, false)).thenReturn(operatorDao1);
    when(operatorService.getOperator(1000, 1, false)).thenReturn(null);
    when(operatorService.getOperator(1100, 3, false)).thenReturn(operatorDao3);

    operatorList = Arrays.asList(operatorDao1, operatorDao2);
    when(operatorService.getOperatorList(1100)).thenReturn(operatorList);

    incidentManager = mock(IncidentManager.class);
    when(incidentManager.getOperatorScoreByTimeFrame(operatorDao2.getEncId(), TimeFrame.DAY))
        .thenReturn(Arrays.asList(osDay));
    when(incidentManager.getOperatorScoreByTimeFrame(operatorDao2.getEncId(), TimeFrame.MONTH))
        .thenReturn(Arrays.asList(osMonth));
    when(incidentManager.getOperatorScoreByTimeFrame(operatorDao2.getEncId(), TimeFrame.Q1))
        .thenReturn(Arrays.asList(osQ1));
    when(incidentManager.getOperatorScoreByTimeFrame(operatorDao2.getEncId(), TimeFrame.Q2))
        .thenReturn(Arrays.asList(osQ2));
    when(incidentManager.getOperatorScoreByTimeFrame(operatorDao2.getEncId(), TimeFrame.Q3))
        .thenReturn(Arrays.asList(osQ4));
    when(incidentManager.getOperatorScoreByTimeFrame(operatorDao2.getEncId(), TimeFrame.Q4))
        .thenReturn(Arrays.asList(osQ4));
    when(incidentManager.getOperatorScoreByTimeFrame(operatorDao2.getEncId(), TimeFrame.YEAR))
        .thenReturn(Arrays.asList(osYear));

    operatorManager = new OperatorManagerImpl(operatorService, incidentManager);
}

```

Afbeelding 27: Voorbeeld @BeforeClass

10.2.2 Mockito

Voor het implementeren van de UT is gebruik gemaakt van het mocking framework Mockito. Mockito maakt het mogelijk om afhankelijkheden van een klasse te imiteren. Hierdoor kan de klasse geïsoleerd worden getest. In onderstaande afbeelding is te zien dat voor de operator testen de operatorService en incidentManager worden vervangen door mock objecten door het gebruik van de annotatie @Mock.

```

@Mock
private static OperatorService operatorService;

@Mock
private static IncidentManager incidentManager;

```

Afbeelding 28: Voorbeeld @Mock

In afbeelding 28 is te zien hoe de Mocks voor de operatorService en incidentManager worden aangemaakt en vervolgens aan de operatorManager (Klasse die wordt getest) worden toegevoegd. Dit wordt gedaan door de volgende regels code:

```

operatorService = mock(OperatorService.class);
incidentManager = mock(IncidentManager.class);
operatorManager = new OperatorManagerImpl(operatorService, incidentManager);

```

Mockito maakt het ook mogelijk om functies die tijdens runtime worden aangeroepen van een van de gemockte objecten te stubben. Studden houdt in dat er voor een functie wordt aangegeven met welke parameters deze wordt aangeroepen en welke return value er teruggegeven moet worden. Hiermee wordt bepaald hoe de gemockte objecten reageren. In onderstaand voorbeeld is te zien hoe de functie 'getOperatorScoreByTimeFrame' van het gemockte object incidentManager wordt nagebootst:

```
when(incidentManager.getOperatorScoreByTimeFrame(operatorDao2.getEncId(), TimeFrame.DAY))
    .thenReturn(Arrays.asList(osDay));
```

Je kunt hier zien dat wanneer deze functie wordt aangeroepen met het encId van object operatorDao2 en TimeFrame.DAY geantwoord wordt met een List(osDay). In afbeelding 27 zijn alle gestubde methoden te zien.

10.2.3 Hamcrest

JUnit maakt het mogelijk om geautomatiseerd testen te schrijven en deze herhaaldelijk uit te voeren. Met de toevoeging van Hamcrest worden deze tests beter leesbaar. In onderstaande afbeelding is de test SYN.T02 geschreven in Hamcrest en default JUnit. Hier is te zien dat Hamcrest de test beter leesbaar maakt.

```

/*****
***   getOperatorListTest | SYN.T02   ***
*****/
@Test
public void getOperatorListTest() {
    List<OperatorResponse> result;

    // Using Hamcrest
    result = operatorManager.getOperatorList(1100);
    assertThat(result.size(), is(2));
    assertThat(result.get(0).getEncId(), is("345-0-09df-3209-09df-3209-09df-3209"));
    assertThat(result.get(1).getEncId(), is("09889-abd7-7882-abd7-7882-abd7-7882"));

    // Using default JUnit
    assertEquals(2, result.size());
    assertEquals("345-0-09df-3209-09df-3209-09df-3209", result.get(0).getEncId());
    assertEquals("09889-abd7-7882-abd7-7882-abd7-7882", result.get(1).getEncId());
}

```

Afbeelding 29: Voorbeeld Hamcrest assertThat vs JUnit assertEquals

10.2.4 HSQLDB

In deze test wordt er geen gebruik gemaakt van een database. De reden hiervan is dat de operatorService, welke wel gebruikt maakt van de database, wordt gemocked. In de UT van de operatorService wordt wel gebruik gemaakt van HSQLDB. In onderstaande afbeelding is te zien dat de operatorService hier niet wordt gemocked. Dit komt doordat dit de klasse is die getest wordt. In de @BeforeClass wordt de HSQLDB opgezet en gevuld met tabellen en testdata. In de @AfterClass wordt de database weer opgeruimd.

```

public class OperatorServiceTest {

    @Autowired
    OperatorService operatorService;

    private static EmbeddedDatabase db;

    @BeforeClass
    public static void setUp() throws Exception {
        db = new EmbeddedDatabaseBuilder()
            .setType(EmbeddedDatabaseType.HSQL)
            .addScript("sql/create-db.sql")
            .addScript("sql/insert-data.sql")
            .build();
    }

    @AfterClass
    public static void tearDown() throws Exception {
        db.shutdown();
    }
}

```

Afbeelding 30: Voorbeeld HSQLB

Dit betekent dat de database alleen gedurende het uitvoeren van deze test aanwezig is. Voor een volgende test zal deze opnieuw worden opgebouwd. Het grote voordeel hiervan is dat voor iedere testklasse je een schone database hebt om tegen te testen.

Verder gaat het schrijven van de test hetzelfde als eerder. Tijdens runtime wordt de test database bevraagd in plaats van de echte database, zoals te zien is in onderstaande afbeelding.

```

@Test
public void getOperatorList() {
    List<OperatorDao> result = new ArrayList<OperatorDao>();
    Integer userId = 1000;

    result = operatorService.getOperatorList(userId);
    assertThat(result.size(), is(3));

    assertThat(result.get(0).getFirstname(), is("Jos"));
    assertThat(result.get(1).getFirstname(), is("Paul"));
    assertThat(result.get(2).getFirstname(), is("Ralph"));
}

```

Afbeelding 31: Voorbeeld getOperatorList in de operatorServiceTest

10.3. Uitwerking Operator-Opvragen

Aan de hand van de subfunctie operator-opvragen wordt het gehele testproces doorlopen, van logische testgevallen tot het uitvoeren van de UT.

10.3.1 Logische testgevallen

Omdat deze subfunctie de risicocategorie midden heeft gekregen dienen er Syntax testen en Algoritme Testen met statement coverage uitgevoerd te worden. Hiervan zijn de volgende logische testgevallen opgesteld:

	Geval 1	Geval 2	Geval 3	Geval 4	Geval 5
operatorEncId	35 Char	> 35 Char	< 35 Char	""	NULL
Resultaat	Geen error	Error	Error	Error	Error

Afbeelding 32: Uitwerking logische testgevallen operator-opvragen

10.3.2 Fysieke testgevallen

De logische testgevallen zijn omgezet naar de volgende fysieke testgevallen:

Id	SYN.T11-1				
Doel van de test	Verifieer dat de gegevens van een valide operator worden opgehaald.				
Preconditie	Een valide operator is aanwezig in het systeem.				
Postconditie	De response bevat de gegevens van de operator.				
Verwijzing testboom	Operator-Opvragen				
Testacties(s)	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call GET <code>"/operator/{operatorEncId}"</code> wordt gedaan met <code>"09889-abd7-7882-abd7-7882"</code> als operatorEncId	De response bevat de gegevens van de opgevraagde valide operator. <pre>operatorDao2 = new OperatorDao(); operatorDao2.setExternalId(2); operatorDao2.setFirstname("Operator 2"); operatorDao2.setLastname("Lastname 2"); operatorDao2.setEncId("09889-abd7-7882-abd7-7882")</pre>		

			09889-abd7-7882-abd7-7882-abd7-7882"); operatorDao2.setEmail("oper@tor2.nl"); operatorDao2.setUserId(1200); operatorDao2.setActive(0);		
--	--	--	---	--	--

Id	SYN.T11-2				
Doel van de test	Verifieer dat er geen gegevens van een valide operator worden opgehaald.				
Preconditie	Een valide operator is aanwezig in het systeem.				
Postconditie	De response bevat geen gegevens van de operator.				
Verwijzing testboom	Operator-Opvragen				
Testactie(s)					
	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call GET "/operator/{operatorEncId}" wordt gedaan met "dfg-67j-456-dfg-67j-456-dfg-67j-456-dfg-67j" als operatorEncId	De response bevat de message met de tekst "OperatorEncId is not valid!"		

Id	SYN.T11-3				
Doel van de test	Verifieer dat er geen gegevens van een valide operator worden opgehaald.				
Preconditie	Een valide operator is aanwezig in het systeem.				
Postconditie	De response bevat geen gegevens van de operator.				

Verwijzing testboom	Operator-Opvragen				
Testactie(s)					
	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call GET “/operator/{operatorEncId}” wordt gedaan met “456-dfg-67j” als operatorEncId	De response bevat de message met de tekst “OperatorEncId is not valid!”		

Id	SYN.T11-4				
Doel van de test	Verifieer dat er geen gegevens van een valide operator worden opgehaald.				
Preconditie	Een valide operator is aanwezig in het systeem.				
Postconditie	De response bevat geen gegevens van de operator.				
Verwijzing testboom	Operator-Opvragen				
Testactie(s)					
	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call GET “/operator/{operatorEncId}” wordt gedaan met “” als operatorEncId	De response bevat de message met de tekst “OperatorEncId is not valid!”		

Id	SYN.T11-5				
Doel van de test	Verifieer dat er geen gegevens van een valide operator worden opgehaald.				
Preconditie	Een valide operator is aanwezig in het systeem.				

Postconditie	De response bevat geen gegevens van de operator.				
Verwijzing testboom	Operator-Opvragen				
Testactie(s)					
	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call GET "/operator/{operatorEncId}" wordt gedaan met NULL als operatorEncId	De response bevat de message met de tekst "OperatorEncId is not valid!"	<pre> if(operatorEncId == null) { logger.error(errorText + "EncId is NULL!"); return false; } </pre>	

Id	AT.T03-1				
Doel van de test	Succesvol verkrijgen van een Operator				
Preconditie	Er is een operator aanwezig in het systeem.				
Postconditie	De response bevat de gegevens van de opgevraagde operator.				
Paden	1-3-4				
Verwijzing testboom	Operator-Opvragen				
Testactie(s)					
	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	getOperator wordt aangeroepen met als	De gegevens van de opgevraagde operator worden teruggegeven. operatorDao2.setExternalId(2); operatorDao2.setFirstname(")		

	operatorEncId "09889-abd7-7882-abd7-7882-abd7-7882" en userId 1000.	Operator 2"); operatorDao2.setLastname("Lastname 2"); operatorDao2.setEncId("09889-abd7-7882-abd7-7882-abd7-7882"); operatorDao2.setEmail("operator2.nl"); operatorDao2.setUserId(1200); operatorDao2.setActive(0);		
--	--	--	--	--

Id	AT.T03-2				
Doel van de test	De validatie van het operatorEncId geeft een foutmelding, omdat deze niet valide is.				
Preconditie	Er is een operator aanwezig in het systeem.				
Postconditie	De response bevat een foutmelding				
Paden	2-5				
Verwijzing testboom	Operator-Opvragen				
Testactie(s)	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	getOperator wordt aangeroepen met als operatorEncId "" en userId 1000.	De response bevat een message met de tekst: "OperatorEncId is not valid!"		

10.3.3 Implementatie fysieke testgevallen

In onderstaande afbeeldingen zijn screenshots opgenomen van de implementatie van de fysieke testgevallen. Bij de testgevallen is met comments bijgehouden om welk fysiek testgeval het gaat. Dit laat goed de traceability zien.

```

/*****
***   getOperatorTest | SYN.T11 & AT.T03   ***
*****/
@Test
public void getOperatorTest() {
    OperatorResponse operatorResponse = null;

    //SYN.T11-1
    String operatorEncId = operatorDao2.getEncId();
    operatorResponse = operatorManager.getOperator(operatorEncId);
    assertThat(operatorResponse, hasProperty("message", equalTo(null)));
    assertThat(operatorResponse.getExternalId(), is(operatorDao2.getExternalId().toString()));
    assertThat(operatorResponse.getEncId(), is(operatorDao2.getEncId()));
    assertThat(operatorResponse.getFirstName(), is(operatorDao2.getFirstName()));
    assertThat(operatorResponse.getLastName(), is(operatorDao2.getLastName()));
    assertThat(operatorResponse.getEmail(), is(operatorDao2.getEmail()));
    assertThat(operatorResponse.getActive(), is(Boolean.FALSE));

    //SYN.T11-2
    operatorEncId = "dfg-67j-456-dfg-67j-456-dfg-67j-456-dfg-67j";
    operatorResponse = operatorManager.getOperator(operatorEncId);
    assertThat(operatorResponse.getMessage(), is("OperatorEncId is not valid!"));
    assertThat(operatorResponse, hasProperty("externalId", equalTo(null)));
    assertThat(operatorResponse, hasProperty("encId", equalTo(null)));
    assertThat(operatorResponse, hasProperty("firstname", equalTo(null)));
    assertThat(operatorResponse, hasProperty("lastname", equalTo(null)));
    assertThat(operatorResponse, hasProperty("email", equalTo(null)));
    assertThat(operatorResponse, hasProperty("active", equalTo(null)));
    operatorResponse = null;

    //SYN.T11-3
    operatorEncId = "456-dfg-67j";
    operatorResponse = operatorManager.getOperator(operatorEncId);
    assertThat(operatorResponse.getMessage(), is("OperatorEncId is not valid!"));
    assertThat(operatorResponse, hasProperty("externalId", equalTo(null)));
    assertThat(operatorResponse, hasProperty("encId", equalTo(null)));
    assertThat(operatorResponse, hasProperty("firstname", equalTo(null)));
    assertThat(operatorResponse, hasProperty("lastname", equalTo(null)));
    assertThat(operatorResponse, hasProperty("email", equalTo(null)));
    assertThat(operatorResponse, hasProperty("active", equalTo(null)));
    operatorResponse = null;
}

```

Afbeelding 33: Implementatie van SYN.T11-1 t/m SYN.T11-3

```

//SYN.T11-4
operatorEncId = "";
operatorResponse = operatorManager.getOperator(operatorEncId);
assertThat(operatorResponse.getMessage(), is("OperatorEncId is not valid!"));
assertThat(operatorResponse, hasProperty("externalId", equalTo(null)));
assertThat(operatorResponse, hasProperty("encId", equalTo(null)));
assertThat(operatorResponse, hasProperty("firstname", equalTo(null)));
assertThat(operatorResponse, hasProperty("lastname", equalTo(null)));
assertThat(operatorResponse, hasProperty("email", equalTo(null)));
assertThat(operatorResponse, hasProperty("active", equalTo(null)));
operatorResponse = null;

//SYN.T11-5
operatorEncId = null;
operatorResponse = operatorManager.getOperator(operatorEncId);
assertThat(operatorResponse.getMessage(), is("OperatorEncId is not valid!"));
assertThat(operatorResponse, hasProperty("externalId", equalTo(null)));
assertThat(operatorResponse, hasProperty("encId", equalTo(null)));
assertThat(operatorResponse, hasProperty("firstname", equalTo(null)));
assertThat(operatorResponse, hasProperty("lastname", equalTo(null)));
assertThat(operatorResponse, hasProperty("email", equalTo(null)));
assertThat(operatorResponse, hasProperty("active", equalTo(null)));
operatorResponse = null;

//AT.T03-1
operatorEncId = operatorDao2.getEncId();
operatorResponse = operatorManager.getOperator(operatorEncId);
assertThat(operatorResponse, hasProperty("message", equalTo(null)));
assertThat(operatorResponse.getExternalId(), is(operatorDao2.getExternalId().toString()));
assertThat(operatorResponse.getEncId(), is(operatorDao2.getEncId()));
assertThat(operatorResponse.getFirstname(), is(operatorDao2.getFirstname()));
assertThat(operatorResponse.getLastname(), is(operatorDao2.getLastname()));
assertThat(operatorResponse.getEmail(), is(operatorDao2.getEmail()));
assertThat(operatorResponse.getActive(), is(Boolean.FALSE));

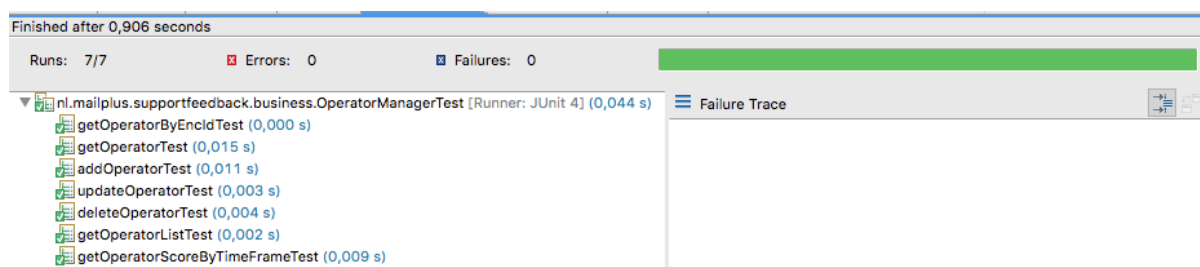
//AT.T03-2
operatorEncId = "";
operatorResponse = operatorManager.getOperator(operatorEncId);
assertThat(operatorResponse.getMessage(), is("OperatorEncId is not valid!"));
assertThat(operatorResponse, hasProperty("externalId", equalTo(null)));
assertThat(operatorResponse, hasProperty("encId", equalTo(null)));
assertThat(operatorResponse, hasProperty("firstname", equalTo(null)));
assertThat(operatorResponse, hasProperty("lastname", equalTo(null)));
assertThat(operatorResponse, hasProperty("email", equalTo(null)));
assertThat(operatorResponse, hasProperty("active", equalTo(null)));
operatorResponse = null;
}

```

Afbeelding 34: Implementatie SYN.T11-4 t/m SYN.T11-5, AT.T03-1 en AT.T03-2

10.3.4 Uitvoering van de testgevallen

Nadat alle testgevallen zijn ingevoerd voor de hoofdfunctie Operator kunnen de test uitgevoerd worden. In onderstaande afbeelding is de uitslag van de tests te zien.



Afbeelding 35: Uitvoering UT met JUnit voor hoofdfunctie Operator

Aan de groene balk rechtsboven is te zien dat alle geschreven tests doen wat wordt verwacht. Dit betekent dat de UT is geslaagd!

Nu blijkt dat de test is geslaagd, is het nog de vraag wat is de code coverage die is behaald? In onderstaande afbeelding is dit te zien.

Element	Coverage	Covered Instructions	Missed Instructions	Total Instructions
▼ OperatorManagerImpl.java	87,3 %	697	101	798
▼ OperatorManagerImpl	87,3 %	697	101	798
OperatorManagerImpl()	0,0 %	0	3	3
OperatorManagerImpl(Operator...	100,0 %	9	0	9
addOperator(Operator)	82,2 %	152	33	185
convertToOperator(OperatorDao)	100,0 %	34	0	34
createEncId(int)	56,0 %	14	11	25
createOperator(Operator)	100,0 %	24	0	24
deleteOperator(String)	78,1 %	50	14	64
getOperator(String)	100,0 %	51	0	51
getOperatorByEncId(String, boo...	100,0 %	17	0	17
getOperatorList(Integer)	100,0 %	57	0	57
getOperatorScoreByTimeFrame(...)	75,0 %	42	14	56
setOperatorFields(Operator, Op...	100,0 %	21	0	21
setOperatorResponseFields(Ope...	100,0 %	27	0	27
updateOperator(Operator)	100,0 %	69	0	69
validateEncId(String, String)	100,0 %	54	0	54
validateOperator(Operator)	70,1 %	61	26	87
validateOperatorEncId(String)	100,0 %	11	0	11

Afbeelding 36: Behaalde code coverage voor OperatorManagerImpl.java

We zien in het screenshot dat er een totale code coverage van 87,3% behaald is met deze testgevallen. Voor de rest zijn er drie methoden die eruit springen qua code coverage:

- OperatorManagerImpl() -> 0,0%;
- createEncId(int) -> 56,0% en
- validateOperator(Operator) -> 70,1%

De lege constructor wordt in zijn geheel niet gebruikt, dat is de reden dat deze een 0,0% opleverd. De methode createEncId(int) is een recursieve functie. Deze wordt niet in zijn geheel doorlopen doordat de encId's meteen uniek zijn.

De methode validateOperator(Operator) controleert of een Operator valide is. Door de testgevallen iets aan te passen en deze op meer verschillende variabelen te laten falen, is hier nog winst te behalen. Als voorbeeld, wanneer je tijdens een testgeval een niet valide Operator wilt hebben, kun je iedere keer de voornaam leeg maken. Hierdoor worden niet alle statements in deze functie doorlopen. Dit levert een lagere code coverage op. Door deze controle in verschillende testgevallen op andere variabelen te laten falen is een winst behaald van 16,1% voor deze functie. De totale code coverage is gestegen van 87,3% naar 89,1%, een winst van 1,8%! In onderstaande afbeelding is de code coverage te zien na deze aanpassingen. De methode validateOperator(Operator) is geselecteerd.

Element		Coverage	Covered Instructions	Missed Instructions	Total Instructions
▼ OperatorManagerImpl.java		89,1 %	711	87	798
▼ OperatorManagerImpl		89,1 %	711	87	798
OperatorManagerImpl()		0,0 %	0	3	3
OperatorManagerImpl(Operator...		100,0 %	9	0	9
addOperator(Operator)		82,2 %	152	33	185
convertToOperator(OperatorDao)		100,0 %	34	0	34
createEncld(int)		56,0 %	14	11	25
createOperator(Operator)		100,0 %	24	0	24
deleteOperator(String)		78,1 %	50	14	64
getOperator(String)		100,0 %	51	0	51
getOperatorByEncld(String, boo...		100,0 %	17	0	17
getOperatorList(Integer)		100,0 %	57	0	57
getOperatorScoreByTimeFrame(...)		75,0 %	42	14	56
setOperatorFields(Operator, Op...		100,0 %	21	0	21
setOperatorResponseFields(Ope...		100,0 %	27	0	27
updateOperator(Operator)		100,0 %	69	0	69
validateEncld(String, String)		100,0 %	54	0	54
validateOperator(Operator)		86,2 %	75	12	87
validateOperatorEncld(String)		100,0 %	11	0	11

Afbeelding 37: Behaalde code coverage na aanpassing methode `validateOperator(Operator)`

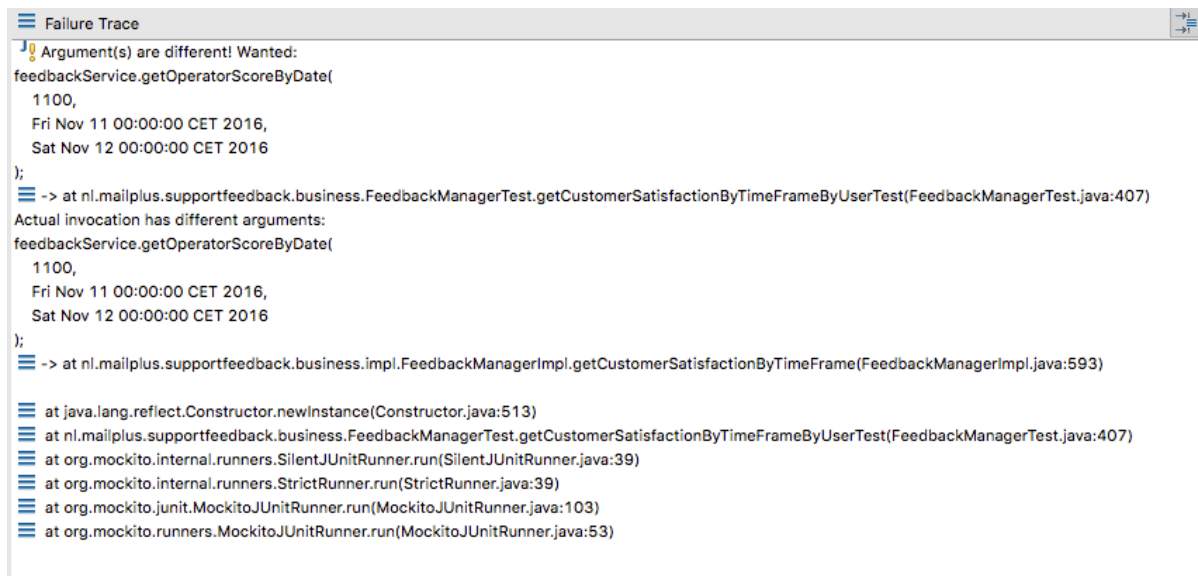
De aanpassingen die gedaan zijn in de testgevallen zijn ook aangepast in de fysieke testgevallen in het testplan⁹, voor dit voorbeeld zijn de aanpassingen hier niet opgenomen.

10.4. Sprint retrospective

Tijdens deze sprint is de functionaliteit van het SFS getest. Het gebruik van TestGoal, nadat dit opmaat is gemaakt voor deze opdracht, voor de afstudeeropdracht is een goede keuze geweest. Het aanpassen heeft echter meer tijd in beslag genomen dan van te voren was verwacht.

Tijdens de implementatie van de fysieke testgevallen is ook tijd verloren gegaan in het mocken en stubben van het object `dateService`. Deze service geeft een referentie naar een `Calendar`-object terug. Doordat er in `IncidentManagerImpl.java` en `FeedbackManagerImpl.java` veel gebruik wordt gemaakt van datums om statistieken op te halen, wordt tijdens het testen deze functie veelvuldig gebruikt. Het resultaat is dat door eerder uitgevoerde tests de waarde van het `Calendar`-object wordt aangepast en de gestubde methoden onverwachte antwoorden gaven. Dit resulteerde in falende testgevallen die lastig te waren debuggen. In onderstaande afbeelding is dit aantoonbaar gemaakt. Beide argumenten lijken identiek.

⁹ Bijlage IX: Testplan



```
Failure Trace
Argument(s) are different! Wanted:
feedbackService.getOperatorScoreByDate(
    1100,
    Fri Nov 11 00:00:00 CET 2016,
    Sat Nov 12 00:00:00 CET 2016
);
-> at nl.mailplus.supportfeedback.business.FeedbackManagerTest.getCustomerSatisfactionByTimeFrameByUserTest(FeedbackManagerTest.java:407)
Actual invocation has different arguments:
feedbackService.getOperatorScoreByDate(
    1100,
    Fri Nov 11 00:00:00 CET 2016,
    Sat Nov 12 00:00:00 CET 2016
);
-> at nl.mailplus.supportfeedback.business.impl.FeedbackManagerImpl.getCustomerSatisfactionByTimeFrame(FeedbackManagerImpl.java:593)

at java.lang.reflect.Constructor.newInstance(Constructor.java:513)
at nl.mailplus.supportfeedback.business.FeedbackManagerTest.getCustomerSatisfactionByTimeFrameByUserTest(FeedbackManagerTest.java:407)
at org.mockito.internal.runners.SilentJUnitRunner.run(SilentJUnitRunner.java:39)
at org.mockito.internal.runners.StrictRunner.run(StrictRunner.java:39)
at org.mockito.junit.MockitoJUnitRunner.run(MockitoJUnitRunner.java:103)
at org.mockito.runners.MockitoJUnitRunner.run(MockitoJUnitRunner.java:53)
```

Afbeelding 38: Voorbeeld dat de argumenten van elkaar verschillen

Het probleem is opgelost met behulp van de bedrijfsmentor. In het test-object (`FeedbackManagerTest.java`) zijn er referenties naar de Calander omgezet naar nieuwe instanties van Calendar objecten. Na deze aanpassing werkte de testgevallen direct.

Tijdens deze sprint zijn er geen toevoegingen gedaan aan het shippable product. De uitgevoerde testen dragen echter wel bij in het geheel, maar telt niet als functionaliteit waar de gebruiker iets aan heeft.

10.5. Reflectie sprint 5

Er is gekozen voor de methodiek TestGoal om het SFS te testen. Dit is een goede keuze geweest, omdat de methodiek een duidelijk stappenplan volgt. Echter doordat dit stappenplan erg uitgebreid is zijn hier een aantal aanpassingen op gedaan om het, in mijn ogen, beter toepasbaar te maken voor deze afstudeeropdracht.

Tijdens het uitvoeren van het testen ben ik er achter gekomen dat het testen van software eigenlijk best leuk is. Het mooie hieraan vind ik dat het een beeld geeft 'hoe goed je het gedaan hebt'. Door het testen van software kan worden aangetoond dat de code uitvoert wat ervan wordt verwacht.

Het gehele testtraject van ontwerpen, aanpassen van TestGoal, het verplaatsen van het testen naar een eigen sprint en het uitvoeren heeft echter veel tijd gekost. Doordat de tijdsdruk voor het inleveren van het afstudeerverslag steeds groter werd zijn helaas niet alle fysieke testgevallen omgezet naar code.

11. Procesevaluatie

In dit hoofdstuk wordt een oordeel gegeven over het gehanteerde proces, de gehanteerde methoden en technieken, de begeleiding en leerpunten.

11.1 Proces

Mijn afstuderen is op te delen in twee delen. Het eerste deel loopt vanaf mei 2016 tot september 2016 en het tweede deel begint in september 2016 en eindigt 9 januari 2017.

In het eerste deel van het afstuderen miste een goede aanpak en methodiek. Hierdoor zijn de werkzaamheden niet gestructureerd verlopen. En is er in september besloten om verlenging aan te vragen.

De tweede periode verliep veel beter. Vanaf het begin is er gebruik gemaakt van de Agile methode Scrum om de afstudeeropdracht op te splitsen in verschillende incrementen. Ondanks dat dit voor mij de eerste keer was dat ik volgens Scrum te werk ben gegaan is deze methode mij bevalen. Het grote voordeel is dat werkzaamheden worden verdeeld over meerdere sprints waardoor deze behapbaar worden gemaakt. Ook de mogelijkheid tot terugkoppeling en het aangeven van verbeteringen zijn als pluspunt ervaren. Het is gebleken dat niet alle werkzaamheden iedere sprint zijn behaald, wat ik als erg jammer heb ervaren. Ook hier biedt Scrum een oplossing voor, namelijk het terug plaatsen van deze user stories zodat deze een volgende sprint kunnen worden meegenomen.

11.2 Methoden en Technieken

Tijdens de uitvoering van de afstudeeropdracht zijn veel verschillende en voor de student nieuwe methoden en technieken gebruikt. Dit bracht een extra moeilijkheid met zich mee, maar heeft er ook voor gezorgd dat de opdracht een echte uitdaging werd.

Voor de start van de afstudeeropdracht had ik geen ervaring met Spring MVC, Jersey JAR-RS, Jackson, EclipseLink, Velocity, Hamcrest, Mockito en HSQLDB. Deze frameworks zijn tijdens de uitvoering van de afstudeeropdracht met meer of mindere maten aanbod gekomen en hebben geleid tot een extra leercurve. Er is veel geleerd over het gebruik en implementatie van deze frameworks op zichzelf en in combinatie met elkaar. Er wordt niet beweerd dat ik momenteel een expert ben in één van de gebruikte technieken, maar ik ken momenteel aardig mijn weg. Door het toepassen van frameworks wordt de programmeur gedwongen om volgens een bepaald stramien te werken. Dit resulteert in code die beter te onderhouden en uitbreidbaar is.

11.3 Begeleiding

Ik ben erg tevreden over de begeleiding die ik heb gekregen tijdens het uitvoeren van mijn afstudeeropdracht. De begeleiding vanuit de opleiding is goed geweest en ik heb goede en concrete feedback gekregen op mijn ingeleverde werk ook kon ik met mijn vragen terecht.

De begeleiding vanuit Mailplus B.V. is als erg fijn ervaren. Tijdens de vergaderingen, en zeker ook daarbuiten, was er de mogelijkheid om vragen te stellen en hulp te vragen. Dit kon gaan over kleine zaken of over grotere problemen waar ik op dat moment tegen aan liep. Zeker door het meedenken, de adviezen en hulp heb ik dit resultaat kunnen bereiken!

Allen dank hiervoor.

11.4 Leerpunten

De afstudeerperiode is er een geweest van ups en downs. Van lange dagen en volle weken en weekenden. Nu ik terug kijk op de afgelopen periode en het product dat is opgeleverd zijn er zeker leerpunten te vinden.

Gestructureerd werken is een van de eerste die in mij opkomt. Het ontbreken van een goede structuur en aanpak in het eerste deel van de afstudeeropdracht, heeft geleid tot minder efficiënt werk. Er zijn in deze periode zeker belangrijke stappen gezet naar het resultaat, maar ik liep er tegen aan dat er geen goede structuur en aanpak was om te volgen. Tijdens het tweede deel van de afstudeeropdracht was er wel een goede structuur en aanpak. Hierdoor werd er efficiënter gewerkt en werden grotere stappen gemaakt. Als deze structuur een aanpak vanaf het begin was toegepast had dit er bijvoorbeeld voor kunnen zorgen dat het opgeleverde product meer functionaliteiten heeft.

Nu ik terug kijk op de afstudeeropdracht en met de kennis van nu had ik de requirements met een andere visie ingedeeld. Nu lag de visie meer op het verdiepen van het huidige product dan op het afstudeerverslag. Wanneer ik ervoor had gekozen om de indeling zo te maken dat deze meer gericht was op het afstudeerverslag had de opzet van het eindproduct breder geweest. Bijvoorbeeld de mogelijkheid tot het voeren van meer talen zou ertoe hebben geleid dat de opzet breder wordt, het systeem kan overweg met meerdere talen. Deze andere insteek zou kunnen hebben geleid tot meer waardoor de verantwoording in het verslag eenvoudig wordt. De gekozen indeling is niet verkeerd of slecht te noemen, alleen anders en wanneer ik de opdracht over zou doen zou ik dit wellicht anders aanpakken.

12. Product evaluatie

In dit hoofdstuk wordt een oordeel gegeven over de totstandkoming van het opgeleverde product.

12.1 Support Feedback Systeem v2

Het opgeleverde eindproduct (hierna: versie 2) heeft ten opzichte van de huidige implementatie (hierna: versie 1) meer functionaliteiten. Aan de Rest-API zijn meer functionaliteiten toegevoegd. Versie 2 is geheel te bedienen via de Rest-API zo kunnen er incidenten, gebruikers en behandelaars worden aangemaakt en bewerkt. Verder zijn er meer mogelijkheden tot het uitvragen van statistieken toegevoegd en huidige mogelijkheden uitgebreid. Met deze toevoegingen is het bijvoorbeeld mogelijk om de statistieken meer flexibel te tonen. Het is nu bijvoorbeeld mogelijk om de statistieken met elkaar te vergelijken per periode en/of deze te tonen in grafieken. Een voorbeeld van een uitbreiding van de statistieken is het toevoegen van meer informatie bij de live scores en de percentages die nu niet meer als hele percentages worden getoond. Het is verder mogelijk in versie 2 om ervoor te kiezen om de feedback mails niet te verzenden via het SFS, wat in versie 1 verplicht was. Last but not least is het in versie 2 mogelijk om met meerdere gebruikers tegelijk gebruik te maken van het SFS.

Tijdens de uitvoering van de afstudeeropdracht zijn alle must-haves en should-haves geïmplementeerd. In het afstudeerplan¹⁰ staat vermeld dat een doel is om meerdere gebruikers toe te staan in versie 2. Hierdoor wordt het mogelijk om versie 2 te gaan verkopen als SaaS oplossing en/of te gebruiken door meerdere afdelingen. Ook staat er vermeld als doel dat er door middel van een Rest-API gecommuniceerd kan worden met het SFS. Zoals te lezen is in het afstudeerverslag zijn deze doelen behaald en hiermee het doel van de afstudeeropdracht. De bedrijfsmentor beaamt deze bevinding in zijn beoordeling welke is toegevoegd als bijlage XII.

12.2 OAuth 1.0a

In versie 1 van het SFS werd de autorisatie gedaan met behulp van OAuth 2. Deze implementatie is voldoende voor de eerste versie. In de nieuwe versie, versie 2, had ik dit graag omgezet naar OAuth 1.0a, zoals is uitgelegd in het hoofdstuk over methoden en technieken. Helaas is dit door tijdgebrek niet in versie 2 terecht gekomen. Dit vind ik persoonlijk erg jammer en hoop wanneer er verder wordt ontwikkeld aan versie 2 dit toegevoegd kan worden.

OAuth staat nog steeds vernoemd als gebruikte methode in het afstudeerverslag. Ondanks dat de implementatie niet in versie 2 is gekomen, is er tijd en aandacht besteed aan het onderzoek naar deze methode.

12.3 Aanbevelingen

Als laatst wil ik graag aanbevelingen doen voor de toekomstige ontwikkeling van het SFS. Als eerste de implementatie van OAuth 1.0a voor de autorisatie van de Rest-API. Doordat er

¹⁰ Bijlage XI: Afstudeerplan

recent aanpassingen zijn doorgevoerd in de werkwijze en processen van de development afdelingen binnen Mailplus B.V. zou ik graag de volgende aanbevelingen doen.

12.3.1 Frontend

Een van de vernieuwingen is het gebruik van de nieuwe frontend techniek Angular 2. Door het toepassen van Angular 2 binnen het SFS, blijven de gebruikte technieken voor de frontend consistent binnen Mailplus B.V. Ook levert dit het voordeel op dat de nieuwe design library gebruikt kan worden, wat de ontwikkeling standaardiseert, verbeterd en versneld. Het gebruik van Angular vergroot ook de schaalbaarheid en uitbreidbaarheid van het product.

12.3.2 Backend

Een andere aanbeveling is het volledig scheiden van de front- en backend. Hierdoor ontstaat een duidelijke scheiding in verantwoordelijkheden tussen de frontend geschreven in Angular voor het weergeven van de juiste informatie en de backend voor het verwerken van deze informatie.

12.3.3 TopDesk integratie

Momenteel is de integratie van TopDesk nog geïntegreerd in versie 2. Mijn advies is om dit op te nemen in een aparte applicatie. Deze applicatie heeft als doel om de mails vanuit TopDesk om te zetten naar Rest-API verzoeken, welke vervolgens naar het SFS worden verstuurd. Meer uitleg is te lezen in het hoofdstuk over sprint 5.

13. Beroepstaken

Dit hoofdstuk beschrijft de beroepstaken die in dit afstudeeronderzoek gedefinieerd zijn.

13.1 Beroepstaak 2.2

Ontwerpen, bouwen en bevragen van een database niveau: 4 (zelfstandig, complex)

Het ontwerpen en implementeren van het gegevensmodel voor de relationele database. Het omzetten en implementeren van het fysieke ontwerp met behulp van SQL. Het uitdenken en het ontwerpen van de benodigde query's en constraints met behulp van ODL. De database draait op een Tomcat applicatie cluster met een MySQL database. Met het ontwerpen van de database structuur wordt rekening gehouden met concurrent gebruikers, performance- en beveiligingsaspecten.

Tijdens de uitvoering van de afstudeeropdracht is de huidige database uitgebreid met meerdere tabellen. De data is hiervoor genormaliseerd. De nieuwe database kan door meerdere concurrent gebruikers worden geraadpleegd. Het ontwerp is in de uitvoering van de opdracht minder complex gebleken van vooraf was ingeschat. Echter door het gebruik van verschillende frameworks, zoals EclipseLink en HSQLDB zit er voldoende complexiteit op dit gebied in de uitgevoerde opdracht.

13.2 Beroepstaak 3.2

Ontwerpen systeemdeel niveau: 4 (zelfstandig, complex)

Het in detail ontwerpen van de applicatie met behulp van design patterns en andere best-practices. Er zal rekening worden gehouden met de frameworks waarin de applicatie uiteindelijk wordt gebouwd.

Zoals te zien is in [bijlage X] is het klassendiagram fors uitgebreid en is er rekening gehouden met toekomstige wijzigen, testbaarheid, onderhoudbaarheid en hergebruik in de code. Verder is er rekening gehouden met het gebruik van best-practices van de gebruikte frameworks en MailPlus B.V. en zijn design patterns ingezet. De grote verscheidenheid aan gebruikte frameworks hebben de complexiteit verhoogd.

13.3 Beroepstaak 3.3

Bouwen applicatie niveau: 4 (zelfstandig, complex)

Het in delen implementeren en documenteren van het ontwerp tot één werkend systeem. De te ontwerpen applicatie zal geschreven worden in de programmeertaal Java. Hiervoor is gekozen omdat dit de standaard programmeertaal is binnen MailPlus. Er zal gebruik worden gemaakt van Spring MVC als web applicatie framework met Jersey (JAX-RS) voor de implementatie van de API. Voor persistence (JPA) zal gebruik worden gemaakt van EclipseLink. Als template engine wordt gebruik gemaakt van Velocity.

Deze beroepstaak is voorafgaand aan de afstudeeropdracht ingeschat op lastig, echter ben ik van mening dat dit achteraf aangepast kan worden naar complex. Tijdens de implementatie van het ontwerp is gebruik gemaakt van vele verschillende frameworks die onbekend waren voor de student. Voor het ontwikkelen van de code is gebruik gemaakt van Eclipse en versiebeheer is gedaan met behulp van Git via SourceTree. De applicatie draait momenteel in een testomgeving.

13.4 Beroepstaak 3.4

Initiëren en plannen van het testproces: 3 (zelfstandig, lastig)

Het betreft het opstellen van een detailtestplan en een mastertestplan. De applicatie bestaat uit meerdere subsystemen. Er wordt gebruik gemaakt van een testmethodiek en er wordt een testrisicoanalyse uitgevoerd. De nadruk ligt op het testen van de functionaliteit. Er is aandacht voor herhaalbaarheid van de testen en een testframework. Acceptatietesten vormen een onderdeel van het testplan.

Vooraf is deze beroepstaak niet genoemd. Achteraf blijkt dat ik deze stappen aantoonbaar heb uitgevoerd op niveau 3. In sprint 5 is het testen van het eindproduct uitgevoerd. Zoals is te lezen in het hoofdstuk is hiervoor een testplan opgesteld. Dit is met beargumenteerde redenen niet opgesplitst in een master- en detailtestplan, zoals hierboven staat beschreven. Voor het schrijven van het testplan is gebruik gemaakt van de testmethodiek TestGoal. In het testplan¹¹ is een 1-D testrisicoanalyse opgenomen. Voor de implementatie van de tests is gebruik gemaakt van het framework JUnit met als toevoegingen: Hamcrest, HSQLDB en Mockito. Deze opzet zorgt voor herbruikbaarheid van de testgevallen.

13.5 Beroepstaak 3.5

Uitvoeren van en rapporteren over het testprocesniveau: 3 (zelfstandig, lastig)

Het opstellen per testobject van de logische- en fysieke testgevallen, het bepalen van de testbasis en verwachte uitkomst. Vervolgens het uitvoeren van de tests en het noteren van de bevindingen. Aan de hand van de uitkomsten van de uitgevoerde tests zal een testrapport worden opgesteld. Als testmethode zal er gebruik worden gemaakt van TestGoal. Voor de uitvoering van geautomatiseerde testgevallen zal gebruik worden gemaakt van het pakket dat reeds in gebruik is binnen MailPlus en anders wordt JUnit gebruikt.

Tijdens de uitvoering van de afstudeeropdracht is het gemaakte testplan gedeeltelijk uitgevoerd. Helaas door tijdgebrek was het niet mogelijk om deze geheel te implementeren. De uitgevoerde testgevallen bevatten Unit Testen voor de service- en business laag van de applicatie. Tijdens de uitvoering van de testen is er rekening gehouden met hergebruik in de toekomst door het gebruik van verschillende (eerder genoemde) frameworks en testtool. Het gebruik hiervan heeft complexiteit toegevoegd aan de uitvoering.

¹¹ Bijlage IX: Testplan

Literatuurlijst

Authenticating with OAuth. (2009). Geraadpleegd op 2 oktober 2016 van <https://oauth.net/core/1.0a/>

Boer, P. de, Bruggink, M., Bruns, M., van de Hoef, N., Peters, G., Roozmond, M., & Wijnands, W. (2015). *Scrum in actie: maak van elk project een succes!* Business Contact.

FasterXML. (2013). Geraadpleegd op 1 oktober 2016 van <http://wiki.fasterxml.com/JacksonHome>

Grood, D.J., de (2008). *TestGoal Leerboek resultaat gedreven software testen*. Den Haag: Sdu uitgevers bv

High Availability and Scalability. (n.d.). Geraadpleegd op 30 augustus 2016 van <http://dev.mysql.com/doc/refman/5.7/en/ha-overview.html>

Johnson, R., Hoeller, J., Donald, K., Sampaleanu, C., Harrop, R., Risberg, T., ... & Templier, T. (2004). The Spring Framework–Reference Documentation. *Interface*, 21.

JUnit (z.d.). Geraadpleegd op 01-01-2017 van http://junit.org/junit4/faq.html#overview_1

Loon, D. D. van, Barendsen, E. E., & Hoppenbrouwers, S. S. (2015). Het overzichtelijk houden van kennis bij het gebruik van Scrum: Een casestudie bij Topicus.

OAuth.net. (z.d.). Geraadpleegd op 2 oktober 2016 van <https://oauth.net/>

OAuth.net. (2009). Geraadpleegd op 2 oktober 2016 van <https://oauth.net/core/1.0a/>

Oracle Corporation. (2016). Geraadpleegd op 30 november 2016 van <https://jersey.java.net/>

Oracle Corporation and/of its affiliates. (2016). *15.8.8 Limits on InnoDB Tables*. Geraadpleegd op 30 augustus 2016 van <http://dev.mysql.com/doc/refman/5.7/en/innodb-restrictions.html>

OWAPS. (z.d.). REST Security Cheat Sheet. Geraadpleegd op 26 november 2016 van https://www.owasp.org/index.php/REST_Security_Cheat_Sheet#Authentication_and_session_management)

Swart, N. de (2010). *Handboek requirements, brug tussen business en ict*. Delft: Eburon Business.

The Apache Software Foundation (2016). Geraadpleegd op 01-01-2017 van <http://velocity.apache.org/>

The Eclipse Foundation (2015). Geraadpleegd op 31-12-2016 van <http://www.eclipse.org/eclipselink/#about>

The HSQL Development Group (mei 2016). Geraadpleegd op 01-01-2017 van <http://hsqldb.org/>

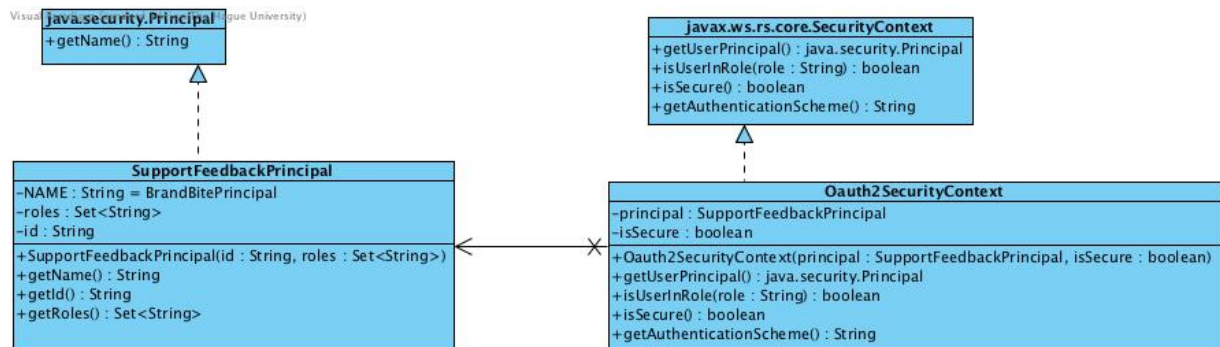
Vogel, L. (2016). Purpose of the Hamcrest matcher framework. Geraadpleegd op 26-11-2016 van <http://www.vogella.com/tutorials/Hamcrest/article.html> - hamcrestoverview

Verklarende woordenlijst

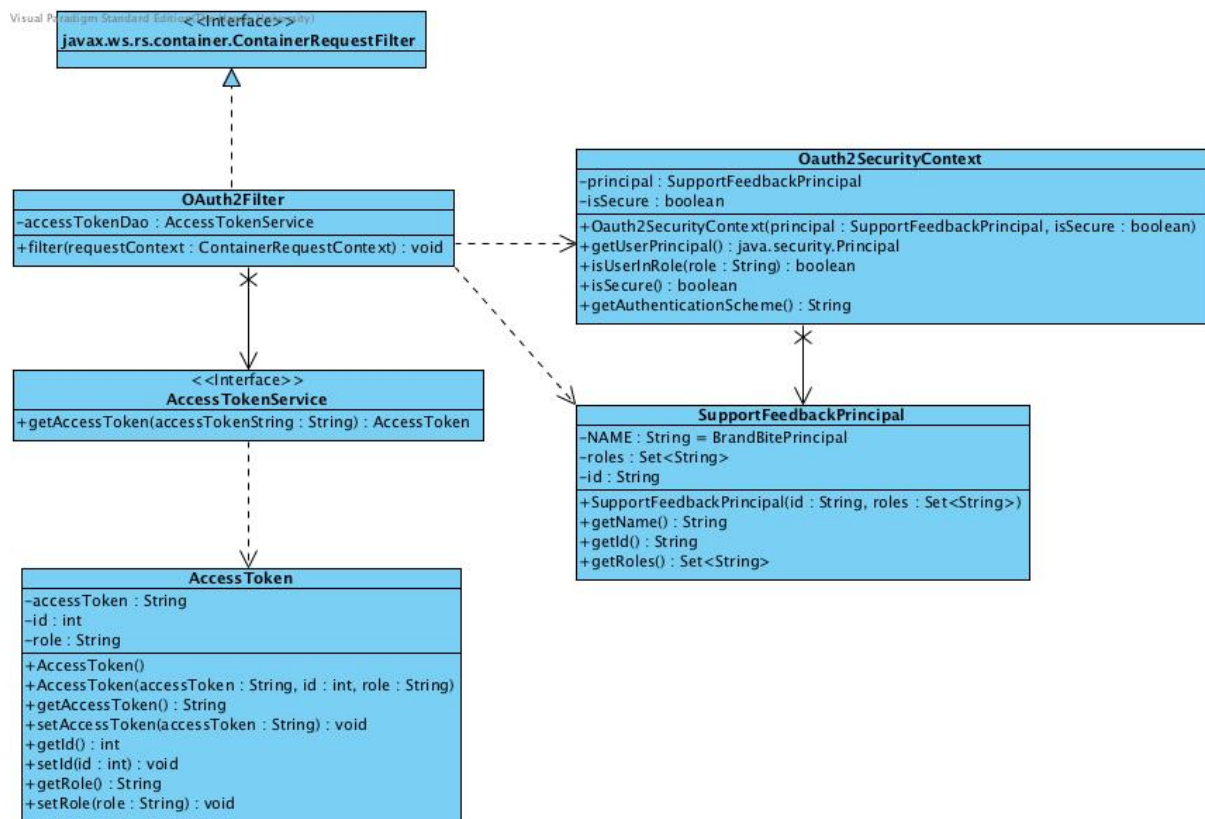
Access Token:	Een waarde die gebruikt wordt door de Consumer voor het verkrijgen van toegang tot de Protected Resources namens de User. Deze wordt gebruikt in plaats van de login gegevens van de User.
Behandelaar:	de medewerker die de feedback verstuurd.
Beoordelaar:	de persoon die de feedback ontvangt.
Consumer:	Een website of een applicatie die gebruik maakt van OAuth om toegang te krijgen tot een Service Provider namens een User.
Fake-object	Een Fake-object is een lichtgewicht implementatie met dezelfde functionaliteiten die gelijk is aan de echte implementatie. Tijdens runtime wordt echter het fake-object gebruikt.
Feedback verzoek:	het verzoek tot het geven van een beoordeling door de beoordelaar.
Feedback mail:	de mail die verzonden wordt aan de beoordelaar met daarin de links om de beoordeling te verrichten.
Feedback monitor:	de applicatie die getoond wordt op de afdeling support.
Feedback link:	een link waarmee een beoordeling gedaan kan worden, een voorbeeld hiervan is een link achter de 'ok-waardering'-knop.
Protected Resource(s):	Data die beheert wordt door de Service Provider, waar toe de Consumer toegang kan krijgen door middel van authenticatie.
Request Token:	Een waarde die gebruikt wordt door de Consumer voor het verkrijgen van autorisatie door de User. Ook wordt deze ingeruild tegen een Access Token.
Service Provider:	Een web applicatie die toegang verschaft via OAuth.
User:	Een persoon die een account heeft bij de Service Provider.

Bijlagen

Bijlage I : Klassendiagram huidige implementatie



Afbeelding 39: OAuth klassen

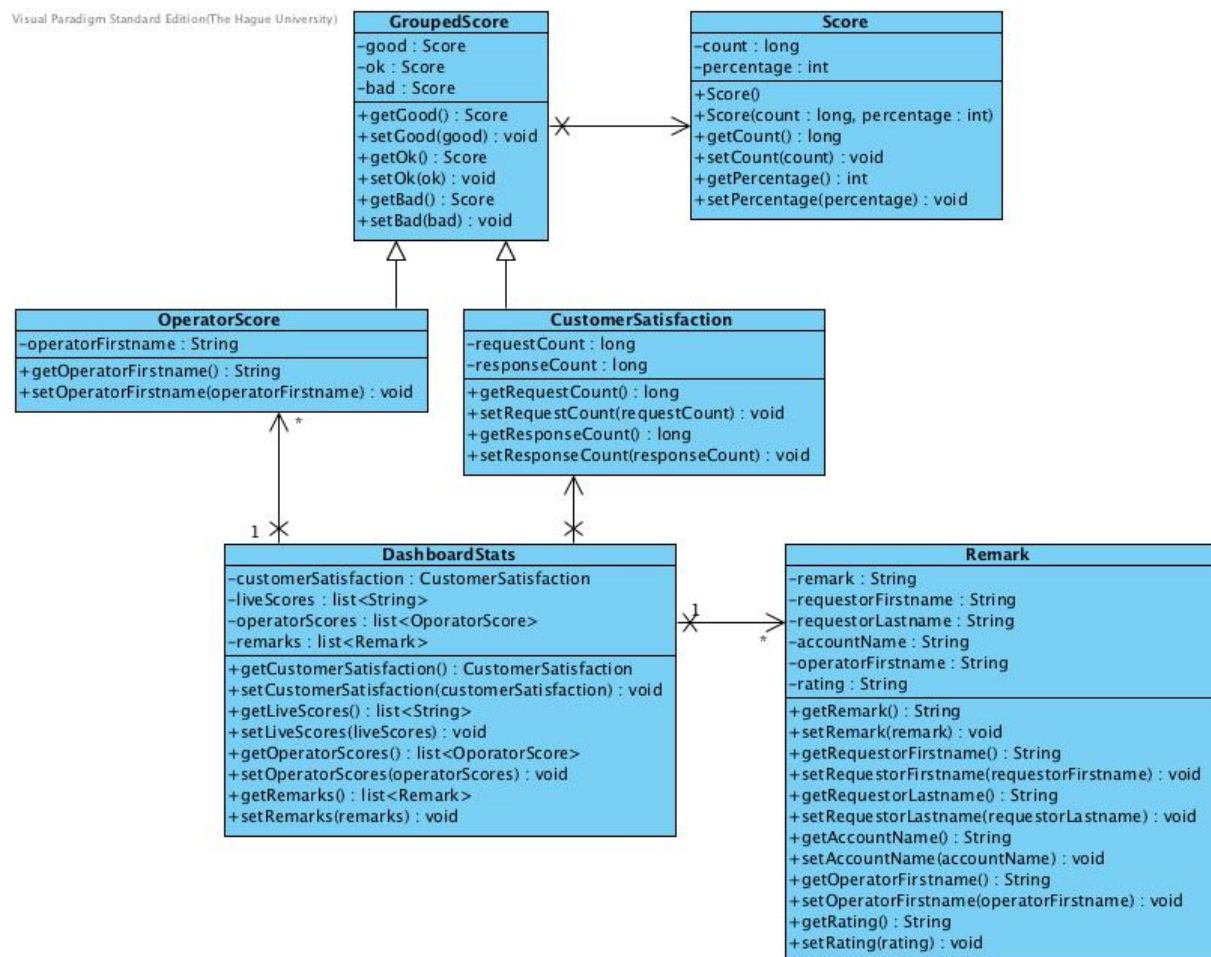


Afbeelding 40: OAuth Filter klassen

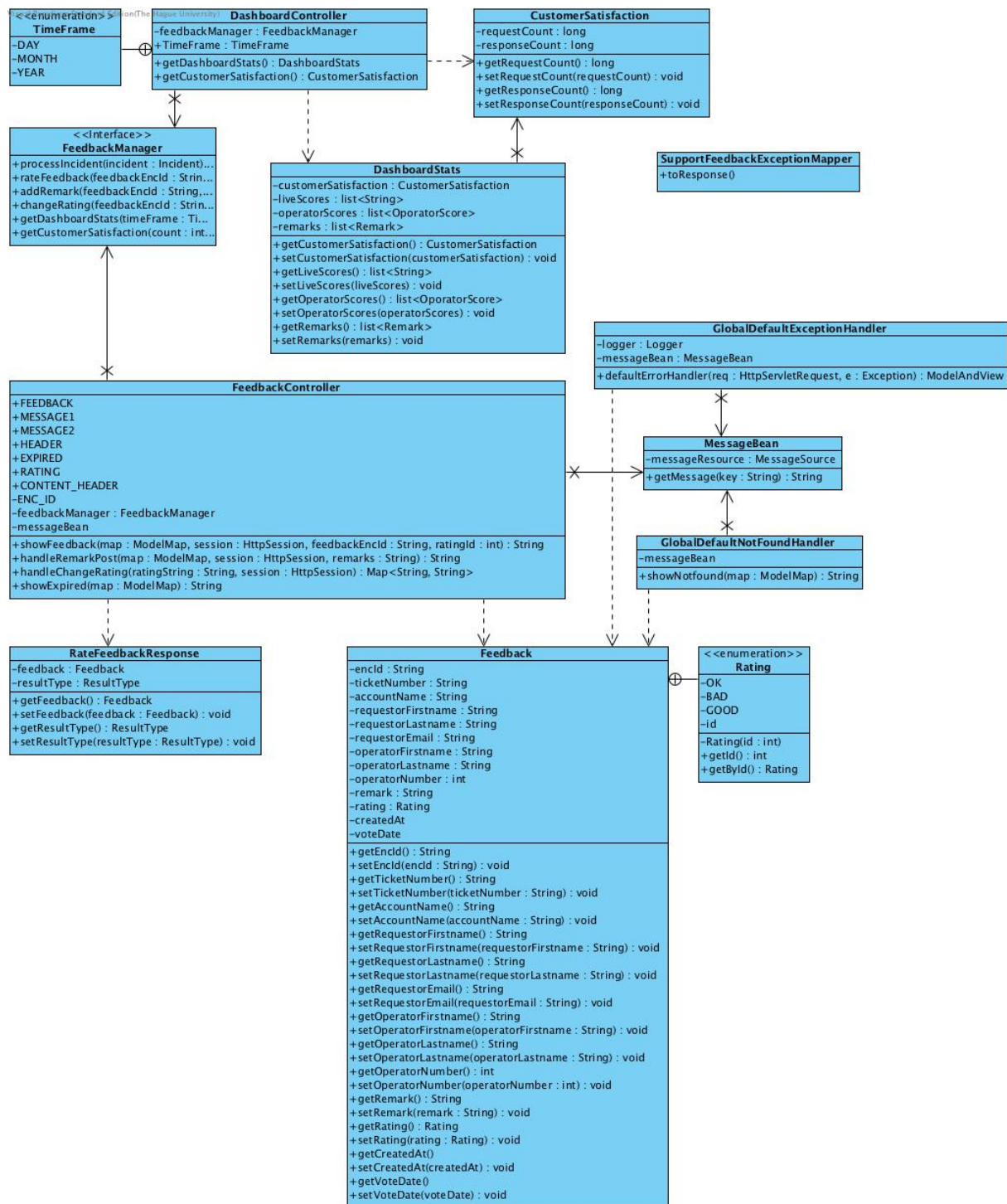
Visual Paradigm Standard Edition(The Hague University)	IdCreator
-keyCharacterString : String = abcdefghijklmnpqrstuvwxyxABCDEFGHIJKLMNOPQRSTUVWXYZ3456789	
-rand : SecureRandom	
-encIdlength : int = 32	
+createEncId() : String	

MessageBean
-messageResource : MessageSource
+getMessage(key : String) : String

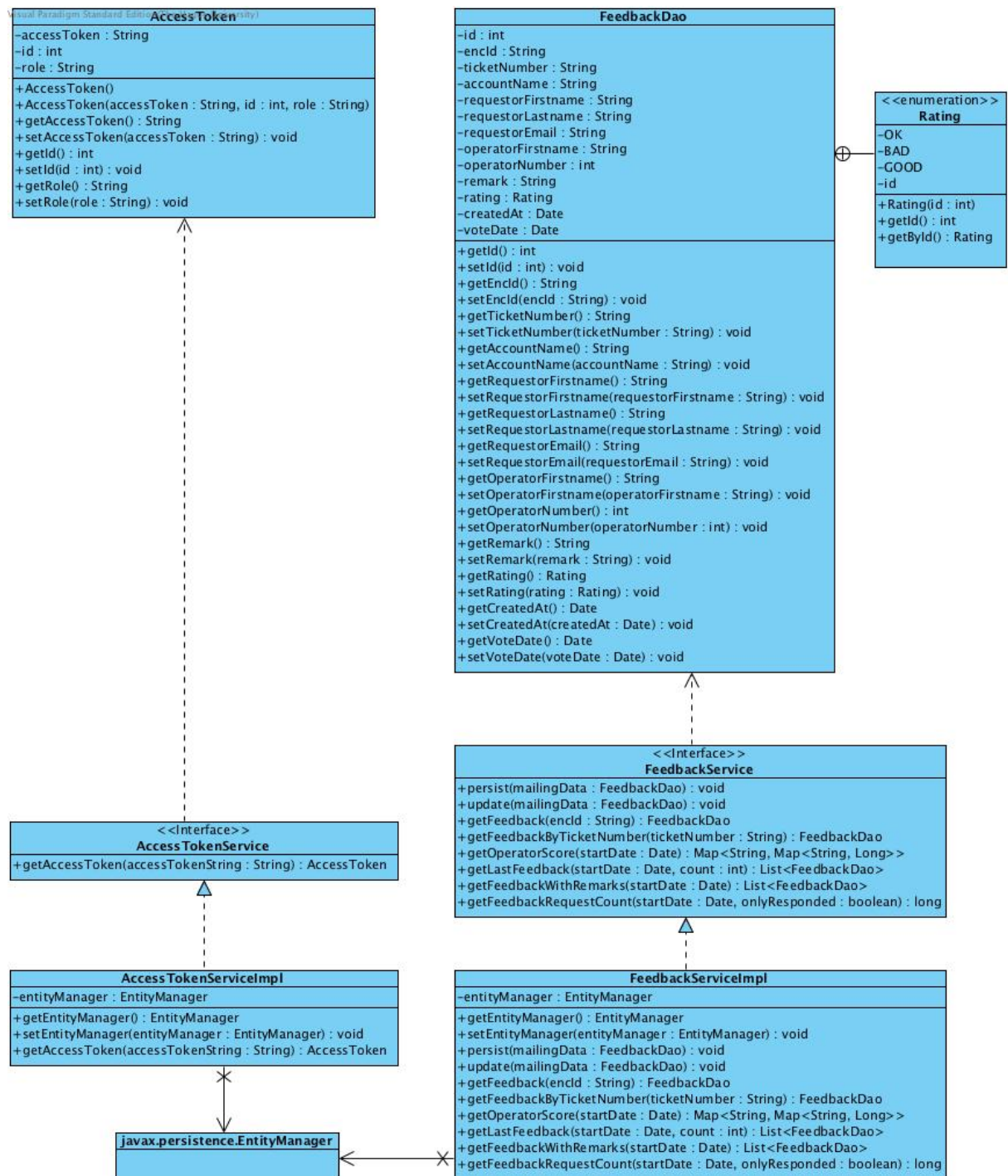
Afbeelding 41: Utility klassen



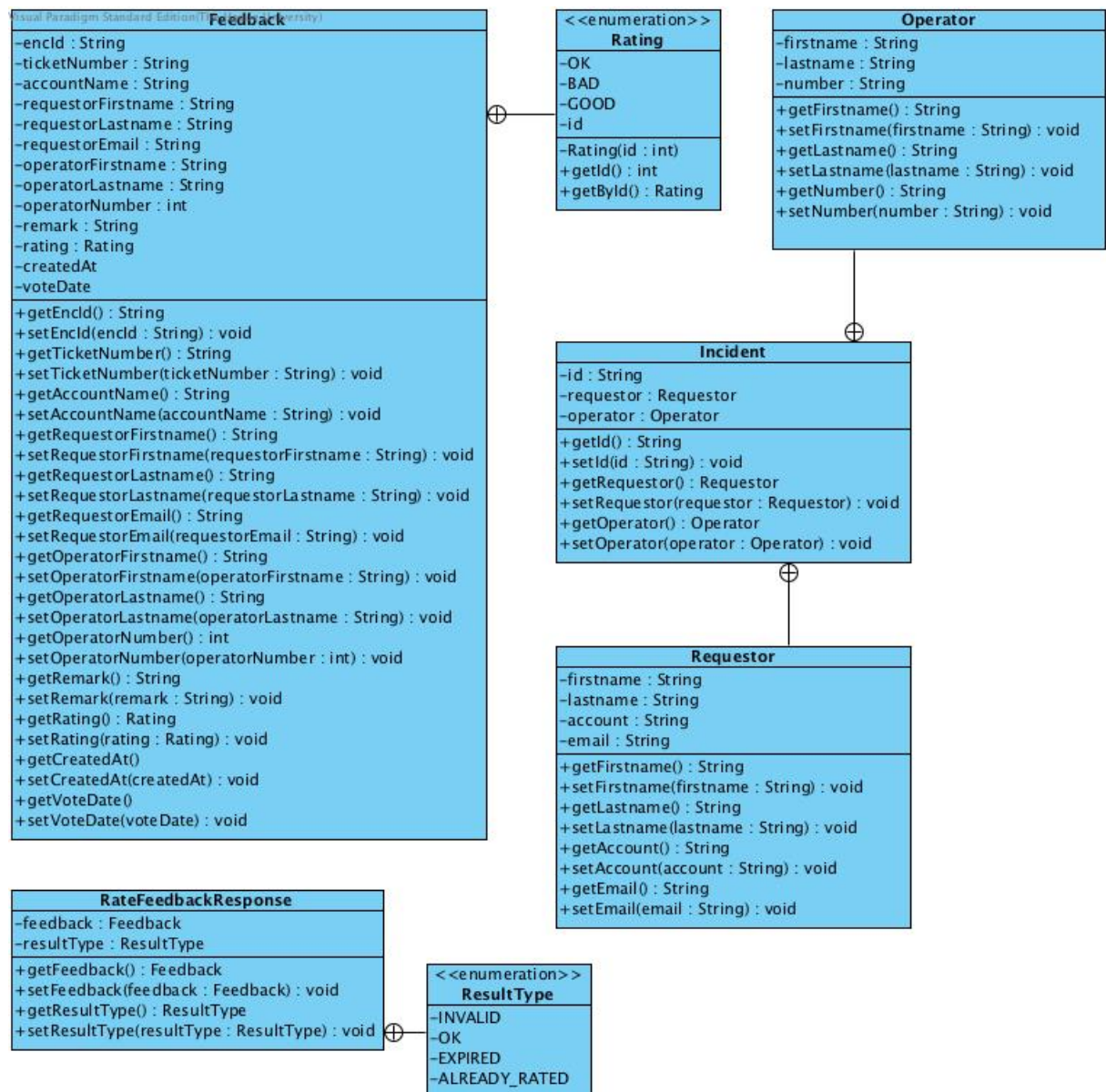
Afbeelding 42: Dashboard klassen



Afbeelding 43: Controller klasse

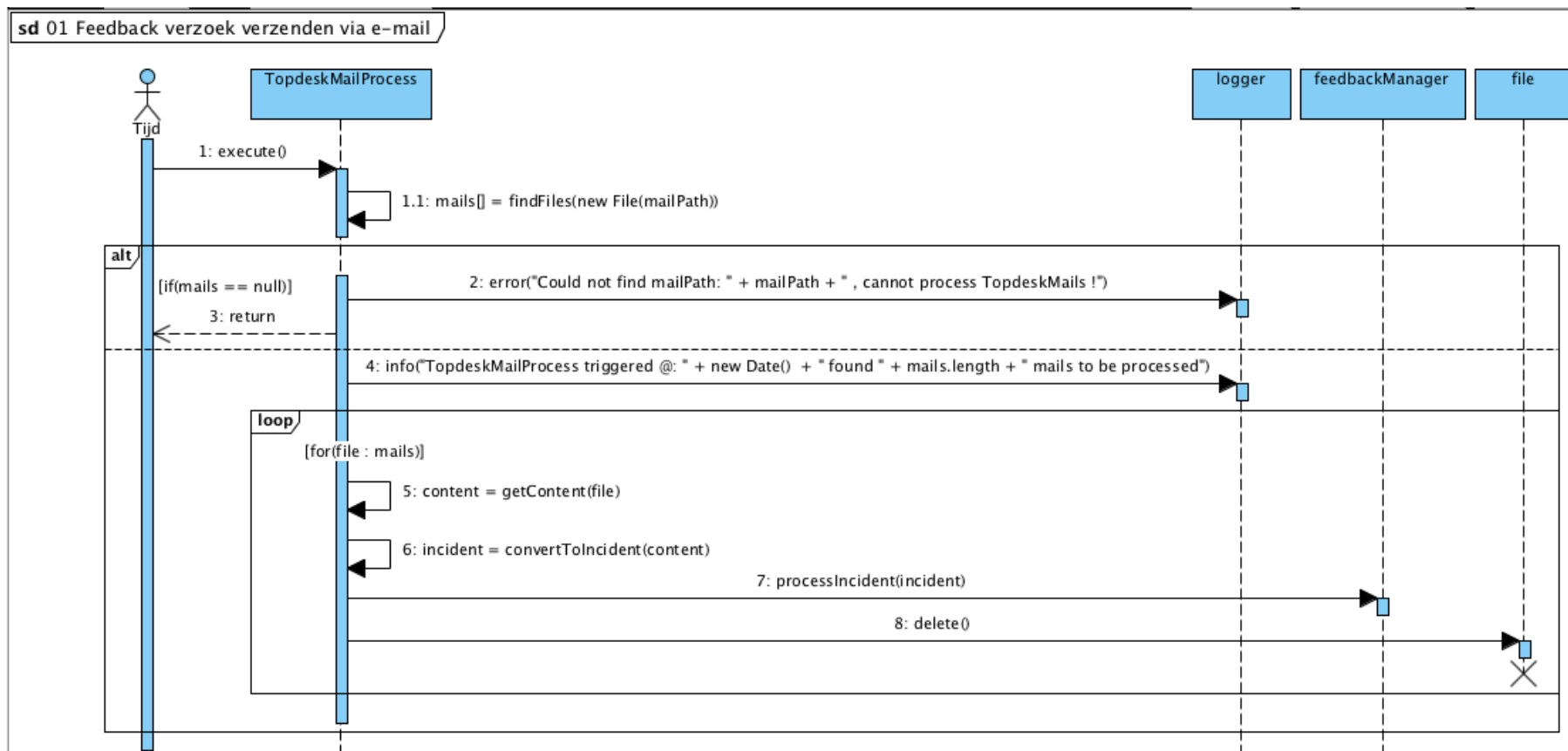


Afbeelding 45: Service klassen



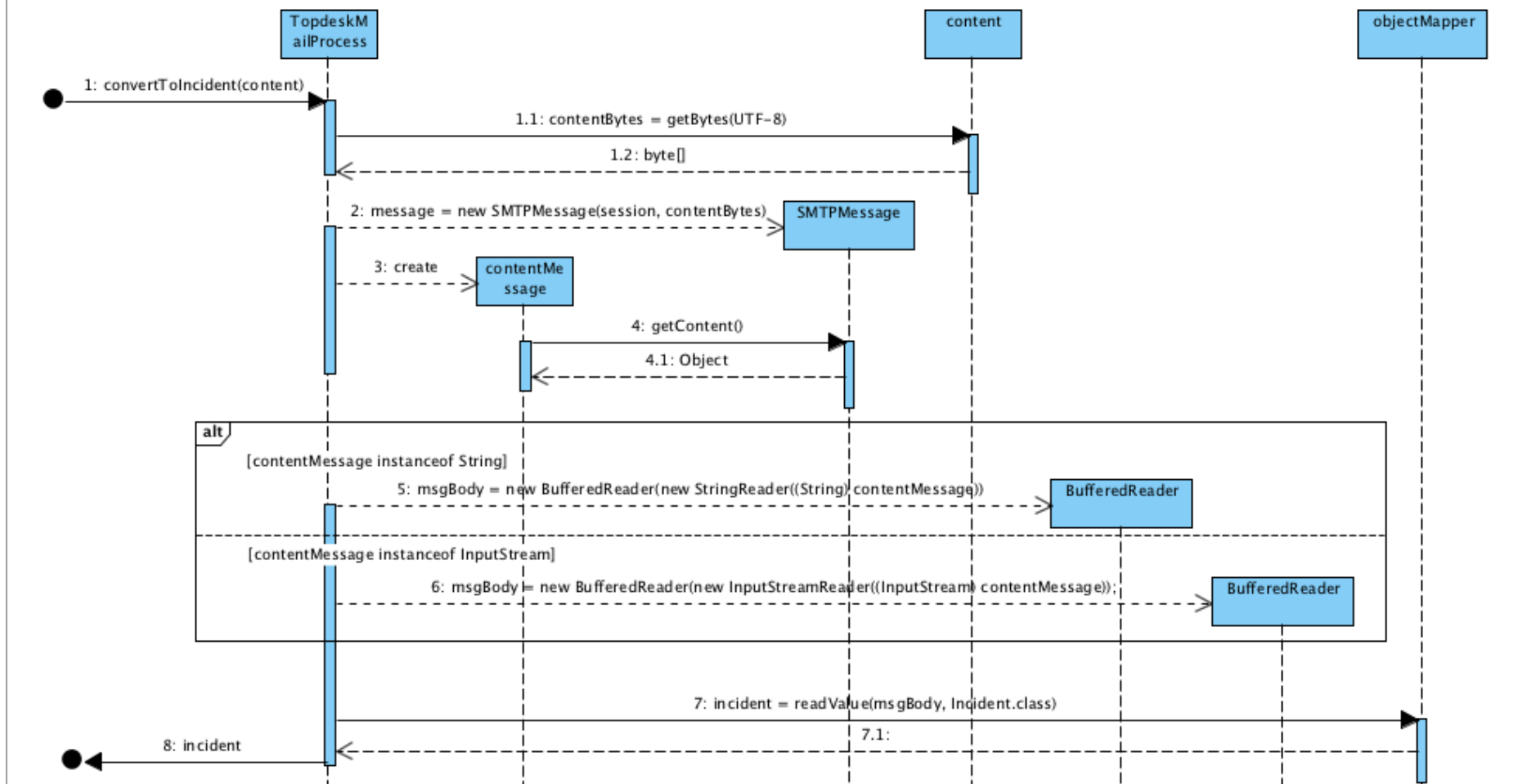
Afbeelding 46: Model klassen

Bijlage II: Sequentiediagrammen key-features huidige implementatie



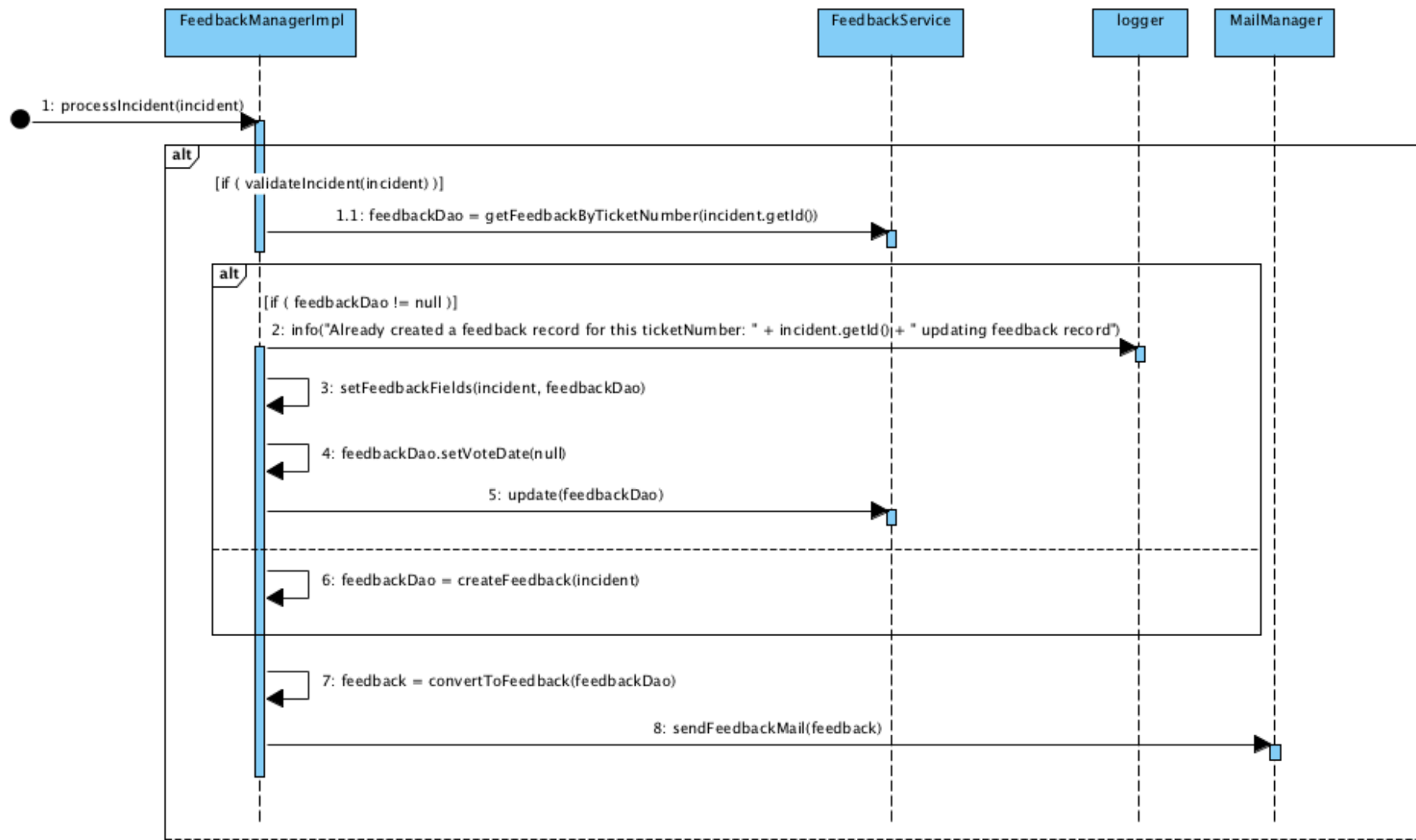
Afbeelding 47: Verwerking van feedback verzoek per mail

sd 01.1 Zet mail om naar incident



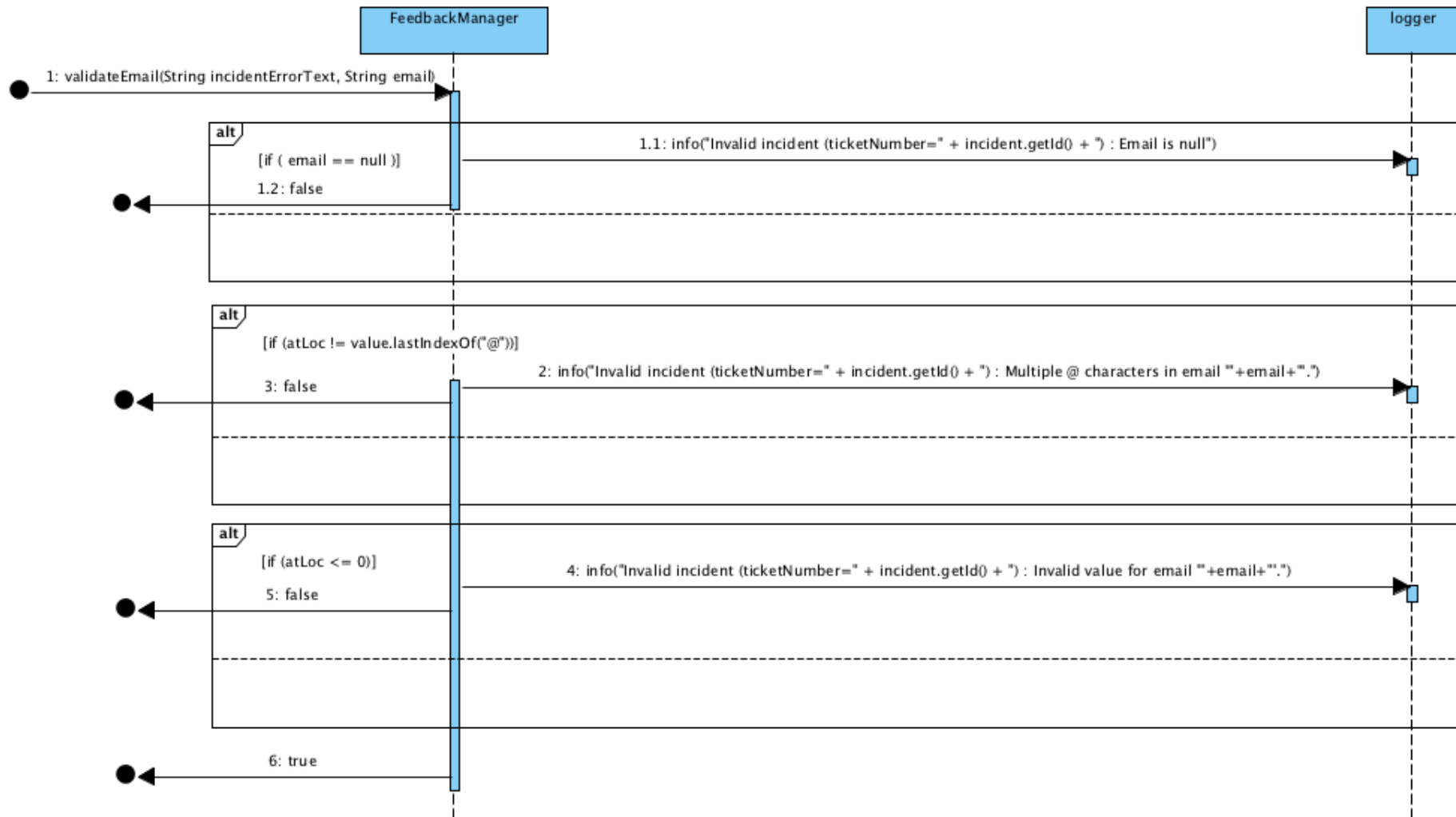
Afbeelding 48: Omzetten van e-mail naar Incident object

sd 01.2 Verwerk Incident



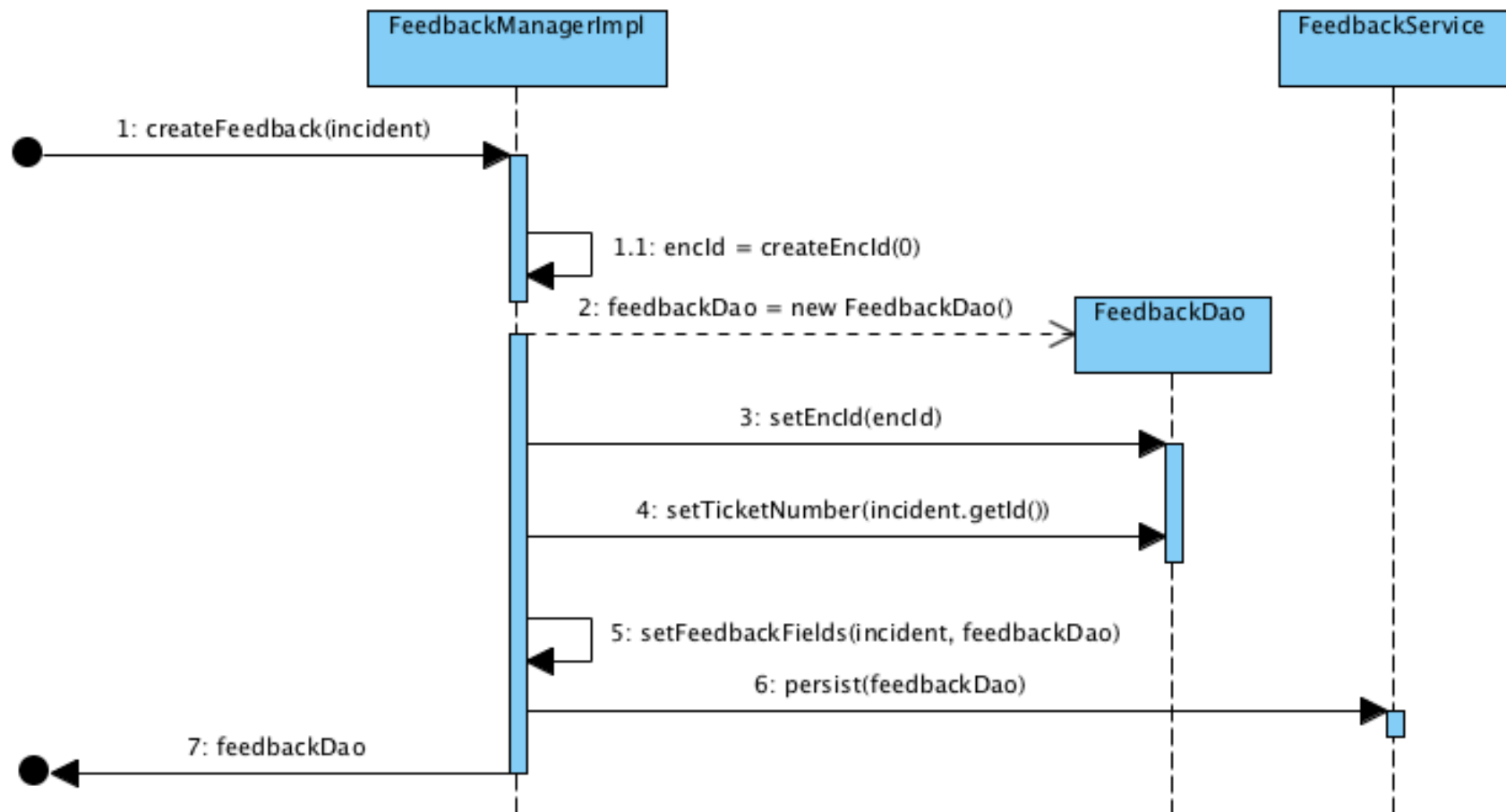
Afbeelding 49: Verwerken van Incident object tot feedbackmail

sd 01.2.1 Valideer incident



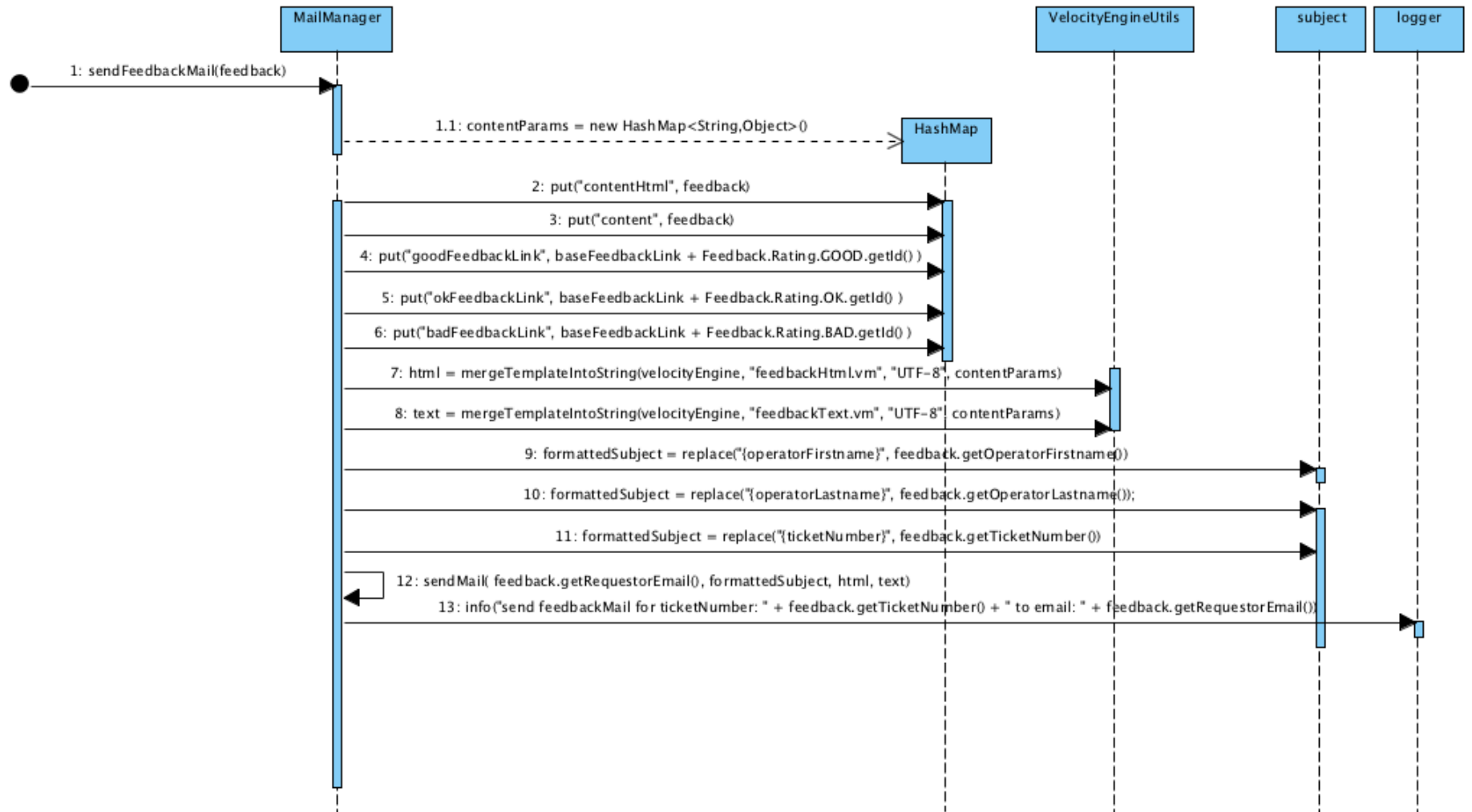
Afbeelding 50: Validatie van Incident object

sd 01.2.2 Creëer nieuw feedback record



Afbeelding 51: Omzetten van Incident object naar record in database

sd 01.2.3 Verzend feedback mail



Afbeelding 52: Detail van verzenden feedbackmail

Bijlage III: Huidige opzet

RRM huidige database

Primaire sleutel = attribuut_naam

Vreemde sleutel = *attribuut_naam*

AccessToken (id, accessToken, role)

Primaire sleutel: id

Feedback (id, accountName, createdAt, encId, operatorFirstname, operatorLastname, operatorNumber, rating, remark, requestorEmail, requestorFirstname, requestorLastname, ticketNumber, voteDate)

Primaire sleutel: id

Index: encId, ticketNumber, operatorNumber (operatorNumber, createdAt)

Bijlage IV: Het concurrentieonderzoek

Wat is het doel van het onderzoek?

Het doel van dit concurrentieonderzoek is het achterhalen welke functionaliteiten andere systemen bieden voor het uitvragen en analyseren van feedback. In het bijzonder feedback over de tevredenheid van klanten kort na een geleverde dienst.

Waarom wordt dit onderzocht?

De uitkomsten van dit onderzoeken kunnen input leveren voor de functionaliteiten van het te ontwikkelen feedback systeem. Welke functionaliteiten maken het dat klanten een bepaald systeem gebruiken? Welke goede functionaliteiten kunnen wij overnemen?

Hoe wordt dit onderzocht?

Aan de hand van vijf vragen over de werking, mogelijkheden en functionaliteiten van de feedback systemen worden de overkomsten en verschillen tussen verschillende feedback systemen op de markt in kaart gebracht.

Op welke manier wordt de feedback uitgevraagd?

Wordt dit veelal gedaan door één vraag te stellen? Of wordt er gebruik gemaakt van een vragenlijst bestaande uit meerdere vragen? Welke vraag- en antwoordtypen worden ondersteund?

Op welke manier(en) kan het feedbackverzoek worden verzonden?

Hoe komt het verzoek tot feedback bij de klant terecht? Gebeurt dit veelal via e-mail of zijn er ook concurrenten die gebruik maken van modernere technieken zoals sociale media?

Welke mogelijkheden tot analyse van de data bieden de andere systemen?

Hebben deze systemen kant en klare dashboards en is het mogelijk om deze naar wens aan te passen? Is het mogelijk om statistieken te exporteren, zo ja hoe?

Hebben de andere systemen slimme en/of extra features?

Bieden concurrenten slimme features aan waar Mailplus B.V. nog niet aan heeft gedacht?

Welke mogelijkheden tot integratie bieden de andere systemen?

Kan je het systeem koppelen met andere systemen zodat de statistieken in een ander systeem getoond kunnen worden? Is het mogelijk om je CRM-pakket er aan te koppelen zodat je gebruik kunt maken van deze data (e-mailadressen, naam, etc.)? Bieden concurrenten een API aan waarmee data opgehaald kan worden uit het systeem?

Door het beantwoorden van bovenstaande vragen kan een goed beeld worden verkregen van de andere feedback systemen op de markt. Door het in kaart brengen van de antwoorden komen er wellicht nog goede features naar boven die overgenomen kunnen worden in het te ontwikkelen feedback systeem.

Hoe worden concurrenten gevonden?

Doormiddel van het gebruik van de zoekmachine Google en verschillende zoektermen wordt verwacht eenvoudig een aantal feedback systemen te kunnen vinden. In nieuwsbrieven van grote bedrijven zoals: Conrad.nl, Bol.com, de Consumentenbond en Coolblue bevindt zich tegenwoordig de mogelijkheid tot het geven van feedback op de nieuwsbrief. Doordat deze manier van feedback gaat over de mailing die wordt verzonden en niet over een geleverde dienst, vallen deze bedrijven buiten de scope van dit onderzoek. Het gaat hier om de volgende twee bedrijven: usabilla.com en closealert.com.

Gebruikte zoekwoorden

De bedrijven en artikelen die op de eerste pagina werden genoemd en welke uniek zijn voor dit onderzoek zijn opgenomen als potentiële concurrenten. Met de volgende zoekwoorden is gezocht met Google:

- klantfeedback
- continue klantfeedback
- klant review systeem
- klant reviews beheren
- klant feedback
- customer feedback tool
- continue CES meten
- customer CES

Gevonden artikelen en bedrijven

Doormiddel van bovengenoemde zoekwoorden is een grote lijst met bedrijven en artikelen ontstaan:

- Storesupport	- Feedbackcompany
- Cysklantfeedback	- vrijdag online (artikel)
- Expoints	- Trustpilot
- Feeddex	- Custorate
- Marketingfacts (artikel)	- Asknicely
- Integron	- Clientheartbeat (artikel)
- Louwers en Partners	- Typeform
- Truqu	- Help Scout
- Groo	- Intercom
- Klantenvertellen.nl	- E-trusted
- Klantok	- Surveymonkey
- Kiyoh	- KCM
- Klantscores.nl	- Checkmarket (artikel)
- Klantenschrijven.nl	
- Thuiswinkel.nl	

Verfijnen van concurrenten

Uit de zoekactie zijn 24 systemen en 4 artikelen naar voren gekomen. De gevonden systemen zijn niet allemaal geschikt om verder te onderzoeken. Na deze inventarisatie zijn verschillende criteria opgesteld om de lijst verder te kunnen verfijnen. Deze criteria zijn:

- Het gaat om het uitvragen van klantfeedback over processen/ diensten;
- Het gaat niet over e-commerce producten/ ondersteuning;
- Het dient een eenvoudige opzet te hebben, laagdrempelig (De opdrachtgever verlangt dat de werking eenvoudig is);
- Het dient een Nederlands bedrijf te zijn;
- Het dient geen KTO te worden;
- Real time data in kunnen zien.

Deze criteria zijn tot stand gekomen door de huidige en toekomstige opzet van het te ontwikkelen systeem voor ogen te houden. Dit systeem biedt de mogelijkheid om feedback te vragen aan klanten na het leveren van een dienst, bijvoorbeeld een support vraag. Het systeem is niet bedoeld voor het geven van een waardering over producten en/of een bedrijf. Het geven van feedback dient eenvoudig te zijn en niet veel tijd in beslag te nemen. Ook hoeft het geen vragenlijst te zijn. De statistieken dienen real-time inzichtelijk te zijn, zoals nu. Omdat Mailplus B.V. zich voornamelijk richt op de Nederlandse markt dienen de systemen door Nederlandse bedrijven aangeboden te worden.

Na het toepassen van deze selectie criteria blijven de volgende bedrijven over, welke verder onderzocht zullen worden:

- Cysklantfeedback
- Expoints
- Feeddex
- Truqu
- KCM

Analyse methode

Omdat dit onderzoek minder gestructureerde data heeft opgeleverd en omdat het aantal concurrenten laag is, is er voor gekozen om de data te analyseren volgens horizontaal vergelijken.

Deze methode is uitermate geschikt voor het analyseren van minder gestructureerde data. In een tabelvorm worden alle antwoorden per feedback systeem op een rij gezet. In de laatste kolom staan de bevindingen. Deze bevindingen kunnen bestaan uit overeenkomsten en/of verschillen tussen de systemen.

	Cysklantfeedback	Expoints	Feeddex	Truqu	KCM	Bevindingen
<p><u>Vraag 1:</u> Op welke manier wordt de feedback uitgevraagd?</p> <p><u>Waarom:</u> Om te bepalen of de huidige manier van uitvragen uniek/afdoende is, is het van belang te weten wat de mogelijkheden zijn met andere systemen.</p>	<p>Uitvragen van feedback gebeurt door <u>4 simpele vragen</u>, welke ook nog eens worden getoond op onderdelen van het bedrijf. Onderdelen waar een klant niet mee te maken heeft gehad, komen ook niet terug in de vragenlijst. De vragen kunnen een cijfer zijn <u>(0-10) of open antwoorden</u></p>	<p>Expoints biedt verschillende vormen van clustering, specifieke condities, variabele <u>antwoordcategorieën en -schalen</u>. Ook kan er gebruik worden gemaakt van <u>open- en gesloten vragen</u> eventueel aangevuld met categorisering.</p>	<p>Uitvragen van feedback voor een medewerker is onderdeel van het geheel. Continue kun je de <u>CES en NPS</u> uitvragen. De CES werkt met een <u>11-punts schaal (0-10)</u> en de NPS met een <u>5-punts schaal (1-5)</u> uitvragen.</p>	<p>Het is een manier van 1-op-veel. <u>Feedback over verschillende onderwerpen</u>. Vragen kunnen onder andere bestaan uit: <u>openvragen, meerkeuze, keuze vragen, ratings</u>.</p>	<p>Dit pakket kan meerdere vormen uitvragen. Ze bieden o.a. klanttevredenheid, medewerkerstevredenheid, NPS, CES, Net Effort Score (NES) en Customer Journey. <u>Alle mogelijke vraagtypen worden aangeboden</u>.</p>	<p>-) zowel open als gesloten vragen zijn mogelijk -) Antwoordschalen worden veel gebruikt -) Tonen van Kengetallen/ KPI's -) Meerdere vragen, nooit een vraag</p>
<p><u>Vraag 2:</u> Op welke manier(en) kan het feedbackverzoek worden verzonden?</p> <p><u>Waarom:</u> Om inzicht te krijgen in andere mogelijkheden tot het versturen van een feedbackverzoek.</p>	<p>Per <u>mail</u> en via <u>SMS</u></p>	<p>Na het ontvangen van een data-file door Expoints wordt de feedback per <u>mail</u> verzonden. Op eigen initiatief van de klant is mogelijk via een extra module</p>	<p>Via online, mobile en <u>sociale media</u> methoden.</p>	<p>Continue via een <u>app</u>, via <u>links, QR-codes, sociale media</u> en per <u>mail</u> worden de verzoeken verzonden.</p>	<p>Per <u>e-mail, sms, telefonisch, touchscreens en QR-codes</u></p>	<p>-) Per e-mail -) Per sms -) Per URL -) Sociale media -) QR-codes</p>
<p><u>Vraag 3:</u> Welke</p>	<p>Uitgebreide</p>	<p>Real-time dashboard met</p>	<p>Uitgebreid</p>	<p>Analyses op</p>	<p>Niet duidelijke voorbeelden,</p>	<p>-) eigen dashboard</p>

<p>mogelijkheden tot analyse van de data bieden de andere systemen?</p> <p><u>Waarom:</u> Ik wil weten welke analyse mogelijkheden andere systemen bieden om een beeld te krijgen van de mogelijkheden.</p>	<p>analyse mogelijkheden met dashboard dat te personaliseren is. Prioriteiten Matrices die op basis van de KPI's en belang worden weergegeven. Mogelijkheid tot het tonen van rankings in het dashboard</p>	<p>verschillende tabbladen. 'Homepage' met belangrijke zaken waar de organisatie op wil sturen, 'Interactie module' waar alle feedback inzichtelijk wordt gemaakt en een 'rapportage set' op vrijwel iedere mogelijkheid (grafisch, tabel) om de feedback weer te geven.</p>	<p>real-time dashboard met kengetallen, grafieken en trends</p>	<p>verschillende niveau's. Per vragenlijst. Feedback per type, Requested Feedback, Pending Requests, Requests per type, Active Users, Requests per Group.</p>	<p>maar in het dashboard zijn hiervoor mogelijkheden. Belangrijke KPI's worden direct getoond. Per survey, onderzoek zijn overzichten en verdiepingen/ analyses mogelijk.</p>	<p>-) Tonen van kengetallen/ KPI's</p>
<p><u>Vraag 4:</u> Hebben de andere systemen slimme en/of extra features?</p> <p><u>Waarom:</u> Ik wil weten wat andere systemen voor slimme features hebben waarmee ze zich onderscheiden van de concurrenten.</p>	<p>Veel herhalende handelingen worden geautomatiseerd door slimme instellingen. <u>Rollen die op verschillend niveau naar de data kijken.</u> Automatisch verzenden van rapportages binnen de eigen organisatie. Mogelijkheid tot het ontvangen van <u>push mails</u> wanneer er</p>	<p><u>Opt-out</u> mogelijkheid voor ontvangers (niet meer feedback ontvangen) Mogelijkheid tot <u>feedback-block</u> (bijv. periode van 3mnd geen feedback versturen) Mogelijkheid tot herinnering <u>Eigen tone of voice en look and feel is mogelijk</u></p>	<p><u>E-mail alerts</u> bij lage NPS en/ of wanneer klant contact wenst</p>	<p>Real-time. Je kunt complimenten geven aan anderen (thumbs-up, fles champagne). <u>Werken met rollen en rechten systeem Nederlands en Engels</u> mogelijk Teams zijn mogelijk (geven en krijgen van</p>	<p>ISO 27001 standaard</p>	<p>-) Eigen look en feel -) Eigen tone of voice -) Beperken van versturen feedback verzoeken (time-out/ opt-out) -) E-mail alerts -) Closed loop feedback principe -) Rollen en rechten systeem -) Verschillende talen ondersteunen</p>

	<p>bepaalde antwoorden zijn gegeven.</p> <p>Ingebouwd ticket systeem op basis van ingevulde antwoorden op feedback verzoeken.</p> <p>Feedback kan na het oplossen van een <u>ontevredenheid</u> opnieuw worden verzonden (closed loop feedback principe)</p> <p><u>Eigen look and feel van de klant</u> (opdrachtgever)</p>			<p>feedback)</p> <p>Anoniem is mogelijk. <u>Eigen look and feel is mogelijk.</u></p>		
<p><u>Vraag 5:</u> Welke mogelijkheden tot integratie bieden de andere systemen?</p> <p><u>Waarom:</u> Ik wil deze integratie mogelijkheden weten zodat wij dit kunnen meenemen in het systeem.</p>	<p>Integratie mogelijk met iOS en Android devices d.m.v. een app.</p> <p>Integraties mogelijk met verschillende review sites.</p>	<p>Er zijn mogelijkheden tot koppelingen met CRM-, rapportage- en administratiepakketten.</p> <p>Feedback is te exporteren naar Word, Excel of PDF.</p>		<p>Export naar Excel en CSV zijn mogelijk</p>	<p>Koppelen is mogelijk met veel CRM- en Klantsoftware systemen. Er zijn bestaande koppelingen voor AFAS en KANA</p>	<p>-) Export csv en PDF</p> <p>-) Koppeling met externe systemen (API)</p>

Bevindingen

Uit het concurrenten onderzoek is gebleken dat de onderzochte systemen veelal dezelfde functionaliteiten bieden. Alle systemen beschikken over de mogelijkheid om meerdere vragen te stellen en in meerdere vormen (open-, gesloten vragen en antwoordschalen). Het verzoek om feedback te geven wordt veelal via e-mail verzonden. Cysklantfeedback en KCM bieden ook de mogelijkheid tot het versturen van een link via SMS. Truqu en KCM werken ook met de optie om QR-codes in te zetten. Feeddex en KCM bieden als enige de mogelijkheid tot het gebruik van Social Media.

Alle bekeken systemen bieden een dashboard waarin statistieken worden getoond. Sommige systemen bieden meer functionaliteiten in het dashboard dan andere. Zo biedt Cysklantfeedback een geïntegreerd ticket systeem waarmee klant contact onderhouden kan worden. Truqu biedt ook statistieken over gegeven feedback per type, verzonden feedback, uitstaande feedback verzoeken, verzoeken per type en verzoeken per groep.

Alle systemen die onderzocht zijn bieden een vorm van het aanpassen van de 'look and feel' en 'tone of voice'. Verder zijn er nog een paar slimme features die opvallen. Zo biedt Cysklantfeedback het 'closed loop feedback principe' aan. Dit houdt in dat een ontevreden waardering opnieuw verzonden kan worden. Dit kan bijvoorbeeld worden gedaan nadat er contact met de klant is geweest. Expoints biedt een 'feedback lock' en een 'opt-out' mogelijkheid voor de klanten. Hiermee kan er tijdelijk (feedback lock) of permanent (opt-out) een block worden gezet op het versturen van feedback verzoeken aan klanten. Feeddex en Cysklantfeedback kunnen meldingen geven aan de gebruiker bij een bepaalde waardering of wanneer een klant contact wenst. Bij Truqu en Cysklantfeedback kun je werken met rechten en rollen. Hierbij hebben verschillende rollen andere rechten. Truqu is het enige systeem dat heeft aangegeven dat het zowel in het Nederlands als in het Engels gebruikt kan worden. Alle systemen hebben een mogelijkheid tot het koppelen met externe systemen en/of het downloaden van statistieken als Excel- (csv) of PDF bestand.

Conclusie

Uit het onderzoek blijkt dat de onderzochte systemen ieder een eigen idee hebben van het uitvragen van klant feedback. Het ene systeem is nog uitgebreider dan het andere. Het ene bedrijf gooit met mooie termen zoals NPS, CES, Customer Journey, enzovoort. Hieruit blijkt dat de onderzochte systemen uitgebreider en specialistischer zijn dan het systeem dat Mailplus B.V. voor ogen heeft.

Er kan geconcludeerd worden dat alle systemen de mogelijkheid bieden tot het uitvragen van meerdere vragen met meerdere antwoord mogelijkheden. Wanneer er gebruik wordt gemaakt van puntenschalen als antwoordmogelijkheid zien we veelal 11-punts- en 5-punts schalen. Deze bevinding uit het concurrentie onderzoek is besproken met de opdrachtgever. Hier is uitgekomen dat Mailplus B.V. graag vasthoudt aan deze ene vraag met de 3-punts schaal. Hiervoor is gekozen omdat het systeem zoals het nu is, erg laagdrempelig is voor ontvangers om te gebruiken. Het enige dat ze hoeven te doen is het klikken op één van de drie smiley iconen en optioneel een opmerking te geven. Mede hierdoor is er een hoog response gehalte. Ook is het lastig om een 11-punts schaal (0-10) in te zetten. Dit komt doordat een 7 voor de ene persoon goed is en voor een ander persoon zeer goed. Hierdoor is het lastig te classificeren wat een gemiddeld cijfer 7 nu precies betekend.

Het versturen van het feedback verzoek doen de onderzochte systemen veelal op de zelfde manier, via e-mail. Er zijn uitzonderingen die gebruik maken van andere technieken

zoals sociale media. Het huidige systeem kan alleen via e-mail verzoeken versturen. Voor het nieuwe systeem kan het wenselijk zijn te kijken naar mogelijke uitbereidingen hierop.

De systemen hebben alleen een uitgebreid dashboard tot hun beschikking waarin de statistieken op alle mogelijke manieren getoond en met elkaar vergeleken kunnen worden. Het huidige systeem heeft een eigen gemaakt dashboard dat zichtbaar is voor de afdeling support. Een dashboard is een mogelijkheid, maar in overleg met de opdrachtgever is er besloten om dit in de eerste versie achter wegen te laten. Hier is voor gekozen om de opdracht niet onnodig ingewikkeld te maken. Om statistieken aan te bieden komt er wel een REST-API waarmee klanten deze kunnen uitvragen en kunnen tonen op de manier dat hen schikt.

In het huidige systeem is het niet mogelijk om feedback verzoeken niet te laten versturen naar klanten die deze liever niet ontvangen. Wanneer het systeem een aanvraag krijgt, wordt deze altijd verzonden. Sommige klanten hebben echter aangegeven deze niet te willen ontvangen, of dat ze van mening zijn dat ze er teveel ontvangen. Uit het onderzoek is gebleken dat er systemen op de markt zijn die hier een oplossing voor hebben. Dit is een voorbeeld van een slimme feature die mogelijk overgenomen wordt.

Bijlage V: Requirements

Business Requirements

Nummer	Omschrijving
BR01	De opdrachtgever wil haar klanten de mogelijkheid bieden om doorlopend en op eenvoudige wijze de klanttevredenheid te meten van hun eigen klanten.
BR02	De opdrachtgever wil graag een nieuwe klantgroep aanboren met het nieuwe systeem.
BR03	De opdrachtgever wil dat het systeem minder afhankelijk is van releases en updates.
BR04	De opdrachtgever wil beheer mogelijkheden in de tool zelf.
BR05	De opdrachtgever wil meer statistieken uit het systeem kunnen halen.
BR06	De opdrachtgever wil meer mogelijkheden met betrekking tot het verzenden van een feedback mail.
BR07	De klant wil de mogelijkheid tot het aanpassen van de te gebruiken teksten in het systeem.
BR08	De klant wil de mogelijkheid tot het voeren van meerdere talen.

User requirements gebaseerd op geïdentificeerde belanghebbende

Nummer	Omschrijving	Belanghebbende	Bron
UR01	Dag/Maand/Jaar rapportage op monitor is niet duidelijk. De overgang van maand naar jaar is te subtiel (alleen de cijfers veranderen en er zijn nog geen grafieken). De rapportage verbeteren en uitbreiden met eerdere resultaten.	Teammanager Support	Requirements Paul Nederlof.docx
UR02	Toevoegen operator/behandelaar binnen het proces. Nieuwe medewerkers moeten ingediend worden via TOPdesk en door escalatie naar 2de lijns worden deze door development opgepakt.	Teammanager Support	Requirements Paul Nederlof.docx
UR03	Aanpassingen aan de uitnodiging (html e-mail) moet worden ingepland met de eerstvolgende release (3-4 weken). Het	Teammanager Support	Requirements Paul Nederlof.docx

	beheer van de e-mail zouden wij liever bij onszelf willen houden		
UR04	Korte omschrijving van call toevoegen in de uitnodiging. Als een klant vaker heeft gesproken met Support kan een korte omschrijving meer duidelijkheid geven waar het gesprek over ging. De korte omschrijving wordt opgehaald van het veld "Korte omschrijving (Details)" uit TOPdesk en is een verplicht veld tijdens registratie. De waarde van het veld moet ergens in de mail geplaatst worden.	Teammanager Support	Requirements Paul Nederlof.docx
UR05	<p>"Ik ontvang jullie feedback te vaak". Soms kan het de klant ook iets teveel worden... Tussen die duizenden contacten die wij hebben uitgenodigd zitten er soms gebruikers tussen die niet zitten te wachten op een uitnodiging. Ze krijgen het te vaak zeggen ze. Dat is bij te houden in TOPdesk. Echter TOPdesk kan ons niet helpen om dan ook geen mail te sturen of die herinnering te negeren. We kunnen namelijk een melding maken bij de gebruikersgegevens maar dat valt niet altijd op. Een blacklist in de database (handmatig toegevoegd door iemand met db toegang) zou irritatie bij de klant kunnen voorkomen.</p> <p>Wij kunnen dan de uitnodiging sturen maar de laatste check vind plaats in de database. Mocht het adres voorkomen in de blacklist, dan wordt de uitnodiging niet verstuurd. Met hierbij de voorkeur dat in de rapportage ergens staat dat wij het wel zouden hebben verstuurd zodat de cijfers blijven kloppen.</p>	Teammanager Support	Requirements Paul Nederlof.docx
UR06	<p>Het tijdstip van verzenden van de getriggerde uitnodiging:</p> <p>a) moet op een vast tijdstip mogelijk zijn. Bijvoorbeeld om 20:00 's avonds naar alle contacten van die dag.</p> <p>b) met een vertraging van een xx minuten</p>	Teammanager Support	Requirements Paul Nederlof.docx
UR07	De klant kan gebruikers/behandelaars beheren. (Toevoegen, verwijderen, wijzigen)	Geïnteresseerde klanten	Requirements Paul Nederlof.docx

UR08	Het systeem kan met extra velden omgaan. M.a.w. toe te voegen in het systeem, vullen en verwerken in de mail. (Net als een databaseveld aanmaken, waarden importeren naar die velden en gebruiken in een mail als {[veldnaam]}))	Geïnteresseerde klanten	Requirements Paul Nederlof.docx
UR09	De feedback kan beheerd worden. De smileys mogen een andere naam krijgen. Zo kan je ook een andere taal kiezen (Good, Ok, Bad)	Geïnteresseerde klanten	Requirements Paul Nederlof.docx
UR10	De klant heeft de uitnodiging/mail in een omgeving waar zij zelf bij kunnen of waar een contactpersoon (of supportafdeling) bij kan voor aanpassingen	Geïnteresseerde klanten	Requirements Paul Nederlof.docx
UR11	De klant heeft de keuze om in stellen dat vooraf bepaald kan worden of de mail wel/niet verstuurd wordt en de keuze of het systeem in staat is om willekeurig de uitnodiging te versturen	Geïnteresseerde klanten	Requirements Paul Nederlof.docx
UR12	Bij een slechte beoordeling kan een mail worden gestuurd naar de behandelaar (en eventueel kopie naar de afdelingsmanager)	Geïnteresseerde klanten	Requirements Paul Nederlof.docx
UR13	Lay-out beheer (logo, kleurtjes)	Geïnteresseerde klanten	Requirements Paul Nederlof.docx
UR14	Het systeem kan een blacklist bijhouden of rekening houden met afmelders	Geïnteresseerde klanten	Requirements Paul Nederlof.docx
UR15	Rapportage vanuit het systeem of in een tool waar de standaard rapportage bijgehouden wordt. Opties zoals download naar pdf, rapportage mailen naar een e-mailadres	Geïnteresseerde klanten	Requirements Paul Nederlof.docx
UR16	Maandelijkse stats naar de beheerder en/of een lijst van e-mailadressen	Geïnteresseerde klanten	Requirements Paul Nederlof.docx
UR17	Beveiliging in het systeem om eenmalig feedback te kunnen geven en met controle om binnen (..) uur te reageren	Geïnteresseerde klanten	Requirements Paul Nederlof.docx

UR18	Landingspagina na feedback (toelichting/motivatie) is te beheren via de tool.	Geïnteresseerde klanten	Requirements Paul Nederlof.docx
UR19	Via de Rest-API is het mogelijk om de gegevens op te vragen van het feedbacksysteem	Geïnteresseerde klanten	Requirements Paul Nederlof.docx
UR20	Er is een pagina beschikbaar waarin de laatste 100 reacties bijgehouden worden. Deze kan geëxporteerd en geïmplementeerd worden op een externe pagina. Logo van Mailplus erbij zodat geïnteresseerden kunnen doorklikken naar onze website.	Geïnteresseerde klanten	Requirements Paul Nederlof.docx
UR21	Het tonen van het procentpunt verschil in good/ok/bad percentages in vergelijking met vorige periode (dag, maand, jaar)	Medewerker support	Features Sander van 't Hof
UR22	Het meesturen van de bestede tijd van een call. De waarde van de gemiddelde bestede tijd of totale bestede tijd moet getoond kunnen worden in dag/maand/jaar en hoort bij de algemene cijfers zoals Aantal feedback verstuurd en Aantal gereageerd.	Medewerker support	Features Sander van 't Hof
UR23	Het tonen van <u>Highcharts</u> grafieken met daarin percentage good/ok/bad in vergelijking met vorige periode (dag, maand, jaar)	Medewerker support	Features Sander van 't Hof
UR24	Het tonen van de naam/bedrijf/callnummer bij de betreffende smileys	Medewerker support	Features Sander van 't Hof
UR25	Totalen feedback weergeven per dag/maand/jaar. Niet alleen de percentages maar ook de absolute getallen	Medewerker support	Features Sander van 't Hof
UR26	Een beoordelaar mag één maal zijn/haar stem uitbrengen per Feedback verzoek;	Product management	Uitvragen requirements
UR27	Indien een beoordelaar opnieuw een beoordeling wil doen, wordt deze beoordeling genegeerd. Hier wordt d.m.v. een webpagina melding gemaakt;	Product management	Uitvragen requirements
UR28	Een Feedback verzoek heeft een geldigheid, bijvoorbeeld één week, wanneer er buiten deze geldigheid een	Product management	Uitvragen requirements

	beoordeling wordt verricht wordt de beoordeling genegeerd. Hier wordt d.m.v. een webpagina melding van gemaakt;		
UR29	Een Feedback verzoek kan binnen de geldigheid worden aangepast door de beoordelaar;	Product management	Uitvragen requirements
UR30	De behandelaar kan een Feedback verzoek meerdere malen versturen. Eventuele eerder gegeven beoordeling (inclusief toelichting) wordt hierdoor overschreven;	Product management	Uitvragen requirements
UR31	Indien een feedback mail bounced, wordt dit niet geregistreerd in het Feedback Systeem. Een mailer-demon melding aan de verzender is voldoende;	Product management	Uitvragen requirements
UR32	Het Feedback Systeem kan een Feedback verzoek genereren	Product management	
UR33	Het Feedback Systeem kan op de volgende manieren een Feedback verzoek genereren: per mail en via een REST-API call (voor het gebruik met Social Media).	Product management	
UR34	Een Feedback verzoek bestaat minimaal uit: <ul style="list-style-type: none"> • Gegevens van te beoordelen hulpvraag (uniek ID); • Gegevens behandelaar (Voornaam, achternaam, e-mail en nummer); • Gegevens beoordelaar (Voornaam, achternaam, e-mail en bedrijfsnaam) 	Product management	
UR35	Er kan gewaardeerd worden op de volgende manieren: 3 smiley-iconen (negatief, gewoon ok en goed) {Wellicht standaard op 3 manieren in DB 0, 1, 2 en labels in frontend, hierdoor is het mogelijk om andere namen te gebruiken}	Product management	
UR36	Het Feedback Systeem kan een gegenereerd Feedback verzoeken verzenden.	Product management	
UR37	Het Feedback Systeem kan op de volgende manieren een Feedback verzoek versturen: per mail.	Product management	

UR38	Een beoordelaar kan met één klik een beoordeling geven.	Product management	
UR39	Een beoordelaar kan een schriftelijke toelichting geven.	Product management	
UR40	De applicatie dient door meerdere gebruikers tegelijk gebruikt te kunnen worden, zonder dat er data vermenging plaats vindt.	Product management	
UR41	De webpagina's dienen responsive te zijn (wijzigen van feedback, geven van toelichting, fout pagina's, dankpagina's, geven van feedback [via link])	Medewerker support	Jos Verhaar
UR42	De API geeft de data in JSON formaat terug.	Medewerker support	Jos Verhaar
UR43	De API accepteert data in JSON formaat.	Medewerker support	Jos Verhaar

Technische beperkingen

Nummer	Omschrijving	Uitleg
TB01	De applicatie wordt geschreven in JAVA	Binnen MailPlus B.V. is dit de standaard
TB02	Als web applicatie framework wordt gebruik gemaakt van Spring MVC	Binnen MailPlus B.V. is dit de standaard
TB03	Voor de implementatie van de REST-API wordt gebruik gemaakt van Jersey (JAX-RS)	Binnen MailPlus B.V. is dit de standaard
TB04	Voor persistence (JPA) wordt gebruik gemaakt van EclipseLink	Binnen MailPlus B.V. is dit de standaard
TB05	Voor het opbouwen van de web-templates wordt Velocity gebruikt	Binnen MailPlus B.V. is dit de standaard

User requirements SMART gemaakt

Nummer	Omschrijving	UR	BR	SR
--------	--------------	----	----	----

UR01	Een beoordelaar mag één maal zijn/haar stem uitbrengen per Feedback verzoek	17,26	1	
UR02	Indien een beoordelaar opnieuw een beoordeling wil doen, wordt deze beoordeling genegeerd. Hier wordt d.m.v. een webpagina melding gemaakt	27	1	
UR03	Een Feedback verzoek heeft een geldigheid, bijvoorbeeld één week, wanneer er buiten deze geldigheid een beoordeling wordt verricht wordt de beoordeling genegeerd. Hier wordt d.m.v. een webpagina melding van gemaakt	28	1	
UR04	Een Feedback verzoek kan binnen de geldigheid worden aangepast door de beoordelaar	29	1	
UR05	De behandelaar kan een Feedback verzoek meerdere malen versturen. Eventuele eerder gegeven beoordeling (inclusief toelichting) wordt hierdoor overschreven	30	1	
UR06	Indien een feedback mail bounced, wordt dit niet geregistreerd in het Feedback Systeem. Een mailer-demon melding aan de verzender is voldoende	31	1	
UR07	De Feedback monitor toont een dag-, maand- en jaar overzicht. Er is een duidelijke overgang tussen de verschillende schermen.	1	1	
UR08	De dag-, maand- en jaar overzichten geven de cijfers van de huidige periode en die van de vorige periode (dag/ maand/ jaar).	1	5	
UR09	Het is mogelijk om behandelaren te beheren (toevoegen, bewerken en verwijderen) via het Feedback Systeem.	2, 7	6	
UR10	De tekst van een Feedback verzoek kan worden aangepast.	3, 10, 13	7	
UR11	De layout van een Feedback verzoek kan worden aangepast.	3, 10, 13	4	
UR12	Een Feedback verzoek bestaat minimaal uit: <ul style="list-style-type: none"> • Gegevens van te beoordelen hulpvraag (uniek ID en korte omschrijving); • Gegevens behandelaar (Voornaam, achternaam, e-mail en nummer); • Gegevens beoordelaar (Voornaam, achternaam, e-mail en bedrijfsnaam) 	4, 34	1	
UR13	Het is mogelijk om een Feedback verzoek te beperken voor een bepaalde beoordelaar (blacklist/ afmeldingen bijhouden (helemaal niet sturen), maximaal aantal per tijdseenheid {dag, week, maand})	5, 14	6	
UR14	Wanneer een aanvraag niet leidt tot een Feedback verzoek	5	6	

	wordt de reden hiervan opgeslagen in de database bij de bijbehorende aanvraag. (bijvoorbeeld door blacklist, missen van e-mail adres)			
UR15	Het verzendmoment van het feedback verzoek is aanpasbaar: <ul style="list-style-type: none"> • op een vasttijdstip • met een vertraging van x-aantal minuten 	6	6	
UR16	Het systeem kan met extra velden omgaan. M.a.w. toe te voegen in het systeem, vullen en verwerken in de mail. (Net als een databaseveld aanmaken, waarden importeren naar die velden en gebruiken in een mail als {[veldnaam]}).	8	7	
UR17	Het Feedback Systeem kan overweg met verschillende talen (NL en EN) { Mogelijk door het aanpasbaar maken van de teksten in de front-end}	9	8	
UR18	Het Feedback Systeem kan het versturen van een feedback verzoek wel/ niet versturen. Dit kan op basis van: <ul style="list-style-type: none"> • instellen door gebruiker • automatisch (bepaald door het systeem) 	11	6	
UR19	Wanneer er een lage feedback wordt gegeven wordt de operator (+ eventueel zijn/haar manager) hiervan per mail op de hoogte gebracht.	12	5	
UR20	Rapportage vanuit het systeem of in een tool waar de standaard rapportage bijgehouden wordt. Opties zoals download naar pdf, rapportage mailen naar een e-mailadres.	15	5	
UR21	Maandelijks statistieken naar de beheerder en/of een lijst van e-mailadressen	16	5	
UR22	De tekst van de landingspagina's kan worden aangepast.	13, 18	7	
UR23	De layout van de landingspagina's kan worden aangepast.	13, 18	4	
UR24	Via de Rest-API is het mogelijk om de statistieken op te vragen van het Feedback Systeem .	19	5	
UR25	De laatste x-aantal reacties pagina dient extern implementeerbaar te zijn als hele pagina en als (kleinere) widget.	20	5	
UR26	Vanaf deze externe implementatie kan men doorklikken naar MailPlus voor meer informatie over het Feedback Systeem .	20	5	
UR27	De dag-, maand- en jaar cijfers worden absoluut, in percentage en in (afgeronde) procentpunt getoond.	21, 25	5	
UR28	De waarde van de gemiddelde bestede- of totale bestede tijd moet getoond kunnen worden in dag/maand/jaar en hoort bij de	22	5	

	algemene cijfers zoals Aantal feedback verstuurd en Aantal gereageerd.			
UR29	De mogelijkheid tot het tonen van grafieken met daarin het percentage good/ok/bad in vergelijking met vorige periode (dag, maand en jaar)	23	5	
UR30	Bij de beoordelingen op het dag overzicht heb je de mogelijkheid tot het zien bij welk feedback verzoek deze hoort (naam beoordelaar, bedrijfsnaam en callnummer).	24	5	
UR31	Het Feedback Systeem kan een Feedback verzoek genereren	32	1	
UR32	Het Feedback Systeem kan op de volgende manieren een Feedback verzoek genereren: per mail en via een REST-API call (voor het gebruik met Social Media).	33	1	
UR33	Er kan gewaardeerd worden op de volgende manieren: 3 smiley-iconen (negatief, gewoon ok en goed) {Wellicht standaard op 3 manieren in DB 0, 1, 2 en labels in frontend, hierdoor is het mogelijk om andere namen te gebruiken}	35	1	
UR34	Het Feedback Systeem kan een gegenereerd Feedback verzoeken verzenden.	36	1	
UR35	Het Feedback Systeem kan op de volgende manieren een Feedback verzoek versturen: per mail.	37	1	
UR36	Een beoordelaar kan met één klik een beoordeling geven.	38	1	
UR37	Een beoordelaar kan een schriftelijke toelichting geven.	39	1	
UR38	De applicatie dient door meerdere gebruikers tegelijk gebruikt te kunnen worden, zonder dat er data vermenging plaats vindt.	40	2	
UR39	De webpagina's dienen responsive te zijn (wijzigen van feedback, geven van toelichting, fout pagina's, dankpagina's, geven van feedback [via link])	41	4	
UR40	De API accepteert en geeft data terug in JSON formaat.	42,43	1	

Geprioriteerde Requirements

Voor het prioriteren van de requirements wordt gebruik gemaakt van de prioriteringstechniek MoSCoW-prioritering.

Letter	Woord	Betekenis
--------	-------	-----------

M	Must have	Zonder deze requirements is het systeem onbruikbaar
S	Should have	Hoewel niet wenselijk is het mogelijk deze requirements achterwege te laten - vaak is er dan een workaround nodig
C	Could have	Deze requirements hebben een duidelijke toegevoegde waarde maar het systeem is ook bruikbaar zonder deze requirements
W	Won't have this time	Van deze requirements is afgesproken dat ze niet in de komen release geïmplementeerd zullen worden

Nummer	Omschrijving	Prioritering	Belanghebbende
UR01	Een beoordelaar mag één maal zijn/haar stem uitbrengen per Feedback verzoek	M	Product management
UR02	Indien een beoordelaar opnieuw een beoordeling wil doen, wordt deze beoordeling genegeerd. Hier wordt d.m.v. een webpagina melding gemaakt	M	Product management
UR03	Een Feedback verzoek heeft een geldigheid, bijvoorbeeld één week, wanneer er buiten deze geldigheid een beoordeling wordt verricht wordt de beoordeling genegeerd. Hier wordt d.m.v. een webpagina melding van gemaakt	M	Product management
UR04	Een Feedback verzoek kan binnen de geldigheid worden aangepast door de beoordelaar	M	Product management
UR05	De behandelaar kan een Feedback verzoek meerdere malen versturen. Eventuele eerder gegeven beoordeling (inclusief toelichting) wordt hierdoor overschreven	M	Product management
UR06	Indien een feedback mail bounced, wordt dit niet geregistreerd in het Feedback Systeem. Een mailer-demon melding aan de verzender is voldoende	M	Product management
UR07	De Feedback monitor toont een dag-, maand- en jaar overzicht. Er is een duidelijke overgang tussen de verschillende schermen.	S	Teammanager support
UR08	De dag-, maand- en jaar overzichten geven de cijfers van de huidige periode en die van de vorige periode (dag/ maand/ jaar).	S	Teammanager support

UR09	Het is mogelijk om behandelaren te beheren (toevoegen, bewerken en verwijderen) via het Feedback Systeem.	S	Teammanager support
UR10	De tekst van een Feedback verzoek kan worden aangepast.	C	Teammanager support
UR11	De layout van een Feedback verzoek kan worden aangepast.	C	Teammanager support
UR12	Een Feedback verzoek bestaat minimaal uit: <ul style="list-style-type: none"> • Gegevens van te beoordelen hulpvraag (uniek ID en korte omschrijving); • Gegevens behandelaar (Voornaam, achternaam, e-mail en nummer); • Gegevens beoordelaar (Voornaam, achternaam, e-mail en bedrijfsnaam) 	M	Teammanager support
UR13	Het is mogelijk om een Feedback verzoek te beperken voor een bepaalde beoordelaar (blacklist/ afmeldingen bijhouden (helemaal niet sturen), maximaal aantal per tijdseenheid {dag, week, maand})	C	Teammanager support
UR14	Wanneer een aanvraag niet leidt tot een Feedback verzoek wordt de reden hiervan opgeslagen in de database bij de bijbehorende aanvraag. (bijvoorbeeld door blacklist, missen van e-mail adres)	C	Teammanager support
UR15	Het verzendmoment van het feedback verzoek is aanpasbaar: <ul style="list-style-type: none"> • op een vasttijdstip • met een vertraging van x-aantal minuten 	W	Teammanager support
UR16	Het systeem kan met extra velden omgaan. M.a.w. toe te voegen in het systeem, vullen en verwerken in de mail. (Net als een databaseveld aanmaken, waarden importeren naar die velden en gebruiken in een mail als {[veldnaam]}).	W	Geïnteresseerde klanten
UR17	Het Feedback Systeem kan overweg met verschillende talen (NL en EN) { Mogelijk door het aanpasbaar maken van de teksten in de front-end}	S	Geïnteresseerde klanten
UR18	Het Feedback Systeem kan het versturen van een feedback verzoek wel/ niet versturen. Dit kan op basis van:	W	Geïnteresseerde klanten

	<ul style="list-style-type: none"> • instellen door gebruiker • automatisch (bepaald door het systeem) 		
UR19	Wanneer er een lage feedback wordt gegeven wordt de operator (+ eventueel zijn/haar manager) hiervan per mail op de hoogte gebracht.	W	Geïnteresseerde klanten
UR20	Rapportage vanuit het systeem of in een tool waar de standaard rapportage bijgehouden wordt. Opties zoals download naar pdf, rapportage mailen naar een e-mailadres.	W	Geïnteresseerde klanten
UR21	Maandelijkse statistieken naar de beheerder en/of een lijst van e-mailadressen	W	Geïnteresseerde klanten
UR22	De tekst van de landingspagina's kan worden aangepast.	C	Geïnteresseerde klanten
UR23	De layout van de landingspagina's kan worden aangepast.	C	Geïnteresseerde klanten
UR24	Via de Rest-API is het mogelijk om de statistieken op te vragen van het Feedback Systeem.	M	Geïnteresseerde klanten
UR25	De laatste x-aantal reacties pagina dient extern implementeerbaar te zijn als hele pagina en als (kleinere) widget.	S	Geïnteresseerde klanten
UR26	Vanaf deze externe implementatie kan men doorklikken naar MailPlus voor meer informatie over het Feedback Systeem.	C	Geïnteresseerde klanten
UR27	De dag-, maand- en jaar cijfers worden absoluut, in percentage en in (afgeronde) procentpunt getoond.	S	Medewerker support
UR28	De waarde van de gemiddelde bestede- of totale bestede tijd moet getoond kunnen worden in dag/maand/jaar en hoort bij de algemene cijfers zoals Aantal feedback verstuurd en Aantal gereageerd.	C	Medewerker support
UR29	De mogelijkheid tot het tonen van grafieken met daarin het percentage good/ok/bad in vergelijking met vorige periode (dag, maand en jaar).	S	Medewerker support
UR30	Bij de beoordelingen op het dag overzicht heb je de mogelijkheid tot het zien bij welk feedback verzoek deze hoort (naam	S	Medewerker support

	beoordelaar, bedrijfsnaam en callnummer).		
UR31	Het Feedback Systeem kan een Feedback verzoek genereren	M	
UR32	Het Feedback Systeem kan op de volgende manieren een Feedback verzoek genereren: per mail en via een REST-API call (voor het gebruik met Social Media).	M	
UR33	Er kan gewaardeerd worden op de volgende manieren: 3 smiley-iconen (negatief, gewoon ok en goed) {Wellicht standaard op 3 manieren in DB 0, 1, 2 en labels in frontend, hierdoor is het mogelijk om andere namen te gebruiken}	M	
UR34	Het Feedback Systeem kan een gegenereerd Feedback verzoeken verzenden.	M	
UR35	Het Feedback Systeem kan op de volgende manieren een Feedback verzoek versturen: per mail.	M	
UR36	Een beoordelaar kan met één klik een beoordeling geven.	M	
UR37	Een beoordelaar kan een schriftelijke toelichting geven.	M	
UR38	De applicatie dient door meerdere gebruikers tegelijk gebruikt te kunnen worden, zonder dat er data vermenging plaats vindt.	M	
UR39	De webpagina's dienen responsive te zijn (wijzigen van feedback, geven van toelichting, fout pagina's, dankpagina's, geven van feedback [via link])	C	
UR40	De API accepteert en geeft data terug in JSON formaat.	M	Medewerker support

Bijlage VI: Input voor requirements

Inbreng Paul Nederlof

De wensen concreet (de details volgen daaronder):

- Ik wil dat het systeem minder afhankelijk wordt van releases, updates en overige beperkingen die moeten worden doorgevoerd door o.a. development en product management. Nog steeds moeten veel verzoeken via PM ingepland worden. Dat zijn in mijn ogen onnodige verzoeken omdat deze met een beetje investering in de tool verwerkt kunnen worden en vanaf dat moment support zelf meer onderhoud kan voeren.
- Systeem als product: De klant die de uitnodiging ontvangt is enthousiast en er zijn geïnteresseerden. Momenteel zijn er 20-25 klanten geïnteresseerd. Dit aantal zal stijgen zodra we er meer over gaan vertellen en zodra het als product uitgerold kan worden.
- De laatste 100 reacties pagina op de website is nog niet beschikbaar (ook een verzoek vanuit Marketing).

Eigen wensen:

- Dag/Maand/Jaar rapportage op monitor is niet duidelijk. De overgang van maand naar jaar is te subtiel (alleen de cijfers veranderen en er zijn nog geen grafieken). De rapportage verbeteren en uitbreiden met eerdere resultaten. Sander zag dat de huidige formulieren en survey rapportage tool binnen Mailplus bijvoorbeeld hiermee gekoppeld kan worden zodat je trends kan bijhouden en dus meerdere jaren in mooie grafieken kan tonen. Is het deze maand druk ten opzichte van vorig jaar en hoeveel hebben we inmiddels aan reacties binnen in vergelijking met vorig jaar en wat was onze beste score ooit?
- Toevoegen operator/behandelaar binnen het proces. Nieuwe medewerkers moeten ingediend worden via TOPdesk en door escalatie naar 2^{de} lijns worden deze door development opgepakt.
- Aanpassingen aan de uitnodiging (html e-mail) moet worden ingepland met de eerstvolgende release (3-4 weken). Het beheer van de e-mail zouden wij liever bij onszelf willen houden
- Korte omschrijving van call toevoegen in de uitnodiging. Als een klant vaker heeft gesproken met Support kan een korte omschrijving meer duidelijkheid geven waar het gesprek over ging. De korte omschrijving wordt opgehaald van het veld “Korte omschrijving (Details)” uit TOPdesk en is een verplicht veld tijdens registratie. De waarde van het veld moet ergens in de mail geplaatst worden.
- “Ik ontvang jullie feedback te vaak”. Soms kan het de klant ook iets teveel worden... Tussen die duizenden contacten die wij hebben uitgenodigd zitten er soms gebruikers tussen die niet zitten te wachten op een uitnodiging. Ze krijgen het te vaak zeggen ze. Dat is bij te houden in TOPdesk. Echter TOPdesk kan ons niet helpen om dan ook geen mail te sturen of die herinnering te negeren. We kunnen namelijk een melding maken bij de gebruikersgegevens maar dat valt niet altijd op. Een blacklist in de database (handmatig toegevoegd door iemand met db toegang) zou irritatie bij de klant kunnen voorkomen.
Wij kunnen dan de uitnodiging sturen maar de laatste check vind plaats in de database. Mocht het adres voorkomen in de blacklist, dan wordt de uitnodiging niet verstuurd.

Met hierbij de voorkeur dat in de rapportage ergens staat dat wij het wel zouden hebben verstuurd zodat de cijfers blijven kloppen.

- Het tijdstip van verzenden van de getriggerde uitnodiging:
 - a) moet op een vast tijdstip mogelijk zijn. Bijvoorbeeld om 20:00 's avonds naar alle contacten van die dag.
 - b) met een vertraging van een xx minuten

De tool voor de klant:

Zonder dat wij hierop sturen heeft het feedbacksysteem sinds het begin bij 20+ klanten interesse gewekt en willen dus ook gebruik maken van deze tool.

Helaas voor deze klanten is de tool intern ontwikkeld en voor de Support afdeling op maat en naar wens gemaakt. Het is een lastige opgave om deze tool beschikbaar te maken voor meerdere klanten. Via deze weg wil ik toch proberen om een mogelijkheid te kunnen aanbieden voor deze klanten wat vergelijkbaar is met onze oplossing.

- De klant kan gebruikers/behandelaars beheren. (Toevoegen, verwijderen, wijzigen)
- Het systeem kan met extra velden omgaan. M.a.w. toe te voegen in het systeem, vullen en verwerken in de mail. (Net als een databaseveld aanmaken, waarden importeren naar die velden en gebruiken in een mail als `{{veldnaam}}`)
- De feedback kan beheerd worden. De smileys mogen een andere naam krijgen. Zo kan je ook een andere taal kiezen (Good, Ok, Bad)
- De klant heeft de uitnodiging/mail in een omgeving waar zij zelf bij kunnen of waar een contactpersoon (of supportafdeling) bij kan voor aanpassingen
- De klant heeft de keuze om in stellen dat vooraf bepaald kan worden of de mail wel/niet verstuurd wordt en de keuze of het systeem in staat is om willekeurig de uitnodiging te versturen
- Bij een slechte beoordeling kan een mail worden gestuurd naar de behandelaar (en eventueel kopie naar de afdelingsmanager)
- Lay-out beheer (logo, kleurtjes)
- Het systeem kan een blacklist bijhouden of rekening houden met afmelders
- Rapportage vanuit het systeem of in een tool waar de standaard rapportage bijgehouden wordt. Opties zoals download naar pdf, rapportage mailen naar een e-mailadres
- Maandelijkse stats naar de beheerder en/of een lijst van e-mailadressen
- Beveiliging in het systeem om eenmalig feedback te kunnen geven en met controle om binnen (..) uur te reageren
- Landingspagina na feedback (toelichting/motivatie) is te beheren via de tool.
- Via de Rest-API is het mogelijk om de gegevens op te vragen van het feedbacksysteem
- Er is een pagina beschikbaar waarin de laatste 100 reacties bijgehouden worden. Deze kan geëxporteerd en geïmplementeerd worden op een externe pagina. Logo van Mailplus erbij zodat geïnteresseerden kunnen doorklikken naar onze website.

Inbreng Sander van 't Hof

- Het tonen van het procentpunt verschil in good/ok/bad percentages in vergelijking met vorige periode (dag, maand, jaar) zoals in de happiness report afbeelding
- Het tonen van een omhoog/omlaag-afbeelding bij een hoger of lager percentage in vergelijking met vorige periode (dag, maand, jaar) zoals in de happiness report

afbeelding. Er zal een streepje kunnen worden getoond wanneer het verschil afgerond kleiner is dan een procentpunt.

- Het meesturen van de bestede tijd van een call. De waarde van de gemiddelde bestede tijd of totale bestede tijd moet getoond kunnen worden in dag/maand/jaar en hoort bij de algemene cijfers zoals Aantal feedback verstuurd en Aantal gereageerd
- Het tonen van Highcharts grafieken met daarin percentage good/ok/bad in vergelijking met vorige periode (dag, maand, jaar) zoals in de afbeelding Grafieken good ok bad.
- Het tonen van de naam/bedrijf/callnummer bij de betreffende smileys
- Totalen feedback weergeven per dag/maand/jaar. Niet alleen de percentages maar ook de absolute getallen

Inbreng Jos verhaar

- De API accepteert de data in JSON formaat
- De API antwoord met data in JSON formaat
- De webpagina's dienen responsive te zijn (wijzigen van feedback, geven van toelichting, fout pagina's, dankpagina's, geven van feedback [via link])
- De API calls zijn opgebouwd in een logische en voorspelbare structuur
- Het oude dashboard moet blijven werken in de nieuwe situatie
- Het nieuwe systeem dient door meerdere gebruikers gebruikt te kunnen worden, zonder dat dit problemen oplevert zoals datavermenging.
- Het is **niet** mogelijk om een operator te verwijderen, in plaats hiervan wordt deze inactief
- Voor iedere actieve operator kan er een specificatie worden getoond van de behaalde feedbacks
- Voor iedere inactieve operator wordt er geen specificatie opgehaald, maar de behaalde feedbacks worden wel mee genomen in de totalen.

Bijlage VII: Use Case beschrijvingen

01 Use Case: Verzend feedback verzoek via e-mail

naam	Verzend feedback verzoek via e-mail															
Identifier	UCD01															
Korte omschrijving	Genereren van feedback verzoek door behandelaar via e-mail. Daarna het verzenden van een feedback mail naar de beoordelaar.															
Primaire actor(en)	Gebruiker Feedback Systeem															
Secondaire actor(en)	Geen															
Precondities	De e-mail met de JSON data is goed ontvangen.															
Omschrijving	<table><tr><td>1</td><td>Als het systeem nieuwe mail detecteert.</td></tr><tr><td>1.1</td><td>Logt het systeem info met de tekst: "TopdeskMailProcess triggered @: "+new Date()+" found "+mails.length+" mails to be processed".</td></tr><tr><td>1.2</td><td>Voor iedere gevonden mail</td></tr><tr><td>1.2.1</td><td>Het systeem zet de mail om naar een Incident.</td></tr><tr><td>1.2.2</td><td>Verwerkt het systeem het incident.</td></tr><tr><td>1.2.3</td><td>Het systeem verwijdert de mail.</td></tr><tr><td>2</td><td>Anders logt het systeem een error met de tekst: "Could not find mailPath: "+mailPath+" cannot process TopdeskMails !".</td></tr></table>		1	Als het systeem nieuwe mail detecteert.	1.1	Logt het systeem info met de tekst: "TopdeskMailProcess triggered @: "+new Date()+" found "+mails.length+" mails to be processed".	1.2	Voor iedere gevonden mail	1.2.1	Het systeem zet de mail om naar een Incident.	1.2.2	Verwerkt het systeem het incident.	1.2.3	Het systeem verwijdert de mail.	2	Anders logt het systeem een error met de tekst: "Could not find mailPath: "+mailPath+" cannot process TopdeskMails !".
1	Als het systeem nieuwe mail detecteert.															
1.1	Logt het systeem info met de tekst: "TopdeskMailProcess triggered @: "+new Date()+" found "+mails.length+" mails to be processed".															
1.2	Voor iedere gevonden mail															
1.2.1	Het systeem zet de mail om naar een Incident.															
1.2.2	Verwerkt het systeem het incident.															
1.2.3	Het systeem verwijdert de mail.															
2	Anders logt het systeem een error met de tekst: "Could not find mailPath: "+mailPath+" cannot process TopdeskMails !".															
Postcondities	De feedback mail is verzonden aan de beoordelaar.															
Alternatieve flows	[01.1] Zet mail om naar incident. [01.2] Verwerk Incident.															

01.1 Use Case: Zet mail om naar incident

naam	Zet mail om naar incident										
Identificer	UCD01.1										
Korte omschrijving	Het Feedback Systeem leest de gevonden mail in en zet deze om naar een Incident Object.										
Primaire actor(en)	Het Feedback systeem										
Secondaire actor(en)	Geen										
Precondities	UCD01 is bezig en het systeem heeft een mail gevonden.										
Omschrijving	<table><tr><td>1</td><td>Deze alternatieve flow begint na stap 1.2.1 van de main flow.</td></tr><tr><td>2</td><td>Het systeem leest de mail in.</td></tr><tr><td>3</td><td>Het systeem zet de message body om naar een Incident Object.</td></tr><tr><td>4</td><td>Het systeem geeft het object terug.</td></tr></table>			1	Deze alternatieve flow begint na stap 1.2.1 van de main flow.	2	Het systeem leest de mail in.	3	Het systeem zet de message body om naar een Incident Object.	4	Het systeem geeft het object terug.
1	Deze alternatieve flow begint na stap 1.2.1 van de main flow.										
2	Het systeem leest de mail in.										
3	Het systeem zet de message body om naar een Incident Object.										
4	Het systeem geeft het object terug.										
Postcondities	Geen										
Alternatieve flows	Geen										

01.2 Use Case: Verwerk incident

naam	Verwerk incident	
Identifier	UCD01.2	
Korte omschrijving	Het Feedback Systeem verwerkt het Incident Object.	

Primaire actor(en)	Het systeem																		
Secondaire actor(en)	Geen																		
Precondities	UCD01 is bezig, het incident object dient verwerkt te worden.																		
Omschrijving	<table border="1"> <tr><td>1</td><td>Deze alternatieve flow begint na stap 1.2.2 van de main flow.</td></tr> <tr><td>2</td><td>Als het incident valide is.</td></tr> <tr><td>2.1</td><td>Als er reeds een feedback record bestaat voor het <i>unieke nummer</i>.</td></tr> <tr><td>2.1.1</td><td>Het systeem logt een info met de tekst: "Already created a feedback record for this ticketNumber: "+incident.getId()+" updating feedback record".</td></tr> <tr><td>2.1.2</td><td>Het systeem overschrijft het opgehaalde feedback record met de waarden vanuit het incident.</td></tr> <tr><td>2.1.3</td><td>Het systeem zet de vote date op NULL van het feedback record.</td></tr> <tr><td>2.1.4</td><td>Het systeem slaat het feedback record op in de database.</td></tr> <tr><td>2.2</td><td>Anders maakt het systeem een nieuw feedback record aan.</td></tr> <tr><td>2.3</td><td>De feedback mail wordt verzonden.</td></tr> </table>	1	Deze alternatieve flow begint na stap 1.2.2 van de main flow.	2	Als het incident valide is.	2.1	Als er reeds een feedback record bestaat voor het <i>unieke nummer</i> .	2.1.1	Het systeem logt een info met de tekst: "Already created a feedback record for this ticketNumber: "+incident.getId()+" updating feedback record".	2.1.2	Het systeem overschrijft het opgehaalde feedback record met de waarden vanuit het incident.	2.1.3	Het systeem zet de vote date op NULL van het feedback record.	2.1.4	Het systeem slaat het feedback record op in de database.	2.2	Anders maakt het systeem een nieuw feedback record aan.	2.3	De feedback mail wordt verzonden.
1	Deze alternatieve flow begint na stap 1.2.2 van de main flow.																		
2	Als het incident valide is.																		
2.1	Als er reeds een feedback record bestaat voor het <i>unieke nummer</i> .																		
2.1.1	Het systeem logt een info met de tekst: "Already created a feedback record for this ticketNumber: "+incident.getId()+" updating feedback record".																		
2.1.2	Het systeem overschrijft het opgehaalde feedback record met de waarden vanuit het incident.																		
2.1.3	Het systeem zet de vote date op NULL van het feedback record.																		
2.1.4	Het systeem slaat het feedback record op in de database.																		
2.2	Anders maakt het systeem een nieuw feedback record aan.																		
2.3	De feedback mail wordt verzonden.																		
Postcondities	Geen																		
Alternatieve flows	[01.2.1] Valideer incident. [01.2.2] Creëer nieuw feedback record. [01.2.3] Verzend feedback mail.																		

01.2.1 Use Case: Valideer incident

naam	Valideer incident																						
Identifier	UCD01.2.1																						
Korte omschrijving	Het systeem valideert het incident op basis van het e-mail adres van de beoordelaar.																						
Primaire actor(en)	Het systeem																						
Secondaire actor(en)	Geen																						
Precondities	UCD01.2 is bezig																						
Omschrijving	<table border="1"> <tr><td>1</td><td>Deze alternatieve flow begint na stap 2 van de main flow.</td></tr> <tr><td>2</td><td>Als er geen e-mail adres is gevonden.</td></tr> <tr><td>2.1</td><td>Logt het systeem een info met de tekst: "Invalid incident (ticketNumber="+ incident.getId() + ") : Email is null"</td></tr> <tr><td>2.2</td><td>Het systeem geeft FALSE terug.</td></tr> <tr><td>3</td><td>Als er meerdere @ worden gevonden.</td></tr> <tr><td>3.1</td><td>Logt het systeem een info met de tekst: "Invalid incident (ticketNumber="+ incident.getId() + ") : Multiple @ characters in email '"+email+"'"</td></tr> <tr><td>3.2</td><td>Het systeem geeft FALSE terug.</td></tr> <tr><td>4</td><td>Als er geen @ wordt gevonden.</td></tr> <tr><td>4.1</td><td>Logt het systeem een info met de tekst: "Invalid incident (ticketNumber="+ incident.getId() + ") : Invalid value for email '"+email+"'"</td></tr> <tr><td>4.2</td><td>Het systeem geeft FALSE terug.</td></tr> <tr><td>5</td><td>Het systeem geeft TRUE terug.</td></tr> </table>	1	Deze alternatieve flow begint na stap 2 van de main flow.	2	Als er geen e-mail adres is gevonden.	2.1	Logt het systeem een info met de tekst: "Invalid incident (ticketNumber="+ incident.getId() + ") : Email is null"	2.2	Het systeem geeft FALSE terug.	3	Als er meerdere @ worden gevonden.	3.1	Logt het systeem een info met de tekst: "Invalid incident (ticketNumber="+ incident.getId() + ") : Multiple @ characters in email '"+email+"'"	3.2	Het systeem geeft FALSE terug.	4	Als er geen @ wordt gevonden.	4.1	Logt het systeem een info met de tekst: "Invalid incident (ticketNumber="+ incident.getId() + ") : Invalid value for email '"+email+"'"	4.2	Het systeem geeft FALSE terug.	5	Het systeem geeft TRUE terug.
1	Deze alternatieve flow begint na stap 2 van de main flow.																						
2	Als er geen e-mail adres is gevonden.																						
2.1	Logt het systeem een info met de tekst: "Invalid incident (ticketNumber="+ incident.getId() + ") : Email is null"																						
2.2	Het systeem geeft FALSE terug.																						
3	Als er meerdere @ worden gevonden.																						
3.1	Logt het systeem een info met de tekst: "Invalid incident (ticketNumber="+ incident.getId() + ") : Multiple @ characters in email '"+email+"'"																						
3.2	Het systeem geeft FALSE terug.																						
4	Als er geen @ wordt gevonden.																						
4.1	Logt het systeem een info met de tekst: "Invalid incident (ticketNumber="+ incident.getId() + ") : Invalid value for email '"+email+"'"																						
4.2	Het systeem geeft FALSE terug.																						
5	Het systeem geeft TRUE terug.																						
Postcondities	Geen																						
Alternatieve flows	Geen																						

01.2.2 Use Case: Creëer nieuw feedback record

naam	Creëer nieuw feedback record														
Identifier	UCD01.2.2														
Korte omschrijving	Het systeem maakt een nieuw feedback record aan en slaat deze op in de database.														
Primaire actor(en)	Het systeem														
Secondaire actor(en)	Geen														
Precondities	UCD01.2 is bezig, er dient een nieuw feedback record te worden gemaakt.														
Omschrijving	<table border="1"> <tr><td>1</td><td>Deze alternatieve flow begint na stap 2.2 van de main flow</td></tr> <tr><td>2</td><td>Het systeem genereert een <i>encl.d.</i></td></tr> <tr><td>3</td><td>Het systeem maakt een nieuw feedbackDao Object aan.</td></tr> <tr><td>4</td><td>Het systeem set het encl.d.</td></tr> <tr><td>5</td><td>Het systeem set de andere velden.</td></tr> <tr><td>6</td><td>Het systeem slaat dit feedbackDao Object op in de database.</td></tr> <tr><td>7</td><td>Het systeem geeft dit object terug</td></tr> </table>	1	Deze alternatieve flow begint na stap 2.2 van de main flow	2	Het systeem genereert een <i>encl.d.</i>	3	Het systeem maakt een nieuw feedbackDao Object aan.	4	Het systeem set het encl.d.	5	Het systeem set de andere velden.	6	Het systeem slaat dit feedbackDao Object op in de database.	7	Het systeem geeft dit object terug
1	Deze alternatieve flow begint na stap 2.2 van de main flow														
2	Het systeem genereert een <i>encl.d.</i>														
3	Het systeem maakt een nieuw feedbackDao Object aan.														
4	Het systeem set het encl.d.														
5	Het systeem set de andere velden.														
6	Het systeem slaat dit feedbackDao Object op in de database.														
7	Het systeem geeft dit object terug														
Postcondities	Er is een nieuw record in de database toegevoegd.														
Alternatieve flows	Geen														

01.2.3 Use Case: Verzend feedback mail

naam	Verzend feedback mail																		
Identifier	UCD01.2.3																		
Korte omschrijving																			
Primaire actor(en)	Het systeem																		
Secondaire actor(en)	Geen																		
Precondities	UCD01.2 is bezig, de feedback mail dient verzonden te worden.																		
Omschrijving	<table border="1"> <tr><td>1</td><td>Deze alternatieve flow begint na stap 2.3 van de main flow</td></tr> <tr><td>2</td><td>Het systeem bouwt de HTML versie van de mail op basis van parameters en de HTML template.</td></tr> <tr><td>3</td><td>Het systeem bouwt de Tekst versie van de mail op basis van parameters en de Tekst template.</td></tr> <tr><td>4</td><td>Het systeem vervangt parameters in het onderwerp met waarden uit het feedback record.</td></tr> <tr><td>5</td><td>De mail wordt verzonden naar de beoordelaar.</td></tr> <tr><td>5.1</td><td>De opgebouwde feedback mail wordt omgezet naar een echte mail.</td></tr> <tr><td>5.2</td><td>Het systeem voegt het aan- en van-adres, onderwerp en het bericht toe.</td></tr> <tr><td>5.3</td><td>De feedback mail wordt verzonden.</td></tr> <tr><td>6</td><td>Het systeem logt een info met de tekst: "send feedbackMail for ticketNumber: "+feedback.getTicketNumber()+" to email: "+feedback.getRequestorEmail().</td></tr> </table>	1	Deze alternatieve flow begint na stap 2.3 van de main flow	2	Het systeem bouwt de HTML versie van de mail op basis van parameters en de HTML template.	3	Het systeem bouwt de Tekst versie van de mail op basis van parameters en de Tekst template.	4	Het systeem vervangt parameters in het onderwerp met waarden uit het feedback record.	5	De mail wordt verzonden naar de beoordelaar.	5.1	De opgebouwde feedback mail wordt omgezet naar een echte mail.	5.2	Het systeem voegt het aan- en van-adres, onderwerp en het bericht toe.	5.3	De feedback mail wordt verzonden.	6	Het systeem logt een info met de tekst: "send feedbackMail for ticketNumber: "+feedback.getTicketNumber()+" to email: "+feedback.getRequestorEmail().
1	Deze alternatieve flow begint na stap 2.3 van de main flow																		
2	Het systeem bouwt de HTML versie van de mail op basis van parameters en de HTML template.																		
3	Het systeem bouwt de Tekst versie van de mail op basis van parameters en de Tekst template.																		
4	Het systeem vervangt parameters in het onderwerp met waarden uit het feedback record.																		
5	De mail wordt verzonden naar de beoordelaar.																		
5.1	De opgebouwde feedback mail wordt omgezet naar een echte mail.																		
5.2	Het systeem voegt het aan- en van-adres, onderwerp en het bericht toe.																		
5.3	De feedback mail wordt verzonden.																		
6	Het systeem logt een info met de tekst: "send feedbackMail for ticketNumber: "+feedback.getTicketNumber()+" to email: "+feedback.getRequestorEmail().																		
Postcondities	De feedback mail is verzonden aan de requestor.																		
Alternatieve flows	Geen																		

02 Use Case: Verzend feedback verzoek via Rest-API call

naam	Verzend feedback verzoek via Rest-API call														
Identifier	UCD02														
Korte omschrijving	Het systeem verzendt een feedback mail die is aangemaakt via een Rest-API call.														
Primaire actor(en)	Extern systeem														
Secondaire actor(en)	Geen														
Precondities	Er is via oAuth een succesvolle verbinding gemaakt.														
Omschrijving	<table border="1"> <tr> <td>1</td><td>Het externe systeem doet de Rest-API call post /feedback met de volgende opbouw in de body: <pre>{ "id": "[id]", "shortDescription": "[shortDescription]", "requestor": { "firstname": "[requestorFirstName]", "lastname": "[requestorLasName]", "account": "[requestorAccountName]", "email": "[requestorEmail]" }, "operator": { "firstname": "[operatorFirstName]", "lastname": "[operatorLastName]", "number": [operatorNumber] } }</pre> </td></tr> <tr> <td>2</td><td>Het systeem logt een info met de tekst: "post /feedback triggered @: " +new Date()+ " with id: "+body.id+"."</td></tr> <tr> <td>3</td><td>Als het request valide is.</td></tr> <tr> <td>3.2</td><td>Het systeem verwerkt het incident.</td></tr> <tr> <td>3.3</td><td>Het systeem geeft een succesmelding terug.</td></tr> <tr> <td>4</td><td>Het systeem logt een error met de tekst: "request is not valid, there is no feedback created."</td></tr> <tr> <td>4.1</td><td>Het systeem geeft een foutmelding terug met de reden.</td></tr> </table>	1	Het externe systeem doet de Rest-API call post /feedback met de volgende opbouw in de body: <pre>{ "id": "[id]", "shortDescription": "[shortDescription]", "requestor": { "firstname": "[requestorFirstName]", "lastname": "[requestorLasName]", "account": "[requestorAccountName]", "email": "[requestorEmail]" }, "operator": { "firstname": "[operatorFirstName]", "lastname": "[operatorLastName]", "number": [operatorNumber] } }</pre>	2	Het systeem logt een info met de tekst: "post /feedback triggered @: " +new Date()+ " with id: "+body.id+"."	3	Als het request valide is.	3.2	Het systeem verwerkt het incident.	3.3	Het systeem geeft een succesmelding terug.	4	Het systeem logt een error met de tekst: "request is not valid, there is no feedback created."	4.1	Het systeem geeft een foutmelding terug met de reden.
1	Het externe systeem doet de Rest-API call post /feedback met de volgende opbouw in de body: <pre>{ "id": "[id]", "shortDescription": "[shortDescription]", "requestor": { "firstname": "[requestorFirstName]", "lastname": "[requestorLasName]", "account": "[requestorAccountName]", "email": "[requestorEmail]" }, "operator": { "firstname": "[operatorFirstName]", "lastname": "[operatorLastName]", "number": [operatorNumber] } }</pre>														
2	Het systeem logt een info met de tekst: "post /feedback triggered @: " +new Date()+ " with id: "+body.id+"."														
3	Als het request valide is.														
3.2	Het systeem verwerkt het incident.														
3.3	Het systeem geeft een succesmelding terug.														
4	Het systeem logt een error met de tekst: "request is not valid, there is no feedback created."														
4.1	Het systeem geeft een foutmelding terug met de reden.														
Postcondities	<ul style="list-style-type: none"> Er is een feedback record aangemaakt; Er is een feedback mail verzonden aan de requestor 														
Alternatieve flows	[02.1] Valideer Rest-API request. [01.2] Verwerk Incident.														

02.1 Use Case: Valideer Rest-API request

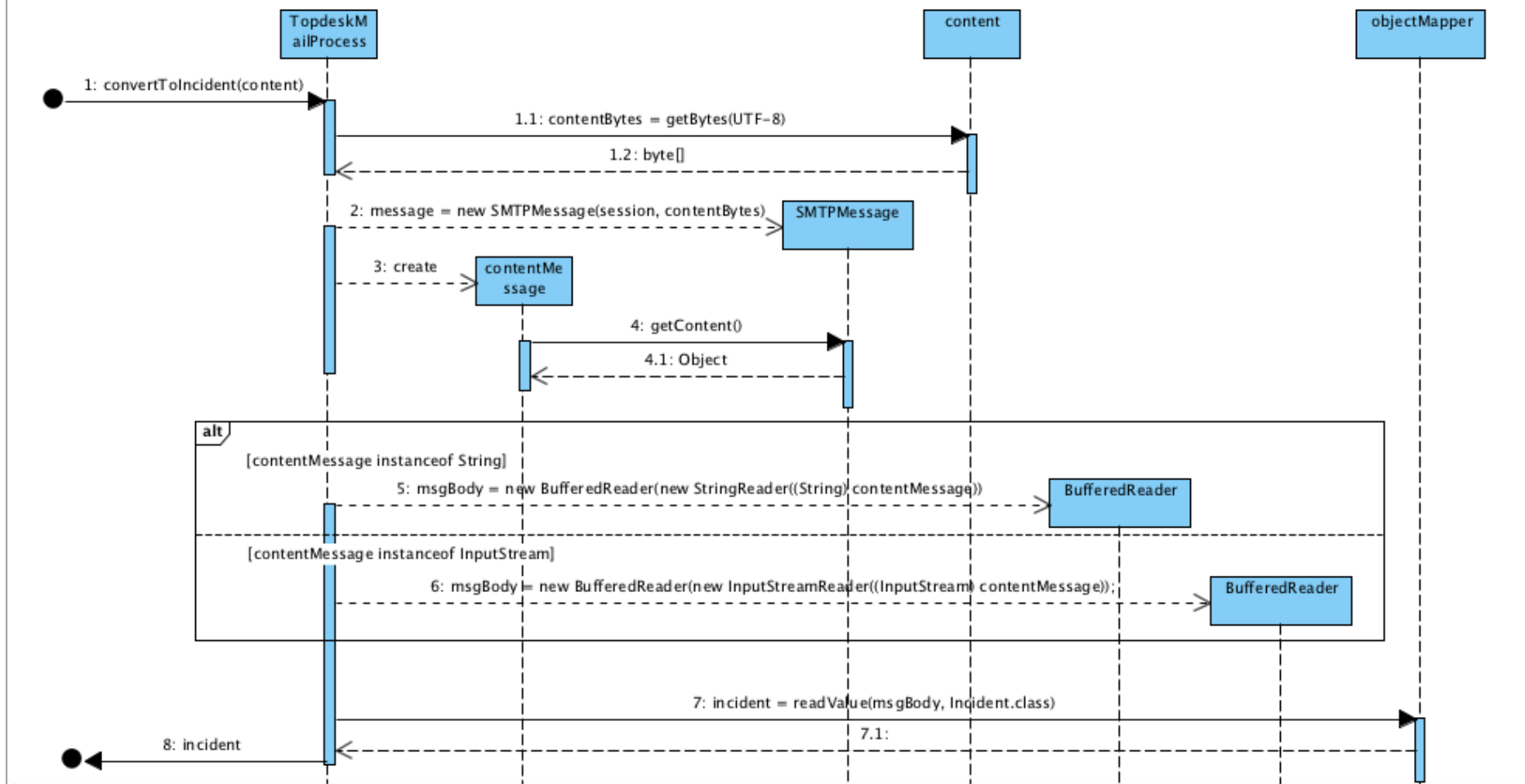
naam	Valideer Rest-API request		
Identifier	UCD02.1		
Korte omschrijving	Het systeem controleert of de verplichte velden aanwezig zijn.		
Primaire actor(en)	Het systeem		
Secondaire actor(en)	Geen		
Precondities	UCD02 is bezig en het Rest-API request dient gevalideerd te worden.		
Omschrijving	<table border="1"> <tr> <td>1</td><td>Deze alternatieve flow begint op stap 3 van de mainflow.</td></tr> </table>	1	Deze alternatieve flow begint op stap 3 van de mainflow.
1	Deze alternatieve flow begint op stap 3 van de mainflow.		

	2	Als er geen <i>id</i> is gevonden.
	2.1	Logt het systeem een info met de tekst: "Invalid incident : ticketNumber is null."
	2.2	Het systeem geeft FALSE terug.
	3	Als er geen <i>account</i> is gevonden.
	3.1	Logt het systeem een info met de tekst: "Invalid incident (ticketNumber=" + incident.getId() + ") : account is null.".
	3.2	Het systeem geeft FALSE terug.
	4	Als er geen <i>operator.number</i> is gevonden.
	4.1	Logt het systeem een info met de tekst: "Invalid incident (ticketNumber="+incident.getId()+") : operator.number is null.".
	4.2	Het systeem geeft FALSE terug.
	5	Het systeem geeft TRUE terug.
Postcondities		
Alternatieve flows		

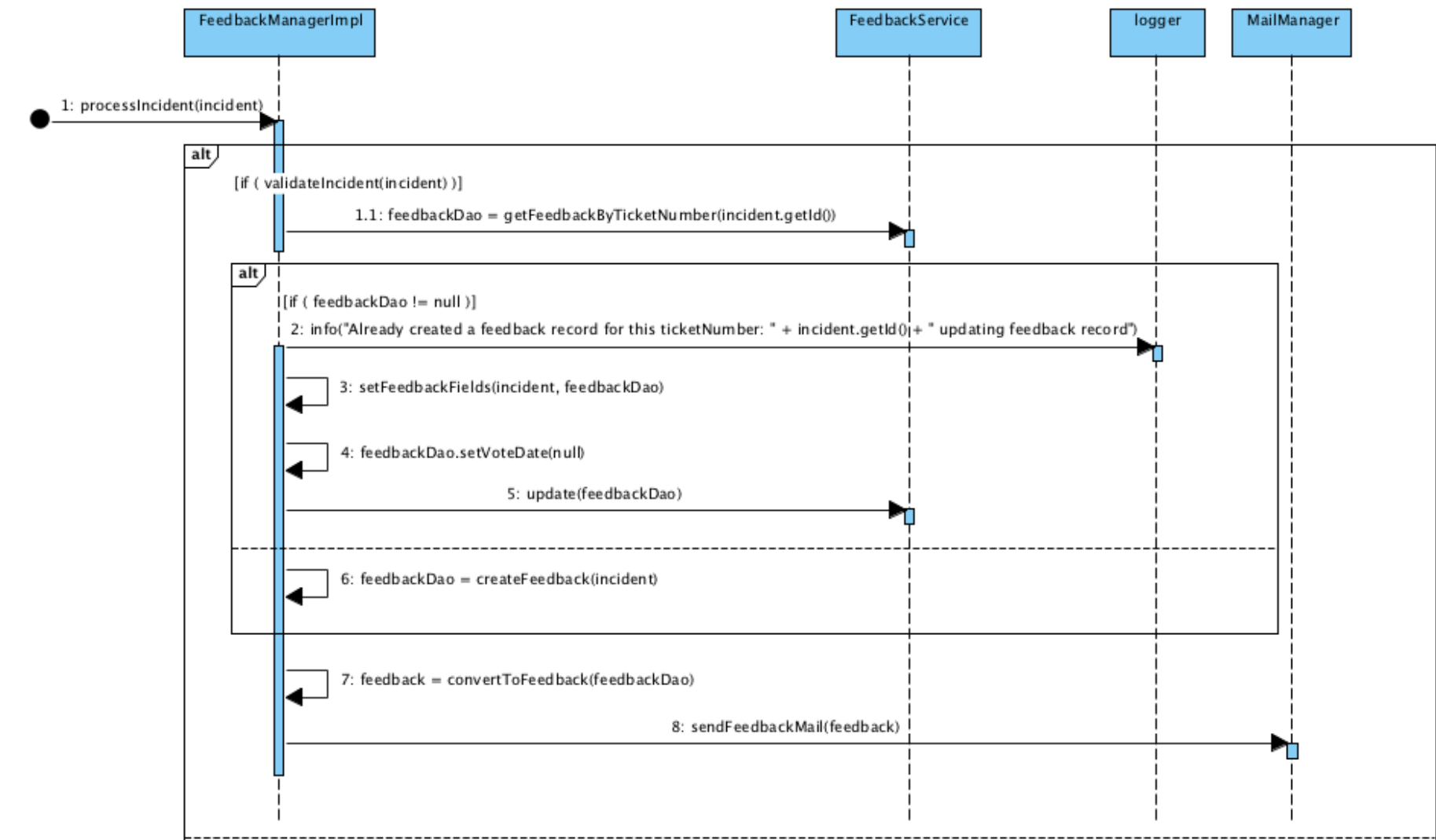
Bijlage VIII: Sequentie Diagrammen huidige implementatie



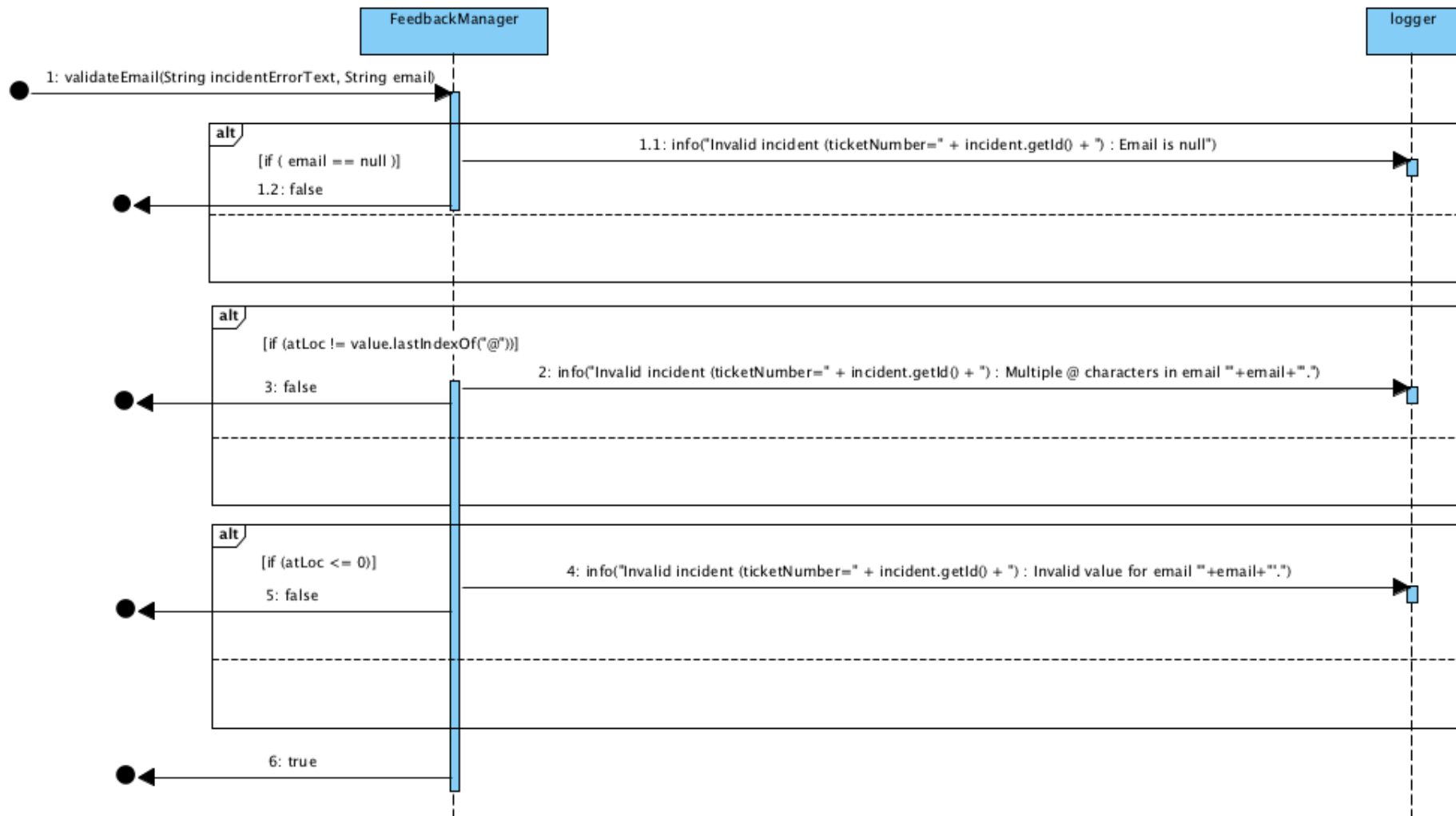
sd 01.1 Zet mail om naar incident



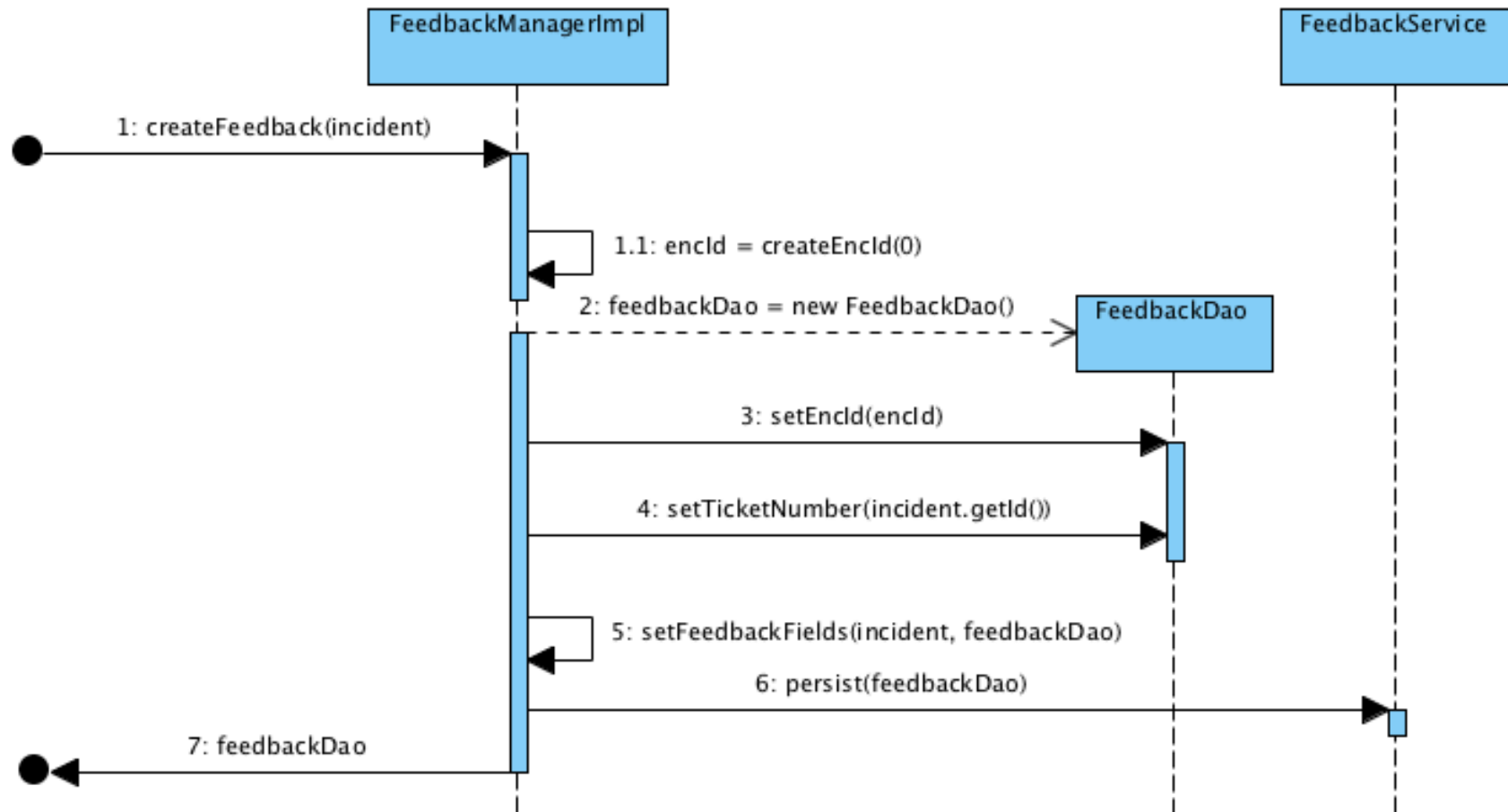
sd 01.2 Verwerk Incident



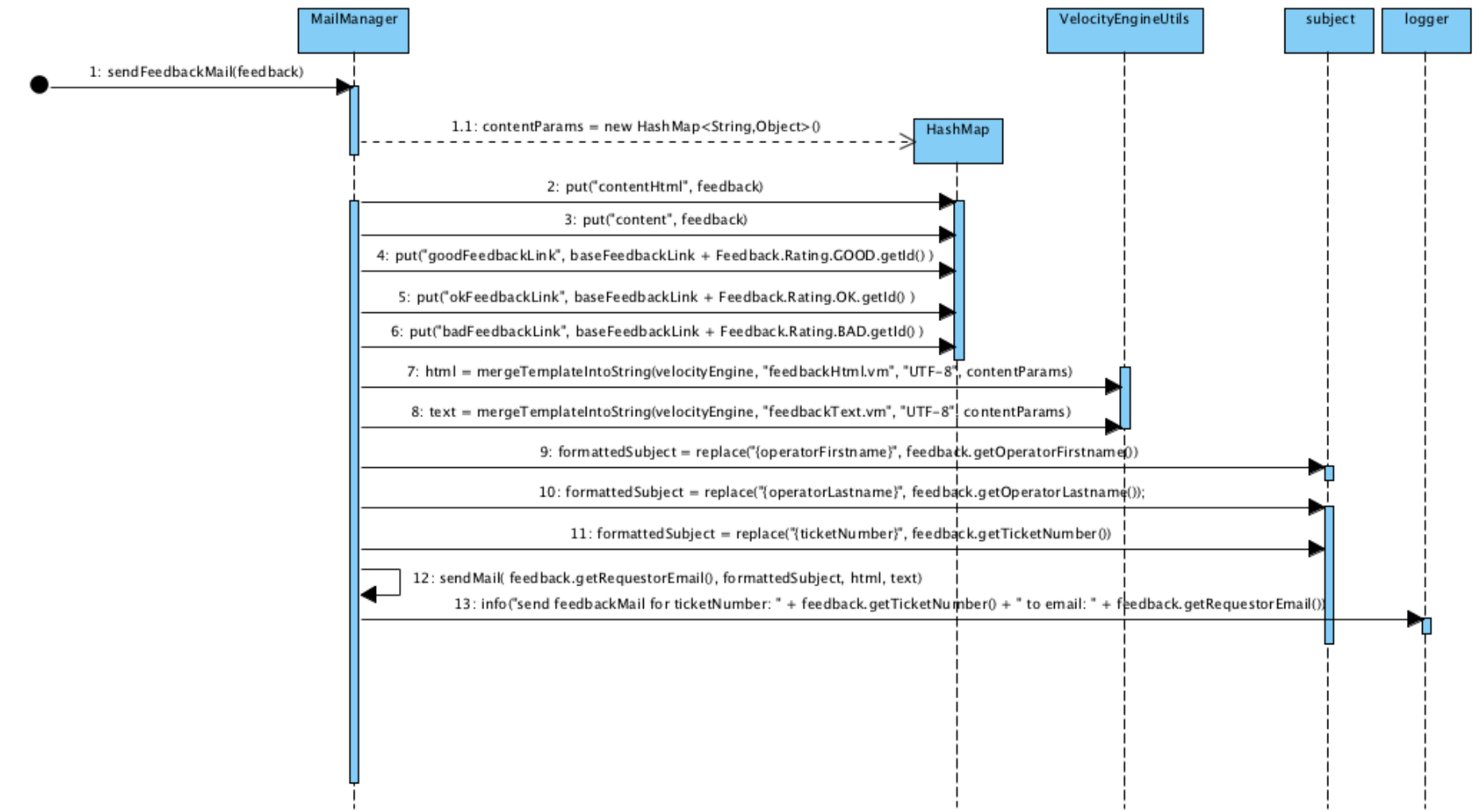
sd 01.2.1 Valideer incident



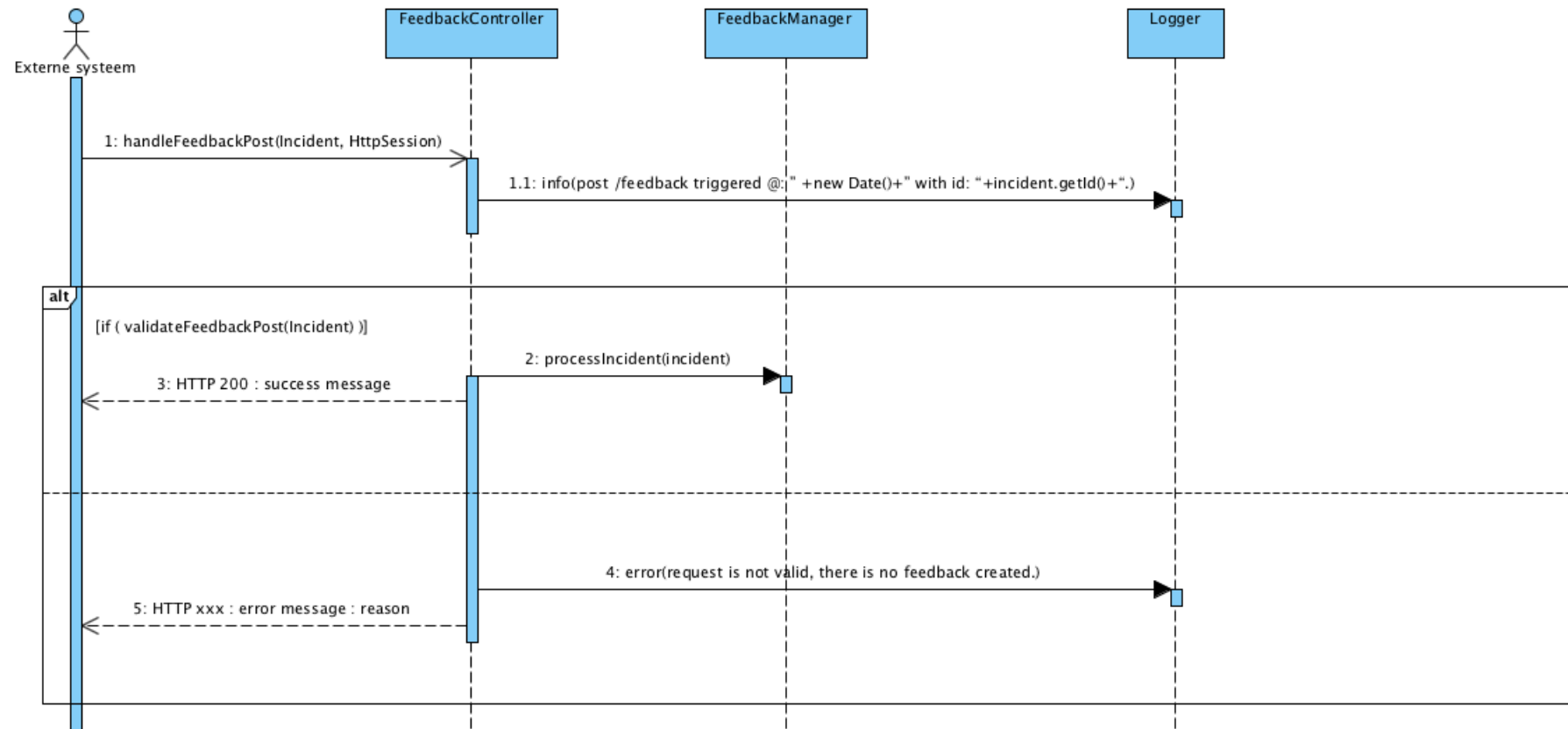
sd 01.2.2 Creëer nieuw feedback record



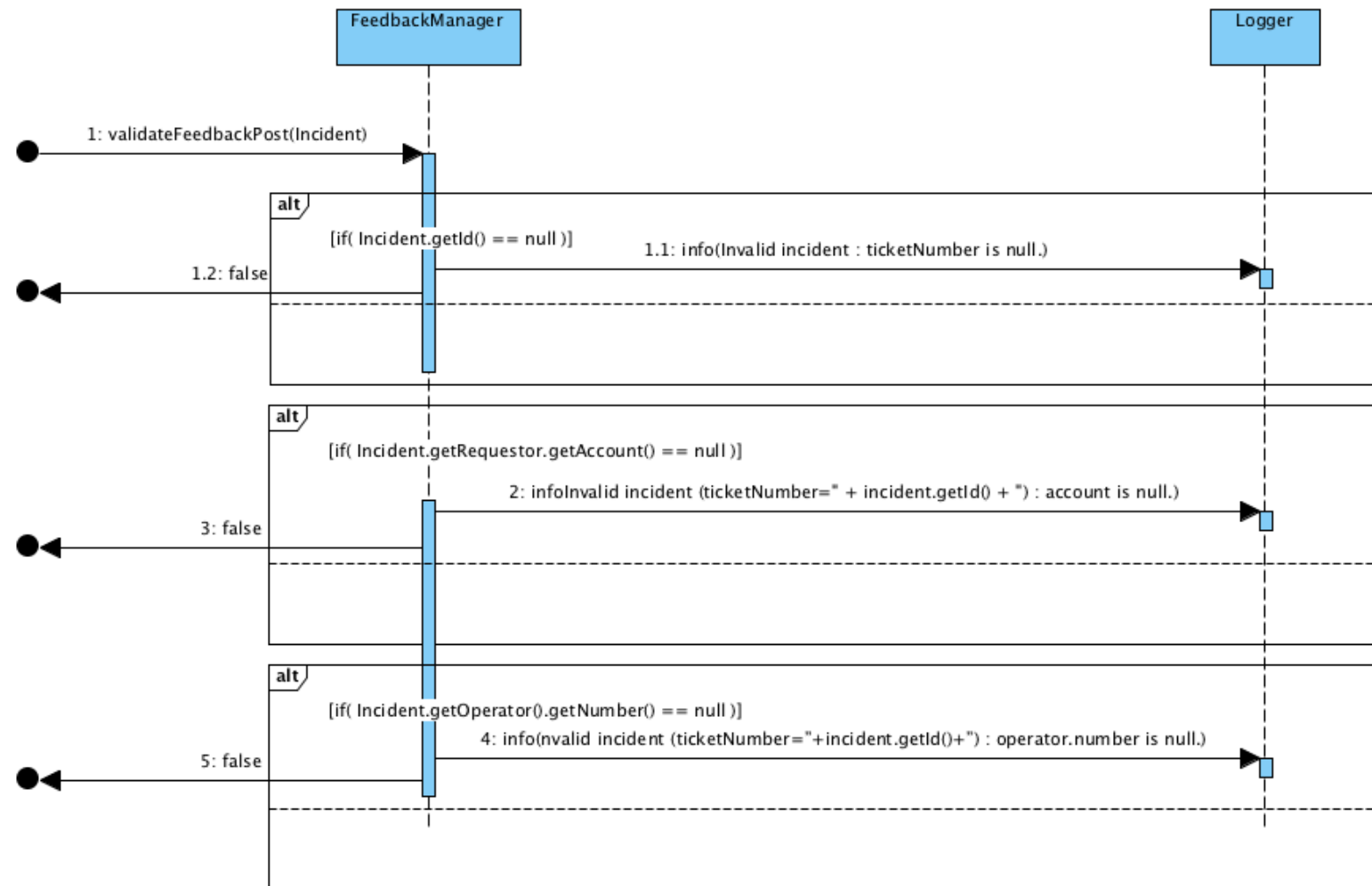
sd 01.2.3 Verzend feedback mail



sd 02 Feedback verzoek verzenden via Rest API call



sd 02.1 Valideer Rest API request



Bijlage IX: Testplan

Testbasis

System Under Test (SUT)

Tijdens het uitvoeren van de tests wordt niet het gehele systeem getest. De nadruk ligt vooral op de toegevoegde waarde. Dit is Rest-API van de applicatie. De MVC kant van het systeem, wat voornamelijk de frontend behelst, valt buiten beschouwing. Het SUT bevat alle toegevoegde functionaliteiten en de Rest-API.

De testbasis voor de uit te voeren tests bestaat uit verschillende documenten die in het afstudeerverslag aan bod komen. Je kunt hierbij denken aan:

- Use Case Beschrijvingen
- Klassendiagram huidig ontwerp
- Beslissingspunten diagrammen
- De geschreven code

Teststrategie

De doel van het testen van de ontwikkelde code is om aan te tonen dat de functionaliteit doet wat beoogd is. Door het maken van een testboom wordt in kaart gebracht welke onderdelen getest gaan worden. Deze is opgesteld tijdens de testrisicoanalyse. In afbeelding 53 wordt de testboom weergegeven.

Tijdens het opstellen van de testrisicoanalyse is gekeken naar de functionaliteiten van het SFS die inzichtelijk zijn gemaakt in de testboom. De functionaliteiten zijn ingedeeld naar belangrijkheid. Functionaliteiten die belangrijk zijn voor het systeem worden meer uitgebreid getest. Je kan hierbij denken aan meer verschillende tests die worden uitgevoerd, maar ook naar een grotere test diepte. Dit houdt in dat er meer testgevallen zijn opgesteld, waardoor er met meer zekerheid kan worden gezegd dat een functionaliteit doet wat er beoogd is.

Voor het testen van het SFS wordt gebruik gemaakt van verschillende testsoorten. Zo worden Unit Testen (hierna: UT), ook bekend als module tests, ingezet om de losse componenten van het systeem te testen. Met UT worden elementaire blokken van het SFS afzonderlijk getest. Deze test verlopen volledig geautomatiseerd zodat deze hergebruikt kunnen worden in de toekomst.

Naast UT wordt er ook gebruik gemaakt van Systeemtesten (hierna: ST). Dit zijn tests die de koppeling tussen verschillende onderdelen test. In de UT is aangetoond dat de onderdelen op zichzelf goed werken, maar werken ze ook goed samen?

Bij het opstellen van de logische testgevallen is gekeken naar de test soort waarvoor deze testgevallen gebruikt gaan worden. Aan de hand hiervan zijn verschillende testtechnieken geselecteerd. Voor de UT zijn Syntax testen geschreven. Met behulp van deze tests wordt gecontroleerd hoe het SFS reageert op valide en niet valide inputwaarden. Voor de ST zijn ook deze Syntax testen ingezet, maar zijn daarnaast ook Algoritme Testen (hierna: AT) met verschillende test diepten en Cause Effect graphing (hierna: CE) ingezet. CE testen zijn beter bekend als beslissingstabellen.

De geformuleerde logische testgevallen hebben als basis gediend voor de fysieke testgevallen. Deze zijn per testtechniek verder uitgewerkt volgens een vast stramien. Dit stramien zorgt ervoor dat de opzet voor ieder testgeval gelijk is en dat deze terug te traceren zijn tot de testboom.

I-D testrisicoanalyse

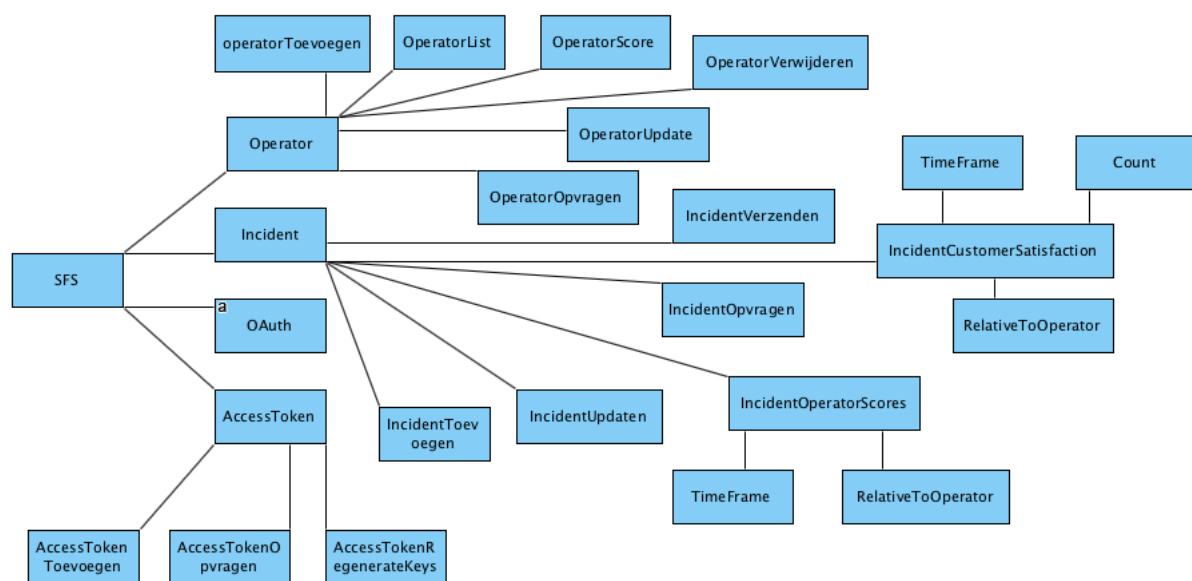
De uitvoering van de testrisicoanalyse bestaat uit een aantal stappen. Als eerste is de testboom opgesteld om de functies en aandachtspunten van het SFS in kaart te brengen.

Voor de aandachtspunten en functies die uit de testboom naar voren zijn gekomen is vervolgens het relatief belang bepaald. Hierin wordt aangegeven hoe belangrijk een bepaalde functie is voor het SFS.

Aan de hand van het relatieve belang zijn de risicocategorieën opgesteld. Hier wordt bepaald welke technieken en test diepte er voor welke functionaliteit wordt ingezet. Hiermee wordt bepaald hoe uitgebreid een functie getest gaat worden.

Testboom

Voor de testboom is het SFS onderverdeeld in hoofd- en subfuncties. Een hoofdfunctie is een verzameling van subfuncties. De hoofdfuncties in de testboom zijn: Operator, AccessToken, Incident en OAuth. Deze hoofdfuncties zijn verder onderverdeeld in de getoonde subfuncties.



Afbeelding 53: Testboom

Bepalen relatief belang

Voor de subfuncties die zijn bepaald met de testboom is het relatieve belang bepaald. Dit is gedaan door de student (verdeling 1) en de product owner (verdeling 2). Ieder heeft maximaal 69 punten gekregen om te verdelen over de verschillende subfuncties. De verdeling is te zien in onderstaande tabel.

Risicogebied	Verdeling 1	Verdeling 2	Totaal
Operator-Toevoegen	5	5	10
Operator-Updaten	5	5	10
Operator-Opvragen	1	3	4
Operator-Verwijderen	1	1	2
Operator-Get-List	1	1	2

Operator-Operator-Score	1	1	2
Gebruiker-Toevoegen	5	5	10
Gebruiker-Opvragen	1	3	4
Gebruiker-Regenerate-Keys	3	3	6
Incident-Toevoegen	5	5	10
Incident-Update	5	5	10
Incident-Opvragen	5	5	10
Incident-Versturen	5	5	10
Incident-Customer-Satisfaction-Timeframe	3	1	4
Incident-Customer-Satisfaction-Aantal	1	1	2
Incident-Customer-Satisfaction-Relative	3	1	4
Incident-Customer-Satisfaction-Relative-Operator	1	1	2
Incident-Operator-Scores-Timeframe	3	1	4
Incident-Operator-Scores-Relative	3	1	4
Incident-Operator-Scores-Relative-Operator	1	1	2
Incident-Live-Scores-Timeframe	1	1	2
Incident-Remarks-Timeframe	1	1	2
OAuth	9	5	14
Totaal	69	61	130

[tabel verdeling relatief belang]

Bepalen risicocategorieën

De verdeling van de subfuncties naar relatief belang levert de input voor het bepalen van de risicocategorieën. Deze zijn onderverdeeld naar aanleiding van de volgende verdeling.

Risicocategorie	Inhoud
Kritisch	Belangrijkste 10% van de risicogebieden
Hoog	Daaropvolgende 20% van de risicogebieden
Midden	Daaropvolgende 30% van de risicogebieden
Laag	Minst belangrijke 40% van de risicogebieden

[tabel verdeling risicocategorieën, bron: tabel 8.4, blz 100]

Het verdelen van de subfuncties gaat op aantal. Onder kritisch vallen 10% van de subfuncties. Voor het SFS zijn dit de drie functionaliteiten met de hoogste score. De vijf functionaliteiten met de hoogste score daarna, vallen in de categorie hoog. De zeven functionaliteiten met de hoogste score daarna, vallen in de categorie midden. De laatste acht functionaliteiten vallen in de categorie laag. Aan de hand van de hiervoor genoemde stappen is de volgende verdeling van de subfuncties gemaakt over de risicocategorieën.

Risicocategorie	Risicogebied	Relatief belang
Kritisch (10)	OAuth	14
	Incident-Toevoegen	10
	Incident-Update	10
Hoog (8)	Incident-Opvragen	10
	Incident-Versturen	10
	Gebruiker-Toevoegen	10
	Operator-Update	10
	Operator-Toevoegen	10
Midden (6)	Gebruiker-Regenerate-Keys	6
	Operator-Opvragen	4
	Gebruiker-Opvragen	4
	Incident-Customer-Satisfaction-Timeframe	4
	Incident-Customer-Satisfaction-Relative	4
	Incident-Operator-Scores-Timeframe	4
	Incident-Operator-Scores-Relative	4
Laag (2)	Operator-Verwijderen	2
	Operator-Get-List	2
	Operator-Operator-Score	2
	Incident-Customer-Satisfaction-Aantal	2
	Incident-Customer-Satisfaction-Relative-Operator	2
	Incident-Operator-Scores-Relative-Operator	2
	Incident-Live-Scores-Timeframe	2

	Incident-Remarks-Timeframe	2
--	----------------------------	---

[tabel verdeling (sub)functies over de risicocategorieën]

Nadat de risicocategorieën zijn verdeeld is bepaald met welke testtechnieken en test diepte de functies worden getest. Dit is terug te vinden in onderstaande tabel. Eerst wordt de risicocategorie genoemd, de kolommen ernaast geven de test domeinen: Invoer en Functionele afhandeling genoemd.

Risicocategorie	Invoer	Functionele afhandeling
Kritisch	Syntax: (niet) valide	AT branche coverage + CE
Hoog	Syntax: (niet) valide	AT branche coverage
Midden	Syntax: (niet) valide	AT statement coverage
Laag	Syntax: valide	

[Geselecteerde testtechnieken per test domein en risicocategorie]

In de tabel is te zien dat voor een laag risico functie alleen valide syntax wordt getest op de invoer. Voor een midden risico functie wordt dit uitgebreid met ook de niet valide tests. Hoe hoger in de tabel genoemd, hoe hoger de test sterkte is.

Producten

Logische testgevallen

Aan de hand van de bovenstaande tabel zijn de logische testgevallen opgesteld. Deze zijn per risicocategorie en per testtechniek doorlopen, beginnend met de risicocategorie laag. Nadat de logische gevallen voor de syntax testen waren bepaald zijn de logische testgevallen voor de AT testen van de midden risicocategorie bepaald, enz.

Syntax testen

SYN.T01: *Operator-Verwijderen* [*deleteOperator*]

Stap 1: Bepaling van de attribuutrestricties

operatorEncId: 35 Characters, Verplicht

Stap 2: Bepaal geldige en ongeldige waarden

	Geldig	Ongeldig
operatorEncId	35 Char	> 35
		< 35
		“”
		NULL

Stap 3: Bepaal testgevallen

	Geval 1
operatorEncId	35 Char
Resultaat	Geen error

SYN.T02: Operator-Get-List [*getOperatorList*]

Stap 1: Bepaling van de attribuutrestricties

userId: Integer, Verplicht

Stap 2: Bepaal geldige en ongeldige waarden

	Geldig	Ongeldig
userId	Integer	niet Integer
	>= 1000	<1000
		NULL

Stap 3: Bepaal testgevallen

	Geval 1
userId	Integer >= 1000
Resultaat	Geen error

SYN.T03: Operator-Operator-Score [*getOperatorScoreByTimeFrame*]

Stap 1: Bepaling van de attribuutrestricties

operatorEncId: 35 Characters, Verplicht

timeFrame: enum TimeFrame, Verplicht

Stap 2: Bepaal geldige en ongeldige waarden

	Geldig	Ongeldig
operatorEncId	35 Char	> 35
		< 35
		“”
		NULL

timeFrame	DAY, MONTH, Q1, Q2, Q3, Q4, YEAR	NULL
		niet DAY, MONTH, Q1, Q2, Q3, Q4, YEAR

Stap 3: Bepaal testgevallen

	Geval 1	Geval 2	Geval 3	Geval 4
operatorEncId	35 Char	35 Char	35 Char	35 Char
timeFrame	DAY	MONTH	Q1	Q2
Resultaat	Geen error	Geen error	Geen error	Geen error

	Geval 5	Geval 6	Geval 7
operatorEncId	35 Char	35 Char	35 Char
timeFrame	Q3	Q4	YEAR
Resultaat	Geen error	Geen error	Geen error

SYN.T04: Incident-Customer-Satisfaction-Aantal /getCustomerSatisfactionByCount/

Stap 1: Bepaling van de attribuutrestricties

userId: Integer, Verplicht

count: Integer, minimale waarde 1, Verplicht

Stap 2: Bepaal geldige en ongeldige waarden

	Geldig	Ongeldig
userId	Integer	niet Integer
	≥ 1000	< 1000
		NULL
count	> 0	≤ 0
		NULL
		niet Integer

Stap 3: Bepaal testgevallen

	Geval 1
--	---------

userId	>= 1000
count	> 0
Resultaat	Geen error

SYN.T05: Incident-Customer-Satisfaction-Relative-Operator
/getCustomerSatisfactionRelativeToByOperator/

Stap 1: Bepaling van de attribuurestricties

operatorEncId: 35 Characters, Verplicht
timeFrame: enum TimeFrame, Verplicht

Stap 2: Bepaal geldige en ongeldige waarden

	Geldig	Ongeldig
operatorEncId	35 Char	> 35
		< 35
		“”
		NULL
timeFrame	MONTH, Q1, Q2, Q3, Q4, YEAR	NULL
		niet MONTH, Q1, Q2, Q3, Q4, YEAR
		DAY

Stap 3: Bepaal testgevallen

	Geval 1	Geval 2	Geval 3
operatorEncId	35 Char	35 Char	35 Char
timeFrame	MONTH	Q1	Q2
Resultaat	Geen error	Geen error	Geen error

	Geval 4	Geval 5	Geval 6
operatorEncId	35 Char	35 Char	35 Char
timeFrame	Q3	Q4	YEAR

Resultaat	Geen error	Geen error	Geen error
-----------	------------	------------	------------

SYN.T06: Incident-Operator-Scores-Relative-Operator [/getOperatorScoresRelativeTo/

Stap 1: Bepaling van de attributrestricties

operatorEncId: 35 Characters, Verplicht

timeFrame: enum TimeFrame, Verplicht

Stap 2: Bepaal geldige en ongeldige waarden

	Geldig	Ongeldig
operatorEncId	35 Char	> 35
		< 35
		“”
		NULL
timeFrame	MONTH, Q1, Q2, Q3, Q4, YEAR	NULL
		niet MONTH, Q1, Q2, Q3, Q4, YEAR
		DAY

Stap 3: Bepaal testgevallen

	Geval 1	Geval 2	Geval 3
operatorEncId	35 Char	35 Char	35 Char
timeFrame	MONTH	Q1	Q2
Resultaat	Geen error	Geen error	Geen error

	Geval 4	Geval 5	Geval 6
operatorEncId	35 Char	35 Char	35 Char
timeFrame	Q3	Q4	YEAR
Resultaat	Geen error	Geen error	Geen error

SYN.T07: Incident-Live-Scores-Timeframe [/getLiveScoresByTimeFrame/

Stap 1: Bepaling van de attributrestricties

userId: Integer, Verplicht

timeFrame: enum TimeFrame, Verplicht

Stap 2: Bepaal geldige en ongeldige waarden

	Geldig	Ongeldig
userId	Integer	niet Integer
	>= 1000	<1000
		NULL
timeFrame	DAY, MONTH, Q1, Q2, Q3, Q4, YEAR	NULL
		niet DAY, MONTH, Q1, Q2, Q3, Q4, YEAR

Stap 3: Bepaal testgevallen

	Geval 1	Geval 2	Geval 3	Geval 4
userId	>= 1000	>= 1000	>= 1000	>= 1000
timeFrame	DAY	MONTH	Q1	Q2
Resultaat	Geen error	Geen error	Geen error	Geen error

	Geval 5	Geval 6	Geval 7
userId	>= 1000	>= 1000	>= 1000
timeFrame	Q3	Q4	YEAR
Resultaat	Geen error	Geen error	Geen error

SYN.T08: Incident-Remarks-Timeframe /getRemarksByTimeFrame/

Stap 1: Bepaling van de attributrestricties

userId: Integer, Verplicht

timeFrame: enum TimeFrame, Verplicht

Stap 2: Bepaal geldige en ongeldige waarden

	Geldig	Ongeldig
--	--------	----------

userId	Integer	niet Integer
	>= 1000	<1000
		NULL
timeFrame	DAY, MONTH, Q1, Q2, Q3, Q4, YEAR	NULL
		niet DAY, MONTH, Q1, Q2, Q3, Q4, YEAR

Stap 3: Bepaal testgevallen

	Geval 1	Geval 2	Geval 3	Geval 4
userId	>= 1000	>= 1000	>= 1000	>= 1000
timeFrame	DAY	MONTH	Q1	Q2
Resultaat	Geen error	Geen error	Geen error	Geen error

	Geval 5	Geval 6	Geval 7
userId	>= 1000	>= 1000	>= 1000
timeFrame	Q3	Q4	YEAR
Resultaat	Geen error	Geen error	Geen error

SYN.T09: Gebruiker-Regenerate-Keys [regenerateKeys/

Stap 1: Bepaling van de attributrestricties

userId: Integer, Verplicht

user: AccessToken, Verplicht

Stap 2: Bepaal geldige en ongeldige waarden

	Geldig	Ongeldig
userId	Integer	niet Integer
	>= 1000	<1000
		NULL
userKey	32 Char	“”
		NULL

Stap 3: Bepaal testgevallen

	Geval 1	Geval 2	Geval 3
userId	>= 1000	>= 1000	>= 1000
userKey	32 Char	“”	NULL
Resultaat	Geen error	Error	Error

	Geval 4	Geval 5	Geval 6
userId	niet Integer	< 1000	NULL
userKey	32 Char	32 Char	32 Char
Resultaat	Error	Error	Error

Geval 4-5-6 zijn niet via bijvoorbeeld Postman te testen. Dit komt doordat de consumerKey valide moet zijn en de userId's uit de database komen...

SYN.T10: Gebruiker-Opvragen [/getUser/]

Stap 1: Bepaling van de attribuu Restricties

userId: Integer, Verplicht

userKey: 32 Char, Verplicht

Stap 2: Bepaal geldige en ongeldige waarden

	Geldig	Ongeldig
userId	Integer	niet Integer
	>= 1000	<1000
		NULL
userKey	32 Char	niet Char
		< 32 Char
		> 32 Char
		NULL

Stap 3: Bepaal testgevallen

	Geval 1	Geval 2	Geval 3	Geval 4
--	---------	---------	---------	---------

userId	>= 1000	Niet integer	< 1000	NULL
userKey	32 Char	32 Char	32 Char	32 Char
Resultaat	Geen error	Error	Error	Error

	Geval 5	Geval 6	Geval 7	Geval 8
user	>= 1000	>= 1000	>= 1000	>= 1000
userKey	niet Char	< 32	> 32	NULL
Resultaat	Error	Error	Error	Error

Geval 2-3-4 kunnen deze wel??

SYN.T11: Operator-Opvragen [/getOperator/]

Stap 1: Bepaling van de attribuurestricties

operatorEncId: 35 Characters, Verplicht

Stap 2: Bepaal geldige en ongeldige waarden

	Geldig	Ongeldig
operatorEncId	35 Char	> 35 Char
		< 35 Char
		“”
		NULL

Stap 3: Bepaal testgevallen

	Geval 1	Geval 2	Geval 3	Geval 4
operatorEncId	35 Char	> 35 Char	< 35 Char	“”
Resultaat	Geen error	Error	Error	Error

	Geval 5
operatorEncId	NULL
Resultaat	Error

SYN.T12: Incident-Customer-Satisfaction-Timeframe
/getCustomerSatisfactionByTimeFrame/

Stap 1: Bepaling van de attribuurestricties

userId: Integer, Verplicht

timeFrame: TimeFrame, Verplicht

Stap 2: Bepaal geldige en ongeldige waarden

	Geldig	Ongeldig
userId	Integer	niet Integer
	>= 1000	<1000
		NULL
timeFrame	DAY, MONTH, Q1, Q2, Q3, Q4, YEAR	NULL
		niet DAY, MONTH, Q1, Q2, Q3, Q4, YEAR

Stap 3: Bepaal testgevallen

	Geval 1	Geval 2	Geval 3	Geval 4
userId	>= 1000	>= 1000	>= 1000	>= 1000
timeFrame	DAY	MONTH	Q1	Q2
Resultaat	Geen error	Geen error	Geen error	Geen error

	Geval 5	Geval 6	Geval 7
userId	>= 1000	>= 1000	>= 1000
timeFrame	Q3	Q4	YEAR
Resultaat	Geen error	Geen error	Geen error

	Geval 8	Geval 9	Geval 10
userId	>= 1000	>= 1000	< 1000
timeFrame	NULL	2016	DAY

Resultaat	Geen error	Geen error	Error
-----------	------------	------------	-------

	Geval 11	Geval 12
userId	niet Integer	NULL
timeFrame	DAY	DAY
Resultaat	Error	Error

SYN.T13: Incident-Customer-Satisfaction-Relative [*getCustomerSatisfactionRelativeTo*]

Stap 1: Bepaling van de attribuutrestricties

userId: Integer, Verplicht

timeFrame: TimeFrame, Verplicht

Stap 2: Bepaal geldige en ongeldige waarden

	Geldig	Ongeldig
userId	Integer	niet Integer
	>= 1000	<1000
		NULL
timeFrame	MONTH, Q1, Q2, Q3, Q4, YEAR	NULL
		niet MONTH, Q1, Q2, Q3, Q4, YEAR
		DAY

Stap 3: Bepaal testgevallen

	Geval 1	Geval 2	Geval 3	Geval 4
userId	>= 1000	>= 1000	>= 1000	>= 1000
timeFrame	MONTH	Q1	Q2	Q3
Resultaat	Geen error	Geen error	Geen error	Geen error

	Geval 5	Geval 6	Geval 7
userId	>= 1000	>= 1000	>= 1000
timeFrame	Q4	YEAR	NULL

Resultaat	Geen error	Geen error	Geen error
-----------	------------	------------	------------

	Geval 8	Geval 9	Geval 10 ??
userId	>= 1000	>= 1000	< 1000
timeFrame	2016	DAY	MONTH
Resultaat	Geen error	Geen error	Error

	Geval 11 ??	Geval 12 ??
userId	Niet integer	NULL
timeFrame	MONTH	MONTH
Resultaat	Error	Error

SYN.T14: Incident-Operator-Scores-Timeframe [*getOperatorScoresByTimeFrame*]

Stap 1: Bepaling van de attribuurestricties

userId: Integer, Verplicht

Stap 2: Bepaal geldige en ongeldige waarden

	Geldig	Ongeldig
userId	Integer	niet Integer
	>= 1000	<1000
		NULL
timeFrame	DAY, MONTH, Q1, Q2, Q3, Q4, YEAR	NULL
		niet DAY, MONTH, Q1, Q2, Q3, Q4, YEAR

Stap 3: Bepaal testgevallen

	Geval 1	Geval 2	Geval 3	Geval 4
userId	>= 1000	>= 1000	>= 1000	>= 1000
timeFrame	DAY	MONTH	Q1	Q2

Resultaat	Geen error	Geen error	Geen error	Geen error
-----------	------------	------------	------------	------------

	Geval 5	Geval 6	Geval 7
userId	>= 1000	>= 1000	>= 1000
timeFrame	Q3	Q4	YEAR
Resultaat	Geen error	Geen error	Geen error

	Geval 8	Geval 9	Geval 10 ??
userId	>= 1000	>= 1000	< 1000
timeFrame	NULL	2016	DAY
Resultaat	Error	Error	Error

	Geval 11 ??	Geval 12 ??
userId	niet Integer	NULL
timeFrame	DAY	DAY
Resultaat	Error	Error

SYN.T15: Incident-Operator-Scores-Relative [*getOperatorScoresRelativeTo*]

Stap 1: Bepaling van de attribuurestricties

userId: Integer, Verplicht

timeFrame: TimeFrame, Verplicht

Stap 2: Bepaal geldige en ongeldige waarden

	Geldig	Ongeldig
userId	Integer	niet Integer
	>= 1000	<1000
		NULL
timeFrame	MONTH, Q1, Q2, Q3, Q4, YEAR	NULL
		niet MONTH, Q1, Q2, Q3, Q4, YEAR

		DAY
--	--	-----

Stap 3: Bepaal testgevallen

	Geval 1	Geval 2	Geval 3	Geval 4
userId	>= 1000	>= 1000	>= 1000	>= 1000
timeFrame	MONTH	Q1	Q2	Q3
Resultaat	Geen error	Geen error	Geen error	Geen error

	Geval 5	Geval 6	Geval 7
userId	>= 1000	>= 1000	>= 1000
timeFrame	Q4	YEAR	NULL
Resultaat	Geen error	Geen error	Geen error

	Geval 8	Geval 9	Geval 10 ??
userId	>= 1000	>= 1000	< 1000
timeFrame	2016	DAY	MONTH
Resultaat	Geen error	Geen error	Error

	Geval 11 ??	Geval 12 ??
userId	Niet integer	NULL
timeFrame	MONTH	MONTH
Resultaat	Error	Error

SYN.T16: Incident-Opvragen [*getFeedback*]

Stap 1: Bepaling van de attributrestricties

feedbackEncId: 35 Char, Verplicht

Stap 2: Bepaal geldige en ongeldige waarden

	Geldig	Ongeldig
--	--------	----------

feedbackEncId	35 Char	> 35 Char
		< 35 Char
		“”
		NULL

Stap 3: Bepaal testgevallen

	Geval 1	Geval 2	Geval 3	Geval 4
feedbackEncId	35 Char	> 35 Char	< 35 Char	“”
Resultaat	Geen error	Error	Error	Error

	Geval 5
feedbackEncId	NULL
Resultaat	Error

SYN.T17: Incident-Versturen [*sendFeedbackMail*]

Stap 1: Bepaling van de attributrestricties

feedbackEncId: 35 Char, Verplicht

Stap 2: Bepaal geldige en ongeldige waarden

	Geldig	Ongeldig
feedbackEncId	35 Char	> 35 Char
		< 35 Char
		“”
		NULL

Stap 3: Bepaal testgevallen

	Geval 1	Geval 2	Geval 3	Geval 4
feedbackEncId	35 Char	> 35 Char	< 35 Char	“”
Resultaat	Geen error	Error	Error	Error

	Geval 5
--	---------

feedbackEncId	NULL
Resultaat	Error

SYN.T18: Gebruiker-Toevoegen [*addUser*]

Stap 1: Bepaling van de attributrestricties

userId: Integer, Verplicht

user: AccessToken, Verplicht

Stap 2: Bepaal geldige en ongeldige waarden

	Geldig	Ongeldig
userId	Integer	niet Integer
	>= 1000	<1000
		NULL
user	32 Char	""
		NULL

Stap 3: Bepaal testgevallen

	Geval 1	Geval 2	Geval 3
userId	>= 1000	>= 1000	>= 1000
user	32 Char	""	NULL
Resultaat	Geen error	Error	Error

	Geval 4 ??	Geval 5 ??	Geval 6 ??
userId	niet Integer	< 1000	NULL
user	32 Char	32 Char	32 Char
Resultaat	Error	Error	Error

SYN.T19: Operator-Updaten [*updateOperator*]

Stap 1: Bepaling van de attributrestricties

operator: Operator, Verplicht

Stap 2: Bepaal geldige en ongeldige waarden

	Geldig	Ongeldig
operator	valide operator*	NULL
		niet valide operator

*) Valide betekent verplichte velden zijn valide en gevonden in database.

Stap 3: Bepaal testgevallen

	Geval 1	Geval 2	Geval 3
operator	valide operator	NULL	niet valide operator
Resultaat	Geen error	Error	Error

SYN.T20: Operator-Toevoegen [*addOperator*]

Stap 1: Bepaling van de attributrestricties

operator: Operator, Verplicht

Stap 2: Bepaal geldige en ongeldige waarden

	Geldig	Ongeldig
operator	valide operator*	NULL
		niet valide operator

*) Valide betekent verplichte velden zijn valide, maar niet reeds aanwezig in database.

Stap 3: Bepaal testgevallen

	Geval 1	Geval 2	Geval 3
operator	valide operator	NULL	niet valide operator
Resultaat	Geen error	Error	Error

SYN.T21: Incident-Toevoegen [*addFeedback*]

Stap 1: Bepaling van de attributrestricties

incident: Incident, Verplicht

Stap 2: Bepaal geldige en ongeldige waarden

	Geldig	Ongeldig
--	--------	----------

incident	Valide Incident	NULL
		niet valide Incident

Stap 3: Bepaal testgevallen

	Geval 1	Geval 2	Geval 3
incident	valide Incident	NULL	niet valide Incident
Resultaat	Geen error	Error	Error

SYN.T22: Incident-Updaten [*addFeedback*]

Stap 1: Bepaling van de attributrestricties

incident: Incident, Verplicht

Stap 2: Bepaal geldige en ongeldige waarden

	Geldig	Ongeldig
incident	Valide Incident	NULL
		niet valide Incident

Stap 3: Bepaal testgevallen

	Geval 1	Geval 2	Geval 3
incident	valide Incident	NULL	niet valide Incident
Resultaat	Geen error	Error	Error

Algoritme Testen: statement coverage

AT.T01: Gebruiker-Regenerate-Keys

1 3 4 5 6 8

2 10

1 3 4 5 7 9

4) Definieer testgevallen

	Geval 1	Geval 2	Geval 3
Scenariobeschrijving	Succesvol genereren van nieuwe consumerKey en -	Validate consumerKey geeft een foutmelding, omdat deze niet valide	Update geeft een foutmelding, omdat de update in de database

	Secret.	is.	verkeerd gaat.
Pad	1-3-4-5-6-8	2-10	1-3-4-5-7-9

AT.T02: Gebruiker-Opvragen

1 3 4
2 5

4) Definieer testgevallen

	Geval 1	Geval 2
Scenariobeschrijving	Succesvol verkrijgen van gebruiker.	Validate consumerKey geeft een foutmelding, omdat deze niet valide is.
Pad	1-3-4	2-5

AT.T03: Operator-Opvragen

1 3 4
2 5

4) Definieer testgevallen

	Geval 1	Geval 2
Scenariobeschrijving	Succesvol verkrijgen van een Operator	Validate operatorEncId geeft een foutmelding, omdat deze niet valide is.
Pad	1-3-4	2-5

AT.T04: Incident-Customer-Satisfaction-Timeframe

1 4 5 6
2 3 4 5 6

4) Definieer testgevallen

	Geval 1	Geval 2
Scenariobeschrijving	Succesvol verkrijgen van Customer Satisfaction per TimeFrame	De waarde van het opgegeven TimeFrame wordt omgezet naar DAY en hiervan wordt de Customer Satisfaction opgehaald.
Pad	1-4-5-6	2-3-4-5-6

AT.T05: Incident-Customer-Satisfaction-Relative

1 3 6 8 9 11 14 17
 2 4 3 5 7 8 9 10 13 16
 1 3 5 7 8 9 10 13 16
 1 3 6 8 9 12 15 18

4) Definieer testgevallen

	Geval 1	Geval 2	Geval 3	Geval 4
Scenariobeschrijving	Succesvol ophalen van Customer Satisfaction. Per opgegeven kwartaal en hetzelfde kwartaal van het voorgaande jaar.	De waarde van het opgegeven TimeFrame wordt omgezet naar MONTH. Vervolgens wordt succesvol de Customer Satisfaction opgehaald. Per huidige maand en dezelfde maand van het voorgaande jaar.	De waarde van het opgegeven TimeFrame wordt omgezet naar MONTH. Vervolgens wordt succesvol de Customer Satisfaction opgehaald. Per huidige maand en dezelfde maand van het voorgaande jaar.	Succesvol ophalen van Customer Satisfaction. Per huidig jaar en het voorgaande jaar.
Pad	1-3-6-8-9-11-14-17	2-4-3-5-7-8-9-10-13 -16	1-3-5-7-8-9-10-13-16	1-3-6-8-9-12-15-18

AT.T06: Incident-Operator-Scores-Timeframe

1 4 5 6 7
 2 3 4 5 6 7

4) Definieer testgevallen

	Geval 1	Geval 2
Scenariobeschrijving	Succesvol ophalen van Operator Scores voor het opgegeven TimeFrame.	Het opgegeven TimeFrame wordt omgezet naar DAY. Daarna wordt de Operator Score succesvol opgehaald.
Pad	1-4-5-6-7	2-3-4-5-6-7

AT.T07: Incident-Operator-Scores-Relative

1-4-6-8-9-11-14-16
 2-3-4-5-7-8-9-10-13-17

1-4-5-7-8-9-10-13-17
1-4-6-8-9-12-15-18

4) Definieer testgevallen

	Geval 1	Geval 2	Geval 3	Geval 4
Scenariobeschrijving	Succesvol ophalen van Operator Scores. Per huidig kwartaal en hetzelfde kwartaal van het voorgaande jaar.	De waarde van het opgegeven TimeFrame wordt omgezet naar MONTH. Vervolgens wordt succesvol de Operator Scores opgehaald. Per huidige maand en dezelfde maand van het voorgaande jaar.	De waarde van het opgegeven TimeFrame wordt omgezet naar MONTH. Vervolgens wordt succesvol de Operator Scores opgehaald. Per huidige maand en dezelfde maand van het voorgaande jaar.	Succesvol ophalen van Operator Scores. Per huidig jaar en het voorgaande jaar.
Pad	1-4-6-8-9-11-14-16	2-3-4-5-7-8-9-10-13-17	1-4-5-7-8-9-10-13-17	1-4-6-8-9-12-15-18

Algoritme Testen: branche coverage

AT.T08: Incident-Opvragen

1-3-5
2-4

4) Definieer testgevallen

	Geval 1	Geval 2
Scenariobeschrijving	Succesvol ophalen van Feedback voor het opgegeven feedbackEncId	ValidateFeedbackEncId geeft een error, omdat het opgegeven feedbackEncId niet valide is.
Pad	1-3-5	2-4

AT.T09: Incident-Versturen

1-3-6
2-4
1-5-7

4) Definieer testgevallen

	Geval 1	Geval 2	Geval 3
Scenariobeschrijving	het succesvol verzenden van een feedback mail voor het opgegeven feedbackEncId.	ValidateFeedbackEncId geeft een error, omdat het opgegeven feedbackEncId niet valide is.	SendFeedbackMail geeft FALSE, omdat de feedback mail niet verzonden kon worden.
Pad	1-3-6	2-4	1-5-7

AT.T10: Gebruiker-Toevoegen

1-3-5-7-10

2-4

1-3-6-8

1-3-5-9-11

4) Definieer testgevallen

	Geval 1	Geval 2	Geval 3	Geval 4
Scenariobeschrijving	Het succesvol toevoegen van een User.	ValidateUser geeft een foutmelding, omdat de User niet valide is.	Er wordt een foutmelding gegeven, omdat de gebruiker reeds bestaat.	CreateUser geeft een exception, omdat het toevoegen niet lukt.
Pad	1-3-5-7-10	2-4	1-3-6-8	1-3-5-9-11

AT.T11: Operator-Update

1-2-4-5-9-10-11-12

1-2-3-7

1-2-4-5-6-8

4) Definieer testgevallen

	Geval 1	Geval 2	Geval 3
Scenariobeschrijving	Succesvol updaten van een Operator.	ValidateOperator geeft een foutmelding gegeven, omdat de Operator niet valide is.	Er wordt een foutmelding gegeven omdat de Operator niet bestaat.
Pad	1-2-4-5-9-10-11-12	1-2-3-7	1-2-4-5-6-8

AT.T12: Operator-Toevoegen

1-2-4-6-7-9-13-16-17

1-2-3-5

1-2-4-6-7-10-17

1-2-4-6-8-12-15

1-2-4-6-8-11-14

4) Definieer testgevallen

	Geval 1	Geval 2	Geval 3
Scenariobeschrijving	Het toevoegen van een bestaande Operator welke niet actief is, wordt weer actief gemaakt.	ValidateOperator geeft een foutmelding, omdat de Operator niet valide is.	Het toevoegen van een bestaande Operator welke ook actief is.
Pad	1-2-4-6-7-9-13-16-17	1-2-3-5	1-2-4-6-7-10-17

	Geval 4	Geval 5
Scenariobeschrijving	Het succesvol toevoegen van een nieuwe Operator.	CreateOperator geeft een exception, omdat het toevoegen niet is gelukt.
Pad	1-2-4-6-8-12-15	1-2-4-6-8-11-14

AT.T13: Incident-Toevoegen = Incident-Updaten

1-2-4-7-9-10

1-2-3-5

1-2-4-6-8

4) Definieer testgevallen

	Geval 1	Geval 2	Geval 3
Scenariobeschrijving	Het succesvol toevoegen van een nieuw incident.	ValidateIncident geeft een foutmelding, omdat het incident niet valide is.	ProcesIncident geeft een exception, omdat het verwerken van het incident niet is gelukt.
Pad	1-2-4-7-9-10	1-2-3-5	1-2-4-6-8

Cause Effect graphing

```
@Override
public IncidentResponse addFeedback(Incident incident) {

    IncidentResponse incidentResponse = new IncidentResponse();

    logger.info("Post /incident/ triggered for userId: " + incident.getUserId() + " @: " + new Date()
        + " with id: " + incident.getId() + ".");
    if( validateIncident(incident) ) {
        try {
            feedbackManager.processIncident(incident, false);
            Feedback feedback = feedbackManager.getFeedbackByIncident(incident);
            incidentResponse = getIncidentResponse(feedback, true);
        } catch(Exception e) {
            logger.error("Could not process post /incident", e);
            incidentResponse.setMessage("Could not process incident!");
        }
    } else {
        logger.error("request is not valid, there is no feedback created.");
        incidentResponse.setMessage("Request is not valid!");
    }

    return incidentResponse;
}
```

1. bepaal condities en acties

C1: valide incident

C2: throw exception

A1: processIncident

A2: setMessage => "Could not process incident!"

A3: setMessage => "Request is not valid"

2) Definieer beslissingstabel

Conditioes/ Acties	1	2	3	4
C1: Incident is valide	0	1	0	1
C2: Throw exception	0	0	1	1

A1: Verwerk incident		X		
A2: "Could not process incident"				X
A3: "Request is not valid"	X		X	

3) Reduceer het aantal testgevallen

Testgeval 3 dient verwijderd te worden. De reden is dat er geen mogelijkheid is tot het genereren van een exception wanneer het incident niet valide is. Deze exception kan voor komen wanneer het niet lukt om het incident te verwerken hiervoor dient deze wel valide te zijn. Ook is de uitkomst van dit geval gelijk aan testgeval 1.

4) Bepaal de testgevallen

	Geval 1	Geval 2	Geval 3	Geval 4
--	---------	---------	---------	---------

Scenariobeschrijving	Het request is niet valide, omdat het incident niet valide is.	Het incident wordt verwerkt tot een feedback in de database.	Is niet mogelijk	Er treed een exception op doordat het incident niet verwerkt kan worden.
----------------------	--	--	------------------	--

Fysieke testgevallen

Operator-Verwijderen

Id	SYN.T01				
Doel van de test	Verifieer dat de status van een operator op FALSE staat als deze is verwijderd.				
Preconditie	Operator bestaat en heeft een status TRUE.				
Postconditie	Operator heeft een status FALSE.				
Verwijzing testboom	Operator-verwijderen				
Testactie(s)	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	deleteOperator wordt aangeroepen met operatorEncId "345-0-09df-3209-09df-3209-09df-3209"	De status van de Operator in de response is FALSE		

Operator-Get-List

Id	SYN.T02				
Doel van de test	Verifieer dat er een lijst met de operators (actief en inactief) van de gebruiker wordt teruggegeven.				
Preconditie	De gebruiker is bekend en heeft een aantal operators.				
Postconditie	Er komt een lijst met operators terug als response.				
Verwijzing testboom	Operator-Get-List				

Testactie(s)					
	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	getOperatorList wordt aangeroepen met userId 1100	Een lijst met operators met de volgende encId's: e54beacc-09be-4bc8-bd41-a7b255bbc22b, 61987163-ca2f-4897-8cb3-e786701b41f9, 78222c7a-0c7d-47f8-8f1c-d3e3cfb8dd11		

Operator-Operator-Score

Id	SYN.T03-1				
Doel van de test	Verifieer dat de juiste operator score wordt teruggegeven voor de opgegeven dag en de dag ervoor.				
Preconditie	Operator is aanwezig en heeft op de dag een operator score. Timeframe is DAY				
Postconditie	Als response komen er twee operator scores terug voor de juiste operator.				
Verwijzing testboom	Operator-Operator-Score				
Testactie(s)					
	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call "operator/stats/operator-score/{operatorEncId}" wordt uitgevoerd met operatorEncId "DAO2"			

Id	SYN.T03-2				
Doel van de test	Verifieer dat de juiste operator score wordt teruggegeven voor de opgegeven maand en de maand ervoor.				

Preconditie	Operator is aanwezig en heeft in de maand een operator score. Timeframe is MONTH				
Postconditie	Als response komen er twee operator scores terug voor de juiste operator.				
Verwijzing testboom	Operator-Operator-Score				
Testactie(s)	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call “operator/stats/operator-score/{operatorEncId}” wordt uitgevoerd met operatorEncId “78222c7a-0c7d-47f8-8f1c-d3e3cfb8dd11”			

Id	SYN.T03-3				
Doel van de test	Verifieer dat de juiste operator score wordt teruggegeven voor het opgegeven kwartaal en het kwartaal ervoor.				
Preconditie	Operator is aanwezig en heeft in de maand een operator score. Timeframe is Q1				
Postconditie	Als response komen er twee operator scores terug voor de juiste operator.				
Verwijzing testboom	Operator-Operator-Score				
Testactie(s)	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call “operator/stats/operator-score/{operatorEncId}” wordt uitgevoerd met operatorEncId “78222c7a-0c7d-47f8-8f1c-d3e3cfb8dd11”			

Id	SYN.T03-4				
Doel van de test	Verifieer dat de juiste operator score wordt teruggeven voor het opgegeven kwartaal en het kwartaal ervoor.				
Preconditie	Operator is aanwezig en heeft in de maand een operator score. Timeframe is Q2				
Postconditie	Als response komen er twee operator scores terug voor de juiste operator.				
Verwijzing testboom	Operator-Operator-Score				
Testactie(s)	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call “operator/stats/operator-score/{operatorEncId}” wordt uitgevoerd met operatorEncId “78222c7a-0c7d-47f8-8f1c-d3e3cfb8dd11”			

Id	SYN.T03-5				
Doel van de test	Verifieer dat de juiste operator score wordt teruggeven voor het opgegeven kwartaal en het kwartaal ervoor.				
Preconditie	Operator is aanwezig en heeft in de maand een operator score. Timeframe is Q3				
Postconditie	Als response komen er twee operator scores terug voor de juiste operator.				
Verwijzing testboom	Operator-Operator-Score				
Testactie(s)	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call “operator/stats/operator-score/{operatorEncId}” wordt uitgevoerd met operatorEncId “78222c7a-			

		0c7d-47f8-8f1c-d3e3cfb8dd11 ”			
--	--	----------------------------------	--	--	--

Id	SYN.T03-6				
Doel van de test	Verifieer dat de juiste operator score wordt teruggegeven voor het opgegeven kwartaal en het kwartaal ervoor.				
Preconditie	Operator is aanwezig en heeft in de maand een operator score. Timeframe is Q4				
Postconditie	Als response komen er twee operator scores terug voor de juiste operator.				
Verwijzing testboom	Operator-Operator-Score				
Testactie(s)	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call “operator/stats/operator-score/{operatorEncId}” wordt uitgevoerd met operatorEncId “78222c7a-0c7d-47f8-8f1c-d3e3cfb8dd11”			

Id	SYN.T03-7				
Doel van de test	Verifieer dat de juiste operator score wordt teruggegeven voor het opgegeven jaar en het jaar ervoor.				
Preconditie	Operator is aanwezig en heeft in de maand een operator score. Timeframe is YEAR				
Postconditie	Als response komen er twee operator scores terug voor de juiste operator.				
Verwijzing testboom	Operator-Operator-Score				
Testactie(s)	Testactie	Testactie	Verwachte	Conclusie	Opmerking

	Id		reactie		
	1	De call “operator/stats/operator-score/{operatorEncId} ” wordt uitgevoerd met operatorEncId “78222c7a-0c7d-47f8-8f1c-d3e3cfb8dd11 ”			

Incident-Customer-Satisfaction-Aantal

Id	SYN.T04				
Doel van de test	Verifieer dat het systeem de customer satisfaction terug geeft van de laatste 20 beoordeelde feedbacks en de 20 daarvoor.				
Preconditie	Voor het valide userId zijn 40 feedbacks beoordeeld.				
Postconditie	Als response wordt de customer satisfaction terug gegeven van de laatste 20 beoordeelde feedbacks en de 20 daarvoor.				
Verwijzing testboom	Incident-Customer-Satisfaction-Aantal				
Testactie(s)	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call “incident/stats/customer-satisfaction/count“ wordt uitgevoerd met een count van 20 met het userId 1000.			

Incident-Customer-Satisfaction-Relative-Operator

Id	SYN.T05-1				
Doel van de test	Controleer of er een customer satisfaction als response komt voor een valide operator voor de opgegeven maand en dezelfde maand het jaar ervoor.				
Preconditie	Er zijn voor de valide user feedbacks aanwezig voor het timeframe.				
Postconditie	De customer satisfaction voor de valide operator worden teruggegeven als response voor het huidige jaar en het jaar ervoor.				

Verwijzing testboom	Incident-Customer-Satisfaction-Relative-Operator				
Testactie(s)					
	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call “incident/stats/customer-satisfaction/{operatorEncId}/relative-to” wordt uitgevoerd met timeframe MONTH en als operatorEncId “61987163-ca2f-4897-8cb3-e786701b41f9”			

Id	SYN.T05-2				
Doel van de test	Controleer of er een customer satisfaction als response komt voor een valide operator voor het opgegeven kwartaal en hetzelfde kwartaal het jaar ervoor.				
Preconditie	Er zijn voor de valide user feedbacks aanwezig voor het timeframe.				
Postconditie	De customer satisfaction voor de valide operator worden teruggegeven als response voor het huidige jaar en het jaar ervoor.				
Verwijzing testboom	Incident-Customer-Satisfaction-Relative-Operator				
Testactie(s)					
	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call “incident/stats/customer-satisfaction/{operatorEncId}/relative-to” wordt uitgevoerd met timeframe Q1 en als operatorEncId “61987163-ca2f-4897-8cb3-e786701b41f9”			

Id	SYN.T05-3				
Doel van de test	Controleer of er een customer satisfaction als response komt voor een valide operator voor het opgegeven kwartaal en hetzelfde kwartaal het jaar ervoor.				

Preconditie	Er zijn voor de valide user feedbacks aanwezig voor het timeframe.				
Postconditie	De customer satisfaction voor de valide operator worden teruggegeven als response voor het huidige jaar en het jaar ervoor.				
Verwijzing testboom	Incident-Customer-Satisfaction-Relative-Operator				
Testactie(s)					
	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call “incident/stats/customer-satisfaction/{operatorEncId}/relative-to” wordt uitgevoerd met timeframe Q2 en als operatorEncId “61987163-ca2f-4897-8cb3-e786701b41f9”			

Id	SYN.T05-4				
Doel van de test	Controleer of er een customer satisfaction als response komt voor een valide operator voor het opgegeven kwartaal en hetzelfde kwartaal het jaar ervoor.				
Preconditie	Er zijn voor de valide user feedbacks aanwezig voor het timeframe.				
Postconditie	De customer satisfaction voor de valide operator worden teruggegeven als response voor het huidige jaar en het jaar ervoor.				
Verwijzing testboom	Incident-Customer-Satisfaction-Relative-Operator				
Testactie(s)					
	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call “incident/stats/customer-satisfaction/{operatorEncId}/relative-to” wordt uitgevoerd met timeframe Q3 en als operatorEncId “61987163-ca2f-4897-8cb3-e786701b41f9”			

Id	SYN.T05-5				
Doel van de test	Controleer of er een customer satisfaction als response komt voor een valide operator voor het opgegeven kwartaal en hetzelfde kwartaal het jaar ervoor.				
Preconditie	Er zijn voor de valide user feedbacks aanwezig voor het timeframe.				
Postconditie	De customer satisfaction voor de valide operator worden teruggegeven als response voor het huidige jaar en het jaar ervoor.				
Verwijzing testboom	Incident-Customer-Satisfaction-Relative-Operator				
Testactie(s)					
	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call “incident/stats/customer-satisfaction/{operatorEncId}/relative-to“ wordt uitgevoerd met timeframe Q4 en als operatorEncId “61987163-ca2f-4897-8cb3-e786701b41f9”			

Id	SYN.T05-6				
Doel van de test	Controleer of er een customer satisfaction als response komt voor een valide operator voor het opgegeven jaar en het jaar ervoor.				
Preconditie	Er zijn voor de valide user feedbacks aanwezig voor het timeframe.				
Postconditie	De customer satisfaction voor de valide operator worden teruggegeven als response voor het huidige jaar en het jaar ervoor.				
Verwijzing testboom	Incident-Customer-Satisfaction-Relative-Operator				
Testactie(s)					
	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call “incident/stats/customer-satisfaction/{operatorEncId}/relative-to“ wordt uitgevoerd met			

		timeframe YEAR en als operatorEncId “61987163-ca2f-4897-8cb3-e786701b41f9”			
--	--	--	--	--	--

Incident-Operator-Scores-Relative-Operator

Id	SYN.T06-1				
Doel van de test	Controleer of er een operator score als response komt voor een valide operator voor de opgegeven maand en dezelfde maand het jaar ervoor.				
Preconditie	Er zijn voor de valide user feedbacks aanwezig voor het timeframe.				
Postconditie	De operator score voor de valide operator worden teruggegeven als response voor het huidige jaar en het jaar ervoor.				
Verwijzing testboom	Incident-Operator-Scores-Relative-Operator				
Testactie(s)	Testacti e Id	Testactie	Verwacht e reactie	Conclusi e	Opmerkin g
	1	De call “incident/stats/operator-scores/{operatorEncId}/relative-to” wordt uitgevoerd met timeframe YEAR en als operatorEncId “61987163-ca2f-4897-8cb3-e786701b41f9”			

Id	SYN.T06-2				
Doel van de test	Controleer of er een operator score als response komt voor een valide operator voor het opgegeven kwartaal en hetzelfde kwartaal het jaar ervoor.				
Preconditie	Er zijn voor de valide user feedbacks aanwezig voor het timeframe.				
Postconditie	De operator score voor de valide operator worden teruggegeven als response voor het huidige jaar en het jaar ervoor.				
Verwijzing testboom	Incident-Operator-Scores-Relative-Operator				
Testactie(s)					

	Testacti e Id	Testactie	Verwacht e reactie	Conclusi e	Opmerkin g
	1	De call “incident/stats/operator- scores/{operatorEncId}/relativ e-to“ wordt uitgevoerd met timeframe Q1 en als operatorEncId “61987163- ca2f-4897-8cb3- e786701b41f9”			

Id	SYN.T06-3				
Doel van de test	Controleer of er een operator score als response komt voor een valide operator voor het opgegeven kwartaal en hetzelfde kwartaal het jaar ervoor.				
Preconditie	Er zijn voor de valide user feedbacks aanwezig voor het timeframe.				
Postconditi e	De operator score voor de valide operator worden teruggegeven als response voor het huidige jaar en het jaar ervoor.				
Verwijzing testboom	Incident-Operator-Scores-Relative-Operator				
Testactie(s)	Testacti e Id	Testactie	Verwacht e reactie	Conclusi e	Opmerkin g
	1	De call “incident/stats/operator- scores/{operatorEncId}/relativ e-to“ wordt uitgevoerd met timeframe Q2 en als operatorEncId “61987163- ca2f-4897-8cb3- e786701b41f9”			

Id	SYN.T06-4				
Doel van de test	Controleer of er een operator score als response komt voor een valide operator voor het opgegeven kwartaal en hetzelfde kwartaal het jaar ervoor.				
Preconditie	Er zijn voor de valide user feedbacks aanwezig voor het timeframe.				
Postconditi	De operator score voor de valide operator worden teruggegeven als response				

e	voor het huidige jaar en het jaar ervoor.				
Verwijzing testboom	Incident-Operator-Scores-Relative-Operator				
Testactie(s)	Testacti e Id	Testactie	Verwacht e reactie	Conclusi e	Opmerkin g
	1	De call “incident/stats/operator- scores/{operatorEncId}/relativ e-to“ wordt uitgevoerd met timeframe Q3 en als operatorEncId “61987163- ca2f-4897-8cb3- e786701b41f9”			

Id	SYN.T06-5				
Doel van de test	Controleer of er een operator score als response komt voor een valide operator voor het opgegeven kwartaal en hetzelfde kwartaal het jaar ervoor.				
Preconditie	Er zijn voor de valide user feedbacks aanwezig voor het timeframe.				
Postconditie	De operator score voor de valide operator worden teruggegeven als response voor het huidige jaar en het jaar ervoor.				
Verwijzing testboom	Incident-Operator-Scores-Relative-Operator				
Testactie(s)	Testacti e Id	Testactie	Verwacht e reactie	Conclusi e	Opmerkin g
	1	De call “incident/stats/operator- scores/{operatorEncId}/relativ e-to“ wordt uitgevoerd met timeframe Q4 en als operatorEncId “61987163- ca2f-4897-8cb3- e786701b41f9”			

Id	SYN.T06-6				
----	-----------	--	--	--	--

Doel van de test	Controleer of er een operator score als response komt voor een valide operator voor het opgegeven jaar en het jaar ervoor.				
Preconditie	Er zijn voor de valide user feedbacks aanwezig voor het timeframe.				
Postconditie	De operator score voor de valide operator worden teruggegeven als response voor het huidige jaar en het jaar ervoor.				
Verwijzing testboom	Incident-Operator-Scores-Relative-Operator				
Testactie(s)					
	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call “incident/stats/operator-scores/{operatorEncId}/relative-to” wordt uitgevoerd met timeframe YEAR en als operatorEncId “61987163-ca2f-4897-8cb3-e786701b41f9”			

Incident-Live-Scores-Timeframe

Id	SYN.T07-1				
Doel van de test	Controller of er een live scores als response komt voor het opgegeven valide timeframe.				
Preconditie	Er zijn voor het valide timeframe beoordeelde feedbacks aanwezig.				
Postconditie	De live scores voor het valide timeframe worden teruggegeven als response voor de huidige periode en de periode ervoor.				
Verwijzing testboom	Incident-Live-Scores-Timeframe				
Testactie(s)					
	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call “incident/stats/live-scores” wordt uitgevoerd voor het timeframe DAY en het userId 1000			

Id	SYN.T07-2				
Doel van de test	Controller of er een live scores als response komt voor het opgegeven valide timeframe.				
Preconditie	Er zijn voor het valide timeframe beoordeelde feedbacks aanwezig.				
Postconditie	De live scores voor het valide timeframe worden teruggegeven als response voor de huidige periode en de periode ervoor.				
Verwijzing testboom	Incident-Live-Scores-Timeframe				
Testactie(s)	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call “incident/stats/live-scores“ wordt uitgevoerd voor het timeframe MONTH en het userId 1000			

Id	SYN.T07-3				
Doel van de test	Controller of er een live scores als response komt voor het opgegeven valide timeframe.				
Preconditie	Er zijn voor het valide timeframe beoordeelde feedbacks aanwezig.				
Postconditie	De live scores voor het valide timeframe worden teruggegeven als response voor de huidige periode en de periode ervoor.				
Verwijzing testboom	Incident-Live-Scores-Timeframe				
Testactie(s)	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call “incident/stats/live-scores“ wordt uitgevoerd voor het timeframe Q1 en het userId 1000			

Id	SYN.T07-4				
Doel van de test	Controller of er een live scores als response komt voor het opgegeven valide timeframe.				
Preconditie	Er zijn voor het valide timeframe beoordeelde feedbacks aanwezig.				
Postconditie	De live scores voor het valide timeframe worden teruggegeven als response voor de huidige periode en de periode ervoor.				
Verwijzing testboom	Incident-Live-Scores-Timeframe				
Testactie(s)	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call “incident/stats/live-scores“ wordt uitgevoerd voor het timeframe Q2 en het userId 1000			

Id	SYN.T07-5				
Doel van de test	Controller of er een live scores als response komt voor het opgegeven valide timeframe.				
Preconditie	Er zijn voor het valide timeframe beoordeelde feedbacks aanwezig.				
Postconditie	De live scores voor het valide timeframe worden teruggegeven als response voor de huidige periode en de periode ervoor.				
Verwijzing testboom	Incident-Live-Scores-Timeframe				
Testactie(s)	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call “incident/stats/live-scores“ wordt uitgevoerd voor het timeframe Q3 en het userId 1000			

Id	SYN.T07-6				
Doel van de test	Controller of er een live scores als response komt voor het opgegeven valide timeframe.				
Preconditie	Er zijn voor het valide timeframe beoordeelde feedbacks aanwezig.				
Postconditie	De live scores voor het valide timeframe worden teruggegeven als response voor de huidige periode en de periode ervoor.				
Verwijzing testboom	Incident-Live-Scores-Timeframe				
Testactie(s)	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call “incident/stats/live-scores“ wordt uitgevoerd voor het timeframe Q4 en het userId 1000			

Id	SYN.T07-7				
Doel van de test	Controller of er een live scores als response komt voor het opgegeven valide timeframe.				
Preconditie	Er zijn voor het valide timeframe beoordeelde feedbacks aanwezig.				
Postconditie	De live scores voor het valide timeframe worden teruggegeven als response voor de huidige periode en de periode ervoor.				
Verwijzing testboom	Incident-Live-Scores-Timeframe				
Testactie(s)	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call “incident/stats/live-scores“ wordt uitgevoerd voor het timeframe YEAR en het userId 1000			

Incident-Remarks-Timeframe

Id	SYN.T08-1				
Doel van de test	Controller of er remarks als response komen voor het opgegeven valide timeframe.				
Preconditie	Er zijn voor het valide timeframe remarks aanwezig.				
Postconditie	De remarks voor het valide timeframe worden teruggegeven als response voor de huidige periode.				
Verwijzing testboom	Incident-Remarks-Timeframe				
Testactie(s)	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call “incident/stats/remarks “ wordt uitgevoerd met het timeframe DAY voor het userId 1000.			

Id	SYN.T08-2				
Doel van de test	Controller of er remarks als response komen voor het opgegeven valide timeframe.				
Preconditie	Er zijn voor het valide timeframe remarks aanwezig.				
Postconditie	De remarks voor het valide timeframe worden teruggegeven als response voor de huidige periode.				
Verwijzing testboom	Incident-Remarks-Timeframe				
Testactie(s)	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call “incident/stats/remarks “ wordt uitgevoerd met het timeframe MONTH voor het userId 1000.			

Id	SYN.T08-3				
Doel van de test	Controller of er remarks als response komen voor het opgegeven valide timeframe.				
Preconditie	Er zijn voor het valide timeframe remarks aanwezig.				
Postconditie	De remarks voor het valide timeframe worden teruggegeven als response voor de huidige periode.				
Verwijzing testboom	Incident-Remarks-Timeframe				
Testactie(s)	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call “incident/stats/remarks “ wordt uitgevoerd met het timeframe Q1 voor het userId 1000.			

Id	SYN.T08-4				
Doel van de test	Controller of er remarks als response komen voor het opgegeven valide timeframe.				
Preconditie	Er zijn voor het valide timeframe remarks aanwezig.				
Postconditie	De remarks voor het valide timeframe worden teruggegeven als response voor de huidige periode.				
Verwijzing testboom	Incident-Remarks-Timeframe				
Testactie(s)	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call “incident/stats/remarks “ wordt uitgevoerd met het timeframe Q2 voor het userId 1000.			

Id	SYN.T08-5				
Doel van de test	Controller of er remarks als response komen voor het opgegeven valide timeframe.				
Preconditie	Er zijn voor het valide timeframe remarks aanwezig.				
Postconditie	De remarks voor het valide timeframe worden teruggegeven als response voor de huidige periode.				
Verwijzing testboom	Incident-Remarks-Timeframe				
Testactie(s)	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call “incident/stats/remarks “ wordt uitgevoerd met het timeframe Q3 voor het userId 1000.			

Id	SYN.T08-6				
Doel van de test	Controller of er remarks als response komen voor het opgegeven valide timeframe.				
Preconditie	Er zijn voor het valide timeframe remarks aanwezig.				
Postconditie	De remarks voor het valide timeframe worden teruggegeven als response voor de huidige periode.				
Verwijzing testboom	Incident-Remarks-Timeframe				
Testactie(s)	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call “incident/stats/remarks “ wordt uitgevoerd met het timeframe Q4 voor het userId 1000.			

Id	SYN.T08-7				
Doel van de test	Controller of er remarks als response komen voor het opgegeven valide timeframe.				
Preconditie	Er zijn voor het valide timeframe remarks aanwezig.				
Postconditie	De remarks voor het valide timeframe worden teruggegeven als response voor de huidige periode.				
Verwijzing testboom	Incident-Remarks-Timeframe				
Testactie(s)	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call “incident/stats/remarks” wordt uitgevoerd met het timeframe YEAR voor het userId 1000.			

=====

=====

Gebruiker-Regenerate-Keys

Id	SYN.T09-1				
Doel van de test	Controleer of de consumerkey en -secret anders zijn dan voordat de test werd uitgevoerd.				
Preconditie	Een valide gebruiker is aanwezig in het systeem.				
Postconditie	De valide gebruiker heeft een andere consumerkey en -secret.				
Verwijzing testboom	Gebruiker-Regenerate-Keys				
Testactie(s)	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call “accesstoken/regenerate-keys” wordt uitgevoerd met consumerkey “XXXXXXXXX” en userId 1000			

Id	SYN.T09-2				
Doel van de test	Verifieer dat er geen consumerkey en -secret worden aangepast wanneer er geen consumerkey wordt meegegeven in de aanvraag				
Preconditie	Een valide gebruiker is aanwezig in het systeem.				
Postconditie	De valide gebruiker heeft dezelfde consumerkey en -secret.				
Verwijzing testboom	Gebruiker-Regenerate-Keys				
Testactie(s)					
	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call “accesstoken/regenerate-keys” wordt uitgevoerd met een lege consumerkey en userId 1000	Het response object bevat een message met volgende tekst: "No valid userkey supplied!"		

Id	SYN.T09-3				
Doel van de test	Verifieer dat er geen consumerkey en -secret worden aangepast wanneer er geen valide consumerkey wordt meegegeven in de aanvraag				
Preconditie	Een valide gebruiker is aanwezig in het systeem.				
Postconditie	De valide gebruiker heeft dezelfde consumerkey en -secret.				
Verwijzing testboom	Gebruiker-Regenerate-Keys				
Testactie(s)					
	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call “accesstoken/regenerate-keys” wordt uitgevoerd	Het response object		

		met NULL als consumerkey en userId 1000	bevat een message met volgende tekst: "No valid userkey supplied!"		
--	--	---	--	--	--

Id	SYN.T09-4				
Doel van de test	Verifieer dat er geen consumerkey en -secret worden aangepast wanneer er geen valide userId wordt meegegeven in de aanvraag				
Preconditie	Een valide gebruiker is aanwezig in het systeem.				
Postconditie	De valide gebruiker heeft dezelfde consumerkey en -secret.				
Verwijzing testboom	Gebruiker-Regenerate-Keys				
Testactie(s)	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call "accesstoken/regenerate-keys" wordt uitgevoerd met de consumerkey "XXXXXX" en userId ABCDF	???? Kan dit eigenlijk wel??		

Id	SYN.T09-5				
Doel van de test	Verifieer dat er geen consumerkey en -secret worden aangepast wanneer er geen valide userId wordt meegegeven in de aanvraag				
Preconditie	Een valide gebruiker is aanwezig in het systeem.				
Postconditie	De valide gebruiker heeft dezelfde consumerkey en -secret.				
Verwijzing testboom	Gebruiker-Regenerate-Keys				
Testactie(s)	Testactie	Testactie	Verwachte	Conclusie	Opmerking

	Id		reactie		
	1	De call “accesstoken/regenerate- keys” wordt uitgevoerd met een lege consumerkey en userId 1000	Kan dit wel ??		

Id	SYN.T09-6				
Doel van de test	Verifieer dat er geen consumerkey en -secret worden aangepast wanneer er geen valide userId wordt meegegeven in de aanvraag				
Preconditie	Een valide gebruiker is aanwezig in het systeem.				
Postconditie	De valide gebruiker heeft dezelfde consumerkey en -secret.				
Verwijzing testboom	Gebruiker-Regenerate-Keys				
Testactie(s)	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call “accesstoken/regenerate- keys” wordt uitgevoerd met een lege consumerkey en userId 1000	kan dit wel??		

Gebruiker-Opvragen

Id	SYN.T10-1				
Doel van de test	Verifieer dat de gegevens van een valide gebruiker kunnen worden opgehaald met een valide consumerkey.				
Preconditie	De gebruiker is aanwezig in het systeem.				
Postconditie	De gegevens van de gebruiker worden teruggegeven in de response.				
Verwijzing testboom	Gebruiker-Opvragen				
Testactie(s)	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking

	1	De call GET “accesstoken/” wordt uitgevoerd met de consumerkey “XXXXXX” en userId 1000	De gegevens worden teruggegeven in de response		
--	---	--	--	--	--

Id	SYN.T10-2 KAN DEZE WEL?				
Doel van de test	Verifieer dat de gegevens van een valide gebruiker kunnen worden opgehaald met een valide consumerkey.				
Preconditie	De gebruiker is aanwezig in het systeem.				
Postconditie	De gegevens van de gebruiker worden teruggegeven in de response.				
Verwijzing testboom	Gebruiker-Opvragen				
Testactie(s)	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call GET “accesstoken/” wordt uitgevoerd met de consumerkey “XXXXXX” en userId 1000	De gegevens worden teruggegeven in de response		

Id	SYN.T10-3 KAN DEZE WEL??				
Doel van de test	Verifieer dat de gegevens van een valide gebruiker kunnen worden opgehaald met een valide consumerkey.				
Preconditie	De gebruiker is aanwezig in het systeem.				
Postconditie	De gegevens van de gebruiker worden teruggegeven in de response.				
Verwijzing testboom	Gebruiker-Opvragen				
Testactie(s)	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call GET	De gegevens		

		“accesstoken/” wordt uitgevoerd met de consumerkey “XXXXXX” en userId 1000	worden teruggegeven in de response		
--	--	--	--	--	--

Id	SYN.T10-4 KAN DEZE WEL??				
Doel van de test	Verifieer dat de gegevens van een valide gebruiker kunnen worden opgehaald met een valide consumerkey.				
Preconditie	De gebruiker is aanwezig in het systeem.				
Postconditie	De gegevens van de gebruiker worden teruggegeven in de response.				
Verwijzing testboom	Gebruiker-Opvragen				
Testactie(s)	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call GET “accesstoken/” wordt uitgevoerd met de consumerkey “XXXXXX” en userId 1000	De gegevens worden teruggegeven in de response		

Id	SYN.T10-5				
Doel van de test	Verifieer dat de er geen gegevens van een valide gebruiker kunnen worden opgehaald met een invalide consumerkey.				
Preconditie	De gebruiker is aanwezig in het systeem.				
Postconditie	Er worden geen gegevens van de gebruiker teruggegeven in de response.				
Verwijzing testboom	Gebruiker-Opvragen				
Testactie(s)	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call GET “accesstoken/”	De response bevat een		

		wordt uitgevoerd met de consumerkey 123456 en userId 1000	message met de tekst: "No valid userkey supplied!"		
--	--	---	--	--	--

Id	SYN.T10-6				
Doel van de test	Verifieer dat de er geen gegevens van een valide gebruiker kunnen worden opgehaald met een invalide consumerkey.				
Preconditie	De gebruiker is aanwezig in het systeem.				
Postconditie	Er worden geen gegevens van de gebruiker teruggegeven in de response.				
Verwijzing testboom	Gebruiker-Opvragen				
Testactie(s)	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call GET "accesstoken/" wordt uitgevoerd met de consumerkey <32CHAR en userId 1000	De response bevat een message met de tekst: "No valid userkey supplied!"		

Id	SYN.T10-7				
Doel van de test	Verifieer dat de er geen gegevens van een valide gebruiker kunnen worden opgehaald met een invalide consumerkey.				
Preconditie	De gebruiker is aanwezig in het systeem.				
Postconditie	Er worden geen gegevens van de gebruiker teruggegeven in de response.				
Verwijzing testboom	Gebruiker-Opvragen				
Testactie(s)	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call GET	De response		

		“accesstoken/” wordt uitgevoerd met de consumerkey >32CHAR en userId 1000	bevat een message met de tekst: "No valid userkey supplied!"		
--	--	---	--	--	--

Id	SYN.T10-8				
Doel van de test	Verifieer dat de er geen gegevens van een valide gebruiker kunnen worden opgehaald met een invalide consumerkey.				
Preconditie	De gebruiker is aanwezig in het systeem.				
Postconditie	Er worden geen gegevens van de gebruiker teruggegeven in de response.				
Verwijzing testboom	Gebruiker-Opvragen				
Testactie(s)	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call GET “accesstoken/” wordt uitgevoerd met de consumerkey NULL en userId 1000	De response bevat een message met de tekst: "No valid userkey supplied!"		

Operator-Opvragen

Id	SYN.T11-1				
Doel van de test	Verifieer dat de gegevens van een valide operator worden opgehaald.				
Preconditie	Een valide operator is aanwezig in het systeem.				
Postconditie	De response bevat de gegevens van de operator.				
Verwijzing testboom	Operator-Opvragen				

m					
Testacti e(s)	Testa ctie Id	Testactie	Verwachte reactie	Concl usie	Opmer king
	1	De call GET “/operator/{operat orEncId}” wordt gedaan met “09889-abd7- 7882-abd7-7882- abd7-7882 ” als operatorEncId	De response bevat de gegevens van de opgevraagde valide operator. operatorDao2 = new OperatorDao(); operatorDao2.setExterna lId(2); operatorDao2.setFirstna me("Operator 2"); operatorDao2.setLastna me("Lastname 2"); operatorDao2.setEncId(" 09889-abd7-7882-abd7-7882- abd7-7882"); operatorDao2.setEmail(" oper@tor2.nl"); operatorDao2.setUserId(1200); operatorDao2.setActive(0);		

Id	SYN.T11-2
Doel van de test	Verifieer dat er geen gegevens van een valide operator worden opgehaald.
Preconditie	Een valide operator is aanwezig in het systeem.
Postconditi e	De response bevat geen gegevens van de operator.
Verwijzing testboom	Operator-Opvragen

Testactie(s))					
	Testacti e Id	Testactie	Verwachte reactie	Conclusi e	Opmerkin g
	1	De call GET "/operator/{operatorEncId }" wordt gedaan met "dfg-67j-456-dfg-67j- 456-dfg-67j-456-dfg-67j " als operatorEncId	De response bevat de message met de tekst "OperatorEncl d is not valid!"		

Id	SYN.T11-3				
Doel van de test	Verifieer dat er geen gegevens van een valide operator worden opgehaald.				
Preconditie	Een valide operator is aanwezig in het systeem.				
Postconditi e	De response bevat geen gegevens van de operator.				
Verwijzing testboom	Operator-Opvragen				
Testactie(s))					
	Testacti e Id	Testactie	Verwachte reactie	Conclusi e	Opmerkin g
	1	De call GET "/operator/{operatorEncId }" wordt gedaan met "456-dfg-67j" als operatorEncId	De response bevat de message met de tekst "OperatorEncl d is not valid!"		

Id	SYN.T11-4				
Doel van de test	Verifieer dat er geen gegevens van een valide operator worden opgehaald.				
Preconditie	Een valide operator is aanwezig in het systeem.				
Postconditi e	De response bevat geen gegevens van de operator.				
Verwijzing	Operator-Opvragen				

testboom					
Testactie(s)	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call GET “/operator/{operatorEncId}” wordt gedaan met “” als operatorEncId	De response bevat de message met de tekst “OperatorEncId is not valid!”		

Id	SYN.T11-5				
Doel van de test	Verifieer dat er geen gegevens van een valide operator worden opgehaald.				
Preconditie	Een valide operator is aanwezig in het systeem.				
Postconditie	De response bevat geen gegevens van de operator.				
Verwijzing testboom	Operator-Opvragen				
Testactie(s)	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call GET “/operator/{operatorEncId}” wordt gedaan met NULL als operatorEncId	De response bevat de message met de tekst “OperatorEncId is not valid!”	<pre> if(operatorEncId == null) { logger.error(errorText + "EncId is NULL!"); return false; } </pre>	

Incident-Customer-Satisfaction-Timeframe

Id	SYN.T12-1
Doel van de test	Verifieer dat de customer satisfaction kan worden opgehaald met een valide TimeFrame

Preconditie	Er zijn customer satisfactions aanwezig in het systeem.				
Postconditie	De customer satisfaction van de huidige periode wordt teruggegeven en van de periode ervoor.				
Verwijzing testboom	Incident-Customer-Satisfaction-Timeframe				
Testactie(s)					
	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call “incident/stats/customer-satisfaction” wordt uitgevoerd met als TimeFrame DAY en als userId 1000.			

Id	SYN.T12-2				
Doel van de test	Verifieer dat de customer satisfaction kan worden opgehaald met een valide TimeFrame				
Preconditie	Er zijn customer satisfactions aanwezig in het systeem.				
Postconditie	De customer satisfaction van de huidige periode wordt teruggegeven en van de periode ervoor.				
Verwijzing testboom	Incident-Customer-Satisfaction-Timeframe				
Testactie(s)					
	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call “incident/stats/customer-satisfaction” wordt uitgevoerd met als TimeFrame MONTH en als userId 1000.			

Id	SYN.T12-3				
Doel van de test	Verifieer dat de customer satisfaction kan worden opgehaald met een valide TimeFrame				

Preconditie	Er zijn customer satisfactions aanwezig in het systeem.				
Postconditie	De customer satisfaction van de huidige periode wordt teruggegeven en van de periode ervoor.				
Verwijzing testboom	Incident-Customer-Satisfaction-Timeframe				
Testactie(s)					
	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call “incident/stats/customer-satisfaction” wordt uitgevoerd met als TimeFrame Q1 en als userId 1000.			

Id	SYN.T12-4				
Doel van de test	Verifieer dat de customer satisfaction kan worden opgehaald met een valide TimeFrame				
Preconditie	Er zijn customer satisfactions aanwezig in het systeem.				
Postconditie	De customer satisfaction van de huidige periode wordt teruggegeven en van de periode ervoor.				
Verwijzing testboom	Incident-Customer-Satisfaction-Timeframe				
Testactie(s)					
	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call “incident/stats/customer-satisfaction” wordt uitgevoerd met als TimeFrame Q2 en als userId 1000.			

Id	SYN.T12-5				
Doel van de test	Verifieer dat de customer satisfaction kan worden opgehaald met een valide TimeFrame				

Preconditie	Er zijn customer satisfactions aanwezig in het systeem.				
Postconditie	De customer satisfaction van de huidige periode wordt teruggegeven en van de periode ervoor.				
Verwijzing testboom	Incident-Customer-Satisfaction-Timeframe				
Testactie(s)					
	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call “incident/stats/customer-satisfaction” wordt uitgevoerd met als TimeFrame Q3 en als userId 1000.			

Id	SYN.T12-6				
Doel van de test	Verifieer dat de customer satisfaction kan worden opgehaald met een valide TimeFrame				
Preconditie	Er zijn customer satisfactions aanwezig in het systeem.				
Postconditie	De customer satisfaction van de huidige periode wordt teruggegeven en van de periode ervoor.				
Verwijzing testboom	Incident-Customer-Satisfaction-Timeframe				
Testactie(s)					
	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call “incident/stats/customer-satisfaction” wordt uitgevoerd met als TimeFrame Q4 en als userId 1000.			

Id	SYN.T12-7				
Doel van de test	Verifieer dat de customer satisfaction kan worden opgehaald met een valide TimeFrame				

Preconditie	Er zijn customer satisfactions aanwezig in het systeem.				
Postconditie	De customer satisfaction van de huidige periode wordt teruggegeven en van de periode ervoor.				
Verwijzing testboom	Incident-Customer-Satisfaction-Timeframe				
Testactie(s)					
	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call “incident/stats/customer-satisfaction” wordt uitgevoerd met als TimeFrame YEAR en als userId 1000.			

Id	SYN.T12-8				
Doel van de test	Verifieer dat er customer satisfaction kan worden opgehaald met een niet valide TimeFrame				
Preconditie	Er zijn customer satisfactions aanwezig in het systeem.				
Postconditie	Er wordt customer satisfaction van de huidige periode teruggegeven en van de periode ervoor met als TimeFrame DAY.				
Verwijzing testboom	Incident-Customer-Satisfaction-Timeframe				
Testactie(s)					
	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call “incident/stats/customer-satisfaction” wordt uitgevoerd met als TimeFrame NULL en als userId 1000.	De customer satisfaction voor TimeFrame DAY worden teruggegeven		

Id	SYN.T12-9				
Doel van de test	Verifieer dat er customer satisfaction kan worden opgehaald met een niet valide TimeFrame				

Preconditie	Er zijn customer satisfactions aanwezig in het systeem.				
Postconditie	Er wordt customer satisfaction van de huidige periode teruggegeven en van de periode ervoor met als TimeFrame DAY.				
Verwijzing testboom	Incident-Customer-Satisfaction-Timeframe				
Testactie(s)	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call “incident/stats/customer-satisfaction” wordt uitgevoerd met als TimeFrame 2016 en als userId 1000.	De customer satisfaction voor TimeFrame DAY worden teruggegeven.		

Id	SYN.T12-10 KAN DEZE WEL??				
Doel van de test					
Preconditie					
Postconditie					
Verwijzing testboom	Incident-Customer-Satisfaction-Timeframe				
Testactie(s)	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call “incident/stats/customer-satisfaction” wordt uitgevoerd met als TimeFrame Q1 en als userId 1000.			

Id	SYN.T12-11 KAN DEZE WEL??				
Doel van de test					
Preconditie					

Postconditie					
Verwijzing testboom	Incident-Customer-Satisfaction-Timeframe				
Testactie(s)					
	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1				

Id	SYN.T12-12 KAN DEZE WEL??				
Doel van de test					
Preconditie					
Postconditie					
Verwijzing testboom	Incident-Customer-Satisfaction-Timeframe				
Testactie(s)					
	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1				

Incident-Customer-Satisfaction-Relative

Id	SYN.T13-1				
Doel van de test	Verifieer dat er customer satisfaction kan worden opgehaald met een valide TimeFrame				
Preconditie	Er zijn customer satisfactions aanwezig in het systeem.				
Postconditie	Er wordt customer satisfaction van de huidige periode teruggegeven en van dezelfde periode het jaar ervoor.				
Verwijzing testboom	Incident-Customer-Satisfaction-Relative				
Testactie(s)					
	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call			

		“incident/stats/customer-satisfaction/relative-to“ wordt uitgevoerd met TimeFrame MONTH en userId 1000			
--	--	---	--	--	--

Id	SYN.T13-2				
Doel van de test	Verifieer dat er customer satisfaction kan worden opgehaald met een valide TimeFrame				
Preconditie	Er zijn customer satisfactions aanwezig in het systeem.				
Postconditie	Er wordt customer satisfaction van de huidige periode teruggegeven en van dezelfde periode het jaar ervoor.				
Verwijzing testboom	Incident-Customer-Satisfaction-Relative				
Testactie(s)	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call “incident/stats/customer-satisfaction/relative-to“ wordt uitgevoerd met TimeFrame Q1 en userId 1000			

Id	SYN.T13-3				
Doel van de test	Verifieer dat er customer satisfaction kan worden opgehaald met een valide TimeFrame				
Preconditie	Er zijn customer satisfactions aanwezig in het systeem.				
Postconditie	Er wordt customer satisfaction van de huidige periode teruggegeven en van dezelfde periode het jaar ervoor.				
Verwijzing testboom	Incident-Customer-Satisfaction-Relative				
Testactie(s)	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking

	1	De call “incident/stats/customer-satisfaction/relative-to“ wordt uitgevoerd met TimeFrame Q2 en userId 1000			
--	---	---	--	--	--

Id	SYN.T13-4				
Doel van de test	Verifieer dat er customer satisfaction kan worden opgehaald met een valide TimeFrame				
Preconditie	Er zijn customer satisfactions aanwezig in het systeem.				
Postconditie	Er wordt customer satisfaction van de huidige periode teruggegeven en van dezelfde periode het jaar ervoor.				
Verwijzing testboom	Incident-Customer-Satisfaction-Relative				
Testactie(s)	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call “incident/stats/customer-satisfaction/relative-to“ wordt uitgevoerd met TimeFrame Q3 en userId 1000			

Id	SYN.T13-5				
Doel van de test	Verifieer dat er customer satisfaction kan worden opgehaald met een valide TimeFrame				
Preconditie	Er zijn customer satisfactions aanwezig in het systeem.				
Postconditie	Er wordt customer satisfaction van de huidige periode teruggegeven en van dezelfde periode het jaar ervoor.				
Verwijzing testboom	Incident-Customer-Satisfaction-Relative				
Testactie(s)	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking

	1	De call “incident/stats/customer-satisfaction/relative-to“ wordt uitgevoerd met TimeFrame Q4 en userId 1000			
--	---	---	--	--	--

Id	SYN.T13-6				
Doel van de test	Verifieer dat er customer satisfaction kan worden opgehaald met een valide TimeFrame				
Preconditie	Er zijn customer satisfactions aanwezig in het systeem.				
Postconditie	Er wordt customer satisfaction van de huidige periode teruggegeven en van dezelfde periode het jaar ervoor.				
Verwijzing testboom	Incident-Customer-Satisfaction-Relative				
Testactie(s)	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call “incident/stats/customer-satisfaction/relative-to“ wordt uitgevoerd met TimeFrame YEAR en userId 1000			

Id	SYN.T13-7				
Doel van de test	Verifieer dat er customer satisfaction kan worden opgehaald met een niet valide TimeFrame				
Preconditie	Er zijn customer satisfactions aanwezig in het systeem.				
Postconditie	Er wordt customer satisfaction van de huidige Maand teruggegeven en van dezelfde maand het jaar ervoor.				
Verwijzing testboom	Incident-Customer-Satisfaction-Relative				
Testactie(s)	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking

	1	De call “incident/stats/customer-satisfaction/relative-to” wordt uitgevoerd met TimeFrame NULL en userId 1000			
--	---	---	--	--	--

Id	SYN.T13-8				
Doel van de test	Verifieer dat er customer satisfaction kan worden opgehaald met een niet valide TimeFrame				
Preconditie	Er zijn customer satisfactions aanwezig in het systeem.				
Postconditie	Er wordt customer satisfaction van de huidige Maand teruggegeven en van dezelfde maand het jaar ervoor.				
Verwijzing testboom	Incident-Customer-Satisfaction-Relative				
Testactie(s)	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call “incident/stats/customer-satisfaction/relative-to” wordt uitgevoerd met TimeFrame 2016 en userId 1000			

Id	SYN.T13-9				
Doel van de test	Verifieer dat er customer satisfaction kan worden opgehaald met een niet valide TimeFrame				
Preconditie	Er zijn customer satisfactions aanwezig in het systeem.				
Postconditie	Er wordt customer satisfaction van de huidige Maand teruggegeven en van dezelfde maand het jaar ervoor.				
Verwijzing testboom	Incident-Customer-Satisfaction-Relative				
Testactie(s)	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking

	1	De call “incident/stats/customer- satisfaction/relative-to” wordt uitgevoerd met TimeFrame DAY en userId 1000			
--	---	--	--	--	--

Id	SYN.T13-10 KAN DEZE WEL??				
Doel van de test	Verifieer dat er customer satisfaction kan worden opgehaald met een valide TimeFrame				
Preconditie	Er zijn customer satisfactions aanwezig in het systeem.				
Postconditie	Er wordt customer satisfaction van de huidige periode teruggegeven en van dezelfde periode het jaar ervoor.				
Verwijzing testboom	Incident-Customer-Satisfaction-Relative				
Testactie(s)					
	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1				

Id	SYN.T13-11 KAN DEZE WEL??				
Doel van de test	Verifieer dat er customer satisfaction kan worden opgehaald met een valide TimeFrame				
Preconditie	Er zijn customer satisfactions aanwezig in het systeem.				
Postconditie	Er wordt customer satisfaction van de huidige periode teruggegeven en van dezelfde periode het jaar ervoor.				
Verwijzing testboom	Incident-Customer-Satisfaction-Relative				
Testactie(s)					
	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1				

Id	SYN.T13-12 KAN DEZE WEL??				
Doel van de test	Verifieer dat er customer satisfaction kan worden opgehaald met een valide TimeFrame				

Preconditie	Er zijn customer satisfactions aanwezig in het systeem.				
Postconditie	Er wordt customer satisfaction van de huidige periode teruggegeven en van dezelfde periode het jaar ervoor.				
Verwijzing testboom	Incident-Customer-Satisfaction-Relative				
Testactie(s)					
	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1				

Incident-Operator-Scores-Timeframe

Id	SYN.T14-1				
Doel van de test	Verifieer dat er operator scores kunnen worden opgehaald met een valide TimeFrame.				
Preconditie	Er zijn operator scores aanwezig in het systeem.				
Postconditie	Er worden operator scores van de huidige periode en dezelfde periode ervoor teruggegeven.				
Verwijzing testboom	Incident-Operator-Scores-Timeframe				
Testactie(s)					
	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call “incident/stats/operator-scores” wordt gedaan met TimeFrame DAY en userId 1000			

Id	SYN.T14-2				
Doel van de test	Verifieer dat er operator scores kunnen worden opgehaald met een valide TimeFrame.				
Preconditie	Er zijn operator scores aanwezig in het systeem.				
Postconditie	Er worden operator scores van de huidige periode en dezelfde periode ervoor teruggegeven.				
Verwijzing	Incident-Operator-Scores-Timeframe				

testboom					
Testactie(s)					
	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call “incident/stats/operator-scores” wordt gedaan met TimeFrame MONTH en userId 1000			

Id	SYN.T14-3				
Doel van de test	Verifieer dat er operator scores kunnen worden opgehaald met een valide TimeFrame.				
Preconditie	Er zijn operator scores aanwezig in het systeem.				
Postconditie	Er worden operator scores van de huidige periode en dezelfde periode ervoor teruggegeven.				
Verwijzing testboom	Incident-Operator-Scores-Timeframe				
Testactie(s)					
	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call “incident/stats/operator-scores” wordt gedaan met TimeFrame Q1 en userId 1000			

Id	SYN.T14-4				
Doel van de test	Verifieer dat er operator scores kunnen worden opgehaald met een valide TimeFrame.				
Preconditie	Er zijn operator scores aanwezig in het systeem.				
Postconditie	Er worden operator scores van de huidige periode en dezelfde periode ervoor teruggegeven.				
Verwijzing testboom	Incident-Operator-Scores-Timeframe				

Testactie(s)					
	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call “incident/ stats/operator-scores ” wordt gedaan met TimeFrame Q2 en userId 1000			

Id	SYN.T14-5				
Doel van de test	Verifieer dat er operator scores kunnen worden opgehaald met een valide TimeFrame.				
Preconditie	Er zijn operator scores aanwezig in het systeem.				
Postconditie	Er worden operator scores van de huidige periode en dezelfde periode ervoor teruggegeven.				
Verwijzing testboom	Incident-Operator-Scores-Timeframe				
Testactie(s)					
	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call “incident/ stats/operator-scores ” wordt gedaan met TimeFrame Q3 en userId 1000			

Id	SYN.T14-6				
Doel van de test	Verifieer dat er operator scores kunnen worden opgehaald met een valide TimeFrame.				
Preconditie	Er zijn operator scores aanwezig in het systeem.				
Postconditie	Er worden operator scores van de huidige periode en dezelfde periode ervoor teruggegeven.				
Verwijzing testboom	Incident-Operator-Scores-Timeframe				
Testactie(s)					

	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call “incident/ stats/operator-scores ” wordt gedaan met TimeFrame Q4 en userId 1000			

Id	SYN.T14-7				
Doel van de test	Verifieer dat er operator scores kunnen worden opgehaald met een valide TimeFrame.				
Preconditie	Er zijn operator scores aanwezig in het systeem.				
Postconditie	Er worden operator scores van de huidige periode en dezelfde periode ervoor teruggegeven.				
Verwijzing testboom	Incident-Operator-Scores-Timeframe				
Testactie(s)	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call “incident/ stats/operator-scores ” wordt gedaan met TimeFrame YEAR en userId 1000			

Id	SYN.T14-8				
Doel van de test	Verifieer dat er operator scores kunnen worden opgehaald met een valide TimeFrame.				
Preconditie	Er zijn operator scores aanwezig in het systeem.				
Postconditie	Er worden operator scores van de huidige dag en de dag ervoor teruggegeven.				
Verwijzing testboom	Incident-Operator-Scores-Timeframe				
Testactie(s)	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking

	1	De call “incident/stats/operator-scores” wordt gedaan met TimeFrame NULL en userId 1000			
--	---	--	--	--	--

Id	SYN.T14-9				
Doel van de test	Verifieer dat er operator scores kunnen worden opgehaald met een valide TimeFrame.				
Preconditie	Er zijn operator scores aanwezig in het systeem.				
Postconditie	Er worden operator scores van de huidige dag en de dag ervoor teruggegeven.				
Verwijzing testboom	Incident-Operator-Scores-Timeframe				
Testactie(s)	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call “incident/stats/operator-scores” wordt gedaan met TimeFrame 2016 en userId 1000			

Id	SYN.T14-10 KAN DEZE WEL??				
Doel van de test	Verifieer dat er operator scores kunnen worden opgehaald met een valide TimeFrame.				
Preconditie	Er zijn operator scores aanwezig in het systeem.				
Postconditie	Er worden operator scores van de huidige periode en dezelfde periode ervoor teruggegeven.				
Verwijzing testboom	Incident-Operator-Scores-Timeframe				
Testactie(s)	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call “incident/stats/operator-			

		scores” wordt gedaan met TimeFrame DAY en userId 1000			
--	--	---	--	--	--

Id	SYN.T14-11 KAN DEZE WEL??				
Doel van de test	Verifieer dat er operator scores kunnen worden opgehaald met een valide TimeFrame.				
Preconditie	Er zijn operator scores aanwezig in het systeem.				
Postconditie	Er worden operator scores van de huidige periode en dezelfde periode ervoor teruggegeven.				
Verwijzing testboom	Incident-Operator-Scores-Timeframe				
Testactie(s)	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call “incident/stats/operator-scores” wordt gedaan met TimeFrame DAY en userId 1000			

Id	SYN.T14-12 KAN DEZE WEL??				
Doel van de test	Verifieer dat er operator scores kunnen worden opgehaald met een valide TimeFrame.				
Preconditie	Er zijn operator scores aanwezig in het systeem.				
Postconditie	Er worden operator scores van de huidige periode en dezelfde periode ervoor teruggegeven.				
Verwijzing testboom	Incident-Operator-Scores-Timeframe				
Testactie(s)	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call “incident/stats/operator-scores” wordt gedaan met			

		TimeFrame DAY en userId 1000			
--	--	---------------------------------	--	--	--

Incident-Operator-Scores-Relative

Id	SYN.T15-1				
Doel van de test	Verifieer dat er operator scores kunnen worden opgehaald met een valide TimeFrame van de huidige periode en dezelfde periode het jaar ervoor.				
Preconditie	Er zijn operator scores aanwezig in het systeem.				
Postconditie	Er worden operator scores van de huidige periode en dezelfde periode het jaar ervoor teruggegeven.				
Verwijzing testboom	Incident-Operator-Scores-Relative				
Testactie(s)	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call “incident/stats/operator-scores/relative-to” wordt uitgevoerd met als TimeFrame MONTH en als userId 1000			

Id	SYN.T15-2				
Doel van de test	Verifieer dat er operator scores kunnen worden opgehaald met een valide TimeFrame van de huidige periode en dezelfde periode het jaar ervoor.				
Preconditie	Er zijn operator scores aanwezig in het systeem.				
Postconditie	Er worden operator scores van de huidige periode en dezelfde periode het jaar ervoor teruggegeven.				
Verwijzing testboom	Incident-Operator-Scores-Relative				
Testactie(s)	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call			

		“incident/stats/operator-scores/relative-to“ wordt uitgevoerd met als TimeFrame Q1 en als userId 1000			
--	--	---	--	--	--

Id	SYN.T15-3				
Doel van de test	Verifieer dat er operator scores kunnen worden opgehaald met een valide TimeFrame van de huidige periode en dezelfde periode het jaar ervoor.				
Preconditie	Er zijn operator scores aanwezig in het systeem.				
Postconditie	Er worden operator scores van de huidige periode en dezelfde periode het jaar ervoor teruggegeven.				
Verwijzing testboom	Incident-Operator-Scores-Relative				
Testactie(s)	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call “incident/stats/operator-scores/relative-to“ wordt uitgevoerd met als TimeFrame Q2 en als userId 1000			

Id	SYN.T15-4				
Doel van de test	Verifieer dat er operator scores kunnen worden opgehaald met een valide TimeFrame van de huidige periode en dezelfde periode het jaar ervoor.				
Preconditie	Er zijn operator scores aanwezig in het systeem.				
Postconditie	Er worden operator scores van de huidige periode en dezelfde periode het jaar ervoor teruggegeven.				
Verwijzing testboom	Incident-Operator-Scores-Relative				
Testactie(s)	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking

	1	De call “incident/stats/operator-scores/relative-to” wordt uitgevoerd met als TimeFrame Q3 en als userId 1000			
--	---	--	--	--	--

Id	SYN.T15-5				
Doel van de test	Verifieer dat er operator scores kunnen worden opgehaald met een valide TimeFrame van de huidige periode en dezelfde periode het jaar ervoor.				
Preconditie	Er zijn operator scores aanwezig in het systeem.				
Postconditie	Er worden operator scores van de huidige periode en dezelfde periode het jaar ervoor teruggegeven.				
Verwijzing testboom	Incident-Operator-Scores-Relative				
Testactie(s)	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call “incident/stats/operator-scores/relative-to” wordt uitgevoerd met als TimeFrame Q4 en als userId 1000			

Id	SYN.T15-6				
Doel van de test	Verifieer dat er operator scores kunnen worden opgehaald met een valide TimeFrame van de huidige periode en dezelfde periode het jaar ervoor.				
Preconditie	Er zijn operator scores aanwezig in het systeem.				
Postconditie	Er worden operator scores van de huidige periode en dezelfde periode het jaar ervoor teruggegeven.				
Verwijzing testboom	Incident-Operator-Scores-Relative				
Testactie(s)	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking

	1	De call “incident/stats/operator-scores/relative-to” wordt uitgevoerd met als TimeFrame YEAR en als userId 1000			
--	---	--	--	--	--

Id	SYN.T15-7				
Doel van de test	Verifieer dat er operator scores kunnen worden opgehaald met een valide TimeFrame van de huidige periode en dezelfde periode het jaar ervoor.				
Preconditie	Er zijn operator scores aanwezig in het systeem.				
Postconditie	Er worden operator scores van de huidige maand en dezelfde maand het jaar ervoor teruggegeven.				
Verwijzing testboom	Incident-Operator-Scores-Relative				
Testactie(s)	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call “incident/stats/operator-scores/relative-to” wordt uitgevoerd met als TimeFrame NULL en als userId 1000			

Id	SYN.T15-8				
Doel van de test	Verifieer dat er operator scores kunnen worden opgehaald met een valide TimeFrame van de huidige periode en dezelfde periode het jaar ervoor.				
Preconditie	Er zijn operator scores aanwezig in het systeem.				
Postconditie	Er worden operator scores van de huidige maand en dezelfde maand het jaar ervoor teruggegeven.				
Verwijzing testboom	Incident-Operator-Scores-Relative				
Testactie(s)	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking

	1	De call “incident/stats/operator-scores/relative-to” wordt uitgevoerd met als TimeFrame 2016 en als userId 1000			
--	---	--	--	--	--

Id	SYN.T15-9				
Doel van de test	Verifieer dat er operator scores kunnen worden opgehaald met een valide TimeFrame van de huidige periode en dezelfde periode het jaar ervoor.				
Preconditie	Er zijn operator scores aanwezig in het systeem.				
Postconditie	Er worden operator scores van de huidige maand en dezelfde maand het jaar ervoor teruggegeven.				
Verwijzing testboom	Incident-Operator-Scores-Relative				
Testactie(s)	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call “incident/stats/operator-scores/relative-to” wordt uitgevoerd met als TimeFrame DAY en als userId 1000			

Id	SYN.T15-10 KAN DEZE WEL??				
Doel van de test	Verifieer dat er operator scores kunnen worden opgehaald met een valide TimeFrame van de huidige periode en dezelfde periode het jaar ervoor.				
Preconditie	Er zijn operator scores aanwezig in het systeem.				
Postconditie	Er worden operator scores van de huidige periode en dezelfde periode het jaar ervoor teruggegeven.				
Verwijzing testboom	Incident-Operator-Scores-Relative				
Testactie(s)					
	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking

	1				
--	---	--	--	--	--

Id	SYN.T15-11 KAN DEZE WEL??				
Doel van de test	Verifieer dat er operator scores kunnen worden opgehaald met een valide TimeFrame van de huidige periode en dezelfde periode het jaar ervoor.				
Preconditie	Er zijn operator scores aanwezig in het systeem.				
Postconditie	Er worden operator scores van de huidige periode en dezelfde periode het jaar ervoor teruggegeven.				
Verwijzing testboom	Incident-Operator-Scores-Relative				
Testactie(s)					
	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1				

Id	SYN.T15-12 KAN DEZE WEL??				
Doel van de test	Verifieer dat er operator scores kunnen worden opgehaald met een valide TimeFrame van de huidige periode en dezelfde periode het jaar ervoor.				
Preconditie	Er zijn operator scores aanwezig in het systeem.				
Postconditie	Er worden operator scores van de huidige periode en dezelfde periode het jaar ervoor teruggegeven.				
Verwijzing testboom	Incident-Operator-Scores-Relative				
Testactie(s)					
	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1				

Incident-Opvragen

Id	SYN.T16-1				
Doel van de test	Verifieer dat er gegevens van een incident kunnen worden opgevraagd voor een valide feedbackEncId.				
Preconditie	Er is een feedback aanwezig voor het valide feedbackEncId.				

Postconditie	De informatie voor in de feedback mail wordt teruggegeven voor het gegeven feedbackEncId.				
Verwijzing testboom	Incident-Opvragen				
Testactie(s)					
	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call GET "incident/{feedbackEncId}" wordt aangeroepen met als feedbackEncId "XXXXX" en als userId 1000			

Id	SYN.T16-2				
Doel van de test	Verifieer dat er geen gegevens van een incident kunnen worden opgevraagd voor een niet valide feedbackEncId.				
Preconditie	Er is een feedback aanwezig voor het valide feedbackEncId.				
Postconditie	Er wordt een foutmelding gegeven.				
Verwijzing testboom	Incident-Opvragen				
Testactie(s)					
	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call GET "incident/{feedbackEncId}" wordt aangeroepen met als feedbackEncId ">35CHAR" en als userId 1000	Response bevat een message met de tekst: "No valid encId supplied!"		

Id	SYN.T16-3				
Doel van de test	Verifieer dat er geen gegevens van een incident kunnen worden opgevraagd voor een niet valide feedbackEncId.				

Preconditie	Er is een feedback aanwezig voor het valide feedbackEncId.				
Postconditie	Er wordt een foutmelding gegeven.				
Verwijzing testboom	Incident-Opvragen				
Testactie(s)	Testacti e Id	Testactie	Verwacht e reactie	Conclusi e	Opmerkin g
	1	De call GET "incident/{feedbackEncId} " wordt aangeroepen met als feedbackEncId "<35CHAR" en als userId 1000	Response bevat een message met de tekst: "No valid encId supplied!"		

Id	SYN.T16-4				
Doel van de test	Verifieer dat er geen gegevens van een incident kunnen worden opgevraagd voor een niet valide feedbackEncId.				
Preconditie	Er is een feedback aanwezig voor het valide feedbackEncId.				
Postconditie	Er wordt een foutmelding gegeven.				
Verwijzing testboom	Incident-Opvragen				
Testactie(s)	Testacti e Id	Testactie	Verwacht e reactie	Conclusi e	Opmerkin g
	1	De call GET "incident/{feedbackEncId} " wordt aangeroepen met als feedbackEncId "" en als userId 1000	Response bevat een message met de tekst: "No valid encId supplied!"		

Id	SYN.T16-5				
----	-----------	--	--	--	--

Doel van de test	Verifieer dat er geen gegevens van een incident kunnen worden opgevraagd voor een niet valide feedbackEncId.				
Preconditie	Er is een feedback aanwezig voor het valide feedbackEncId.				
Postconditie	Er wordt een foutmelding gegeven.				
Verwijzing testboom	Incident-Opvragen				
Testactie(s)					
	Testacti e Id	Testactie	Verwacht e reactie	Conclusi e	Opmerkin g
	1	De call GET "incident/{feedbackEncId} " wordt aangeroepen met als feedbackEncId "NULL" en als userId 1000	Response bevat een message met de tekst: "No valid encId supplied!"		

Incident-Versturen

Id	SYN.T17-1				
Doel van de test	Verifieer dat er een feedback mail van het valide feedbackEncId is verzonden aan de beoordelaar.				
Preconditie	Er is een feedback aanwezig voor het valide feedbackEncId.				
Postconditie	De mail is verzonden aan de beoordelaar.				
Verwijzing testboom	Incident-Versturen				
Testactie(s))					
	Testacti e Id	Testactie	Verwacht e reactie	Conclusi e	Opmerkin g
	1	De call "incident/{feedbackEncId}/sen d" wordt aangeroepen met feedbackEncId "XXXXXX" en als userId 1000	De response heeft een message met de tekst:		

			"Feedbac k mail send for encId: " + XXXXX X + "."		
--	--	--	--	--	--

Id	SYN.T17-2				
Doel van de test	Verifieer dat er geen feedback mail van het valide feedbackEncId is verzonden aan de beoordelaar.				
Preconditie	Er is een feedback aanwezig voor het valide feedbackEncId.				
Postconditie	De mail is niet verzonden aan de beoordelaar.				
Verwijzing testboom	Incident-Versturen				
Testactie(s)					
	Testacti e Id	Testactie	Verwacht e reactie	Conclusi e	Opmerkin g
	1	De call “incident/{feedbackEncId}/sen d“ wordt aangeroepen met feedbackEncId “>35CHAR” en als userId 1000	De response heeft een message met de tekst: "No valid encId supplied! "		

Id	SYN.T17-3				
Doel van de test	Verifieer dat er geen feedback mail van het valide feedbackEncId is verzonden aan de beoordelaar.				
Preconditie	Er is een feedback aanwezig voor het valide feedbackEncId.				
Postconditie	De mail is niet verzonden aan de beoordelaar.				
Verwijzing	Incident-Versturen				

testboom					
Testactie(s)					
	Testacti e Id	Testactie	Verwacht e reactie	Conclusi e	Opmerkin g
	1	De call “incident/{feedbackEncId}/send” wordt aangeroepen met feedbackEncId “<35CHAR” en als userId 1000	De response heeft een message met de tekst: "No valid encId supplied!"		

Id	SYN.T17-4				
Doel van de test	Verifieer dat er geen feedback mail van het valide feedbackEncId is verzonden aan de beoordelaar.				
Preconditie	Er is een feedback aanwezig voor het valide feedbackEncId.				
Postconditie	De mail is niet verzonden aan de beoordelaar.				
Verwijzing testboom	Incident-Versturen				
Testactie(s)					
	Testacti e Id	Testactie	Verwacht e reactie	Conclusi e	Opmerkin g
	1	De call “incident/{feedbackEncId}/send” wordt aangeroepen met feedbackEncId “” en als userId 1000	De response heeft een message met de tekst: "No valid encId supplied!"		

Id	SYN.T17-5				
Doel van de test	Verifieer dat er geen feedback mail van het valide feedbackEncId is verzonden aan de beoordelaar.				
Preconditie	Er is een feedback aanwezig voor het valide feedbackEncId.				
Postconditie	De mail is niet verzonden aan de beoordelaar.				
Verwijzing testboom	Incident-Versturen				
Testactie(s)					
	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call “incident/{feedbackEncId}/send” wordt aangeroepen met feedbackEncId “NULL” en als userId 1000	De response heeft een message met de tekst: "No valid encId supplied!"		

Gebruiker-Toevoegen

Id	SYN.T18-1				
Doel van de test	Controleer dat er een nieuwe gebruiker is toegevoegd.				
Preconditie	De gebruiker is niet aanwezig in het systeem.				
Postconditie	De gebruiker is aanwezig in het systeem.				
Verwijzing testboom	Gebruiker-Toevoegen				
Testactie(s)					
	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call POST “accesstoken” wordt	De gebruiker is		

		uitgevoerd met de body: "XXXXXX" en userId 1000.	aangemaakt.		
--	--	--	-------------	--	--

Id	SYN.T18-2				
Doel van de test	Controleer dat er geen nieuwe gebruiker is toegevoegd.				
Preconditie	De gebruiker is niet aanwezig in het systeem.				
Postconditie	De gebruiker is niet aanwezig in het systeem.				
Verwijzing testboom	Gebruiker-Toevoegen				
Testactie(s)	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call POST "accesstoken" wordt uitgevoerd met de body: "" en userId 1000.	De response bevat een message met de tekst: "User not valid!"		

Id	SYN.T18-3				
Doel van de test	Controleer dat er geen nieuwe gebruiker is toegevoegd.				
Preconditie	De gebruiker is niet aanwezig in het systeem.				
Postconditie	De gebruiker is niet aanwezig in het systeem.				
Verwijzing testboom	Gebruiker-Toevoegen				
Testactie(s)	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call POST "accesstoken" wordt uitgevoerd met de body: "NULL" en userId 1000.	De response bevat een message met de tekst: "User not valid!"		

Id	SYN.T18-4 KAN DEZE WEL??				
Doel van de test	Controleer dat er geen nieuwe gebruiker is toegevoegd.				
Preconditie	De gebruiker is niet aanwezig in het systeem.				
Postconditie	De gebruiker is niet aanwezig in het systeem.				
Verwijzing testboom	Gebruiker-Toevoegen				
Testactie(s)					
	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call POST “accesstoken” wordt uitgevoerd met de body: “XXXXXX” en userId 1000.	De gebruiker is aangemaakt.		

Id	SYN.T18-5 KAN DEZE WEL??				
Doel van de test	Controleer dat er geen nieuwe gebruiker is toegevoegd.				
Preconditie	De gebruiker is niet aanwezig in het systeem.				
Postconditie	De gebruiker is niet aanwezig in het systeem.				
Verwijzing testboom	Gebruiker-Toevoegen				
Testactie(s)					
	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call POST “accesstoken” wordt uitgevoerd met de body: “XXXXXX” en userId 1000.	De gebruiker is aangemaakt.		

Id	SYN.T18-6 KAN DEZE WEL??				
Doel van de	Controleer dat er geen nieuwe gebruiker is toegevoegd.				

test					
Preconditie	De gebruiker is niet aanwezig in het systeem.				
Postconditie	De gebruiker is niet aanwezig in het systeem.				
Verwijzing testboom	Gebruiker-Toevoegen				
Testactie(s)					
	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call POST “accesstoken” wordt uitgevoerd met de body: “XXXXXX” en userId 1000.	De gebruiker is aangemaakt.		

Operator-Update

Id	SYN.T19-1				
Doel van de test	Verifieer dat een operator kan worden aangepast.				
Preconditie	Er is een valide operator aanwezig.				
Postconditie	De operator is aangepast.				
Verwijzing testboom	Operator-Update				
Testactie(s)					
	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call PUT “/operator” wordt uitgevoerd met als body: “dao1 + “nieuwe voornaam”” en als userId 1000.	De operator is aangepast met de nieuwe gegevens.		

Id	SYN.T19-2				
Doel van de	Verifieer dat een operator niet kan worden aangepast met een NULL body.				

test					
Preconditie	Er is een valide operator aanwezig.				
Postconditie	De operator is niet aangepast.				
Verwijzing testboom	Operator-Updates				
Testacties(s)	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call PUT "/operator" wordt uitgevoerd met als body: "NULL" en als userId 1000.	De operator is niet aangepast met de nieuwe gegevens. assertThat(operatorResponse.getMessage(), is("Operator not valid!"));	<pre> if(operator == null) { logger.error(operatorErrorText + "Body is null!"); return false; } </pre>	

Id	SYN.T19-3
Doel van de test	Verifieer dat een operator niet kan worden aangepast met niet niet valide operator.
Preconditie	Er is een valide operator aanwezig.
Postconditie	De operator is niet aangepast.
Verwijzing testboom	Operator-Updates
Testactie(

s)	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call PUT “/operator” wordt uitgevoerd met als body: “NIET VALIDE” en als userId 1000.	De operator is niet aangepast met de nieuwe gegevens. assertThat(operatorResponse.getMessage(), is("Operator not valid!"));		

Operator-Toevoegen

Id	SYN.T20-1				
Doel van de test	Verifieer dat er een nieuwe operator is toegevoegd.				
Preconditie	De operator is nog niet aanwezig in het systeem.				
Postconditie	De operator is aanwezig in het systeem.				
Verwijzing testboom	Operator-Toevoegen				
Testactie(s)	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call POST “/operator” wordt uitgevoerd met de body: operatorDao2.setExternalId (2);	De operator is toegevoegd. operatorDao1.setExternalId (1); operatorDao1.setFir		

	<pre> operatorDao2.setFirstname("Operator 2"); operatorDao2.setLastname("Lastname 2"); operatorDao2.setEncId("09889-abd7-7882-abd7-7882-abd7-7882"); operatorDao2.setEmail("oper@tor2.nl"); operatorDao2.setUserId(1200); operatorDao2.setActive(0); " en userId 1000. </pre>	<pre> sname("Operator 1"); operatorDao1.setLastname("Lastname"); operatorDao1.setEncId("345-0-09df-3209-09df-3209-09df-3209"); operatorDao1.setEmail("oper@tor.nl"); operatorDao1.setUserId(1100); operatorDao1.setActive(1); </pre>		
--	---	---	--	--

Id	SYN.T20-2				
Doel van de test	Verifieer dat er geen nieuwe operator is toegevoegd.				
Preconditie	De operator is nog niet aanwezig in het systeem.				
Postconditie	De operator is nog niet aanwezig in het systeem.				
Verwijzing testboom	Operator-Toevoegen				
Testactie(s)					
	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call POST "/operator" wordt uitgevoerd met de body: "NULL" en userId 1000.	De response bevat een message met de tekst: "Could not add new operator!"		

Id	SYN.T20-3				
Doel van	Verifieer dat er geen nieuwe operator is toegevoegd.				

de test					
Preconditie	De operator is nog niet aanwezig in het systeem.				
Postconditie	De operator is nog niet aanwezig in het systeem.				
Verwijzing testboom	Operator-Toevoegen				
Testactie(s)	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call POST “/operator” wordt uitgevoerd met de body: “operatorDao1.setExternalId(1); operatorDao1.setLastname("Lastname"); operatorDao1.setEmail("operator.nl"); operatorDao1.setUserId(1100) ; operatorDao1.setActive(1); ” en userId 1000.	De respons bevat een message met de tekst: "Could not add new operator!"		

Incident-Toevoegen

Id	SYN.T21-1
Doel van de test	Verifieer dat er een nieuw incident is aangemaakt.
Preconditie	Het incident is nog niet aanwezig in het systeem.
Postconditie	Het incident is aanwezig in het systeem.
Verwijzing testboom	Incident-Toevoegen

Testactie(s)					
	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call POST “/incident” wordt uitgevoerd met de body: “XXXXXX” en userId 1000.	Het incident is toegevoegd.		

Id	SYN.T21-2				
Doel van de test	Verifieer dat er geen nieuw incident is aangemaakt.				
Preconditie	Het incident is nog niet aanwezig in het systeem.				
Postconditie	Het incident is niet toegevoegd aan het systeem.				
Verwijzing testboom	Incident-Toevoegen				
Testactie(s)					
	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call POST “/incident” wordt uitgevoerd met de body: “NULL” en userId 1000.	De response bevat een message met de tekst: "request is not valid, there is no feedback created."		

Id	SYN.T21-3				
Doel van de test	Verifieer dat er geen nieuw incident is aangemaakt.				
Preconditie	Het incident is nog niet aanwezig in het systeem.				
Postconditie	Het incident is niet toegevoegd aan het systeem.				
Verwijzing testboom	Incident-Toevoegen				
Testactie(s)					
	Testactie	Testactie	Verwachte reactie	Conclusie	Opmerking

	Id				
	1	De call POST “/incident” wordt uitgevoerd met de body: “NIET VALIDE” en userId 1000.	De response bevat een message met de tekst: "request is not valid, there is no feedback created."		

Incident-Update

Id	SYN.T22-1				
Doel van de test	Verifieer dat het incident met valide incidentEncId is aangepast.				
Preconditie	Er is een feedback aanwezig voor het incidentEncId.				
Postconditie	De feedback is aangepast.				
Verwijzing testboom	Incident-Update				
Testactie(s)	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call PUT “/incident” wordt uitgevoerd met de body: “XXXXXX” en userId 1000.	Het incident is aangepast en het incidentEncId wordt teruggegeven.		

Id	SYN.T22-1				
Doel van de test	Verifieer dat het incident met valide incidentEncId is aangepast.				
Preconditie	Er is een feedback aanwezig voor het incidentEncId.				
Postconditie	De feedback is aangepast.				
Verwijzing testboom	Incident-Update				

Testactie(s)					
	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call PUT “/incident” wordt uitgevoerd met de body: “NULL” en userId 1000.	Het incident is niet aangepast en het incidentEncId wordt teruggegeven.		

Id	SYN.T22-3				
Doel van de test	Verifieer dat het incident met valide incidentEncId niet is aangepast.				
Preconditie	Er is een feedback aanwezig voor het incidentEncId.				
Postconditie	De feedback is niet aangepast.				
Verwijzing testboom	Incident-Updates				
Testactie(s)					
	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call PUT “/incident” wordt uitgevoerd met de body: “NIET VALIDE” en userId 1000.	Het incident is niet aangepast en het incidentEncId wordt teruggegeven.		

Algoritme Test : Statement Covarage

Gebruiker-Regenerate-Keys

Id	AT.T01-1
Doel van de test	Succesvol genereren van nieuwe consumerKey en -Secret.

Preconditie	Er is een valide gebruiker aanwezig.				
Postconditie	De gebruiker heeft een nieuwe consumerKey en -Secret.				
Paden	1-3-4-5-6-8				
Verwijzing testboom	Gebruiker-Regenerate-Keys				
Testactie(s)					
	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	regenerateKeys wordt uitgevoerd met een valide user en als userId 1000.	Een response met de nieuwe waarden voor deze gebruiker.		

Id	AT.T01-2				
Doel van de test	De validatie van de consumerKey geeft een foutmelding, omdat deze niet valide is.				
Preconditie	Er is een valide gebruiker aanwezig.				
Postconditie	Het systeem geeft aan dat de consumerKey niet valide is.				
Paden	2-10				
Verwijzing testboom	Gebruiker-Regenerate-Keys				
Testactie(s)					
	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	regenerateKeys wordt uitgevoerd met een niet valide user en als userId 1000. De consumerKey is "<35CHAR"	De response bevat een message met de tekst: "No valid userkey supplied!"		

Id	AT.T01-3				
Doel van de test	Update geeft een foutmelding, omdat de update in de database verkeerd gaat.				

Preconditie	Er is een valide gebruiker aanwezig.				
Postconditie	De gebruiker heeft geen nieuwe consumerKey en -Secret.				
Paden	1-3-4-5-7-9				
Verwijzing testboom	Gebruiker-Regenerate-Keys				
Testactie(s)	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	regenerateKeys wordt uitgevoerd met een valide gebruiker en als userId 1000.	De response bevat een message met de tekst: "Could not create new key and secret!"		

Gebruiker-Opvragen

Id	AT.T02-1				
Doel van de test	Succesvol verkrijgen van gebruiker.				
Preconditie	Er is een valide gebruiker aanwezig in het systeem.				
Postconditie	De response bevat de gegevens van de opgevraagde gebruiker				
Paden	1-3-4				
Verwijzing testboom	Gebruiker-Opvragen				
Testactie(s)	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	getUser wordt uitgevoerd met een valide gebruiker en als userId 1000.	De response bevat de gegevens van de opgevraagde gebruiker.		

Id	AT.T02-2
----	----------

Doel van de test	Validate consumerKey geeft een foutmelding, omdat deze niet valide is.				
Preconditie	Er is een valide gebruiker aanwezig in het systeem.				
Postconditie	De response geeft een foutmelding.				
Paden	2-5				
Verwijzing testboom	Gebruiker-Opvragen				
Testactie(s)					
	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	getUser wordt uitgevoerd met een niet valide gebruiker en als userId 1000. Als consumerKey wordt "<35CHAR" meegegeven.	De response bevat een message met de tekst: "No valid userkey supplied!"		

Operator-Opvragen

Id	AT.T03-1				
Doel van de test	Succesvol verkrijgen van een Operator				
Preconditie	Er is een operator aanwezig in het systeem.				
Postconditie	De response bevat de gegevens van de opgevraagde operator.				
Paden	1-3-4				
Verwijzing testboom	Operator-Opvragen				
Testactie(s)					
	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	getOperat	De gegevens van de opgevraagde		

	<p>or wordt aangeroe en met als operatorE ncId "09889- abd7- 7882- abd7- 7882- abd7-7882 " en userId 1000.</p>	<p>operator worden teruggegeven. operatorDao2.setExternalId(2);</p> <p>operatorDao2.setFirstname("Operator 2");</p> <p>operatorDao2.setLastname("Lastname 2");</p> <p>operatorDao2.setEncId("09889-abd7-7882-abd7-7882-abd7-7882");</p> <p>operatorDao2.setEmail("operator2.nl");</p> <p>operatorDao2.setUserId(1200);</p> <p>operatorDao2.setActive(0);</p>		
--	--	--	--	--

Id	AT.T03-2				
Doel van de test	De validatie van het operatorEncId geeft een foutmelding, omdat deze niet valide is.				
Preconditie	Er is een operator aanwezig in het systeem.				
Postconditie	De response bevat een foutmelding				
Paden	2-5				
Verwijzing testboom	Operator-Opvragen				
Testactie(s)	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	getOperator wordt aangeroepen met als operatorEncId "" en userId 1000.	De response bevat een message met de tekst: "OperatorEncId is not valid!"		

Incident-Customer-Satisfaction-Timeframe

Id	AT.T04-1				
Doel van de test	Succesvol verkrijgen van Customer Satisfaction per TimeFrame				
Preconditie	Er zijn customer satisfaction aanwezig in het systeem.				
Postconditie	De opgevraagde customer satisfactions worden teruggegeven.				
Paden	1-4-5-6				
Verwijzing testboom	Incident-Customer-Satisfaction-Timeframe				
Testactie(s)	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	getCustomerSatisfactionByTimeFrame wordt aangeroepen met het TimeFrame DAY en als userId 1000.	De response bevat de customer satisfaction voor de dag en de dag ervoor.		

Id	AT.T04-2				
Doel van de test	De waarde van het opgegeven TimeFrame wordt omgezet naar DAY en hiervan wordt de Customer Satisfaction opgehaald.				
Preconditie	Er zijn customer satisfaction aanwezig in het systeem.				
Postconditie	De customer satisfactions voor de dag worden teruggegeven.				
Paden	2-3-4-5-6				
Verwijzing testboom	Incident-Customer-Satisfaction-Timeframe				
Testactie(

s)	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	getCustomerSatisfactionByTimeFrame wordt aangeroepen met het TimeFrame dag en als userId 1000.	Het TimeFrame wordt omgezet naar DAY. De response bevat de customer satisfaction voor de dag en de dag ervoor.		

Incident-Customer-Satisfaction-Relative

Id	AT.T05-1				
Doel van de test	Succesvol ophalen van Customer Satisfaction. Per opgegeven kwartaal en hetzelfde kwartaal van het voorgaande jaar.				
Preconditie	Er zijn customer satisfaction aanwezig voor het opgevraagde TimeFrame.				
Postconditie	De customer satisfaction voor de huidige maand en dezelfde maand het jaar ervoor worden teruggegeven.				
Paden	1-3-6-8-9-11-14-17				
Verwijzing testboom	Incident-Customer-Satisfaction-Relative				
Testactie(s)	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	getCustomerSatisfactionRelativeTo wordt aangeroepen met Q1 als TimeFrame en userId 1000.	De response bevat de customer satisfaction van Q1 van het huidige		

			jaar en het jaar ervoor.		
--	--	--	--------------------------------	--	--

Id	AT.T05-2				
Doel van de test	De waarde van het opgegeven TimeFrame wordt omgezet naar MONTH. Vervolgens wordt succesvol de Customer Satisfaction opgehaald. Per huidige maand en dezelfde maand van het voorgaande jaar.				
Preconditie	Er zijn customer satisfaction aanwezig voor het opgevraagde TimeFrame.				
Postconditie	De customer satisfaction voor de huidige maand en dezelfde maand het jaar ervoor worden teruggegeven.				
Paden	2-4-3-5-7-8-9-10-13 -16				
Verwijzing testboom	Incident-Customer-Satisfaction-Relative				
Testactie(s)	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	getCustomerSatisfactionRelativeTo wordt aangeroepen met Q1-Q2 als TimeFrame en userId 1000.	Het TimeFrame wordt omgezet naar MONTH. De response bevat de customer satisfaction van de huidige maand van het huidige jaar en het jaar ervoor.		

Id	AT.T05-3				
----	----------	--	--	--	--

Doel van de test	Succesvol ophalen van Customer Satisfaction. Per huidige maand en dezelfde maand van het voorgaande jaar.				
Preconditie	Er zijn customer satisfaction aanwezig voor het opgevraagde TimeFrame.				
Postconditie	De customer satisfaction voor de huidige maand en dezelfde maand het jaar ervoor worden teruggegeven.				
Paden	1-3-5-7-8-9-10-13-16				
Verwijzing testboom	Incident-Customer-Satisfaction-Relative				
Testactie(s)	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	getCustomerSatisfactionRelativeTo wordt aangeroepen met DAY als TimeFrame en userId 1000.	Het TimeFrame wordt omgezet naar MONTH. De response bevat de customer satisfaction van de huidige maand van het huidige jaar en het jaar ervoor.		

Id	AT.T05-4
Doel van de test	Succesvol ophalen van Customer Satisfaction. Per huidig jaar en het voorgaande jaar.
Preconditie	Er zijn customer satisfaction aanwezig voor het opgevraagde TimeFrame.
Postconditie	De customer satisfaction voor het huidige jaar en het jaar ervoor worden teruggegeven.

Paden	1-3-6-8-9-12-15-18				
Verwijzing testboom	Incident-Customer-Satisfaction-Relative				
Testactie(s)	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	getCustomerSatisfactionRelativeTo wordt aangeroepen met YEAR als TimeFrame en userId 1000.	De response bevat de customer satisfaction van het huidige jaar en het jaar ervoor.		

Incident-Operator-Scores-Timeframe

Id	AT.T06-1				
Doel van de test	Succesvol ophalen van Operator Scores voor het opgegeven TimeFrame.				
Preconditie	Er zijn operator scores aanwezig in het systeem.				
Postconditie	De operator scores van het opgevraagde TimeFrame worden teruggegeven.				
Paden	1-4-5-6-7				
Verwijzing testboom	Incident-Operator-Scores-Timeframe				
Testactie(s)	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	getOperatorScoresByTimeFrame wordt uitgevoerd met DAY als TimeFrame en 1000 als userId.	De operator scores voor de huidige dag en de dag ervoor worden teruggegeven		

			n.		
--	--	--	----	--	--

Id	AT.T06-2				
Doel van de test	Omzetten van een niet valide TimeFrame en daarna succesvol ophalen van Operator Scores voor het TimeFrame DAY.				
Preconditie	Er zijn operator scores aanwezig in het systeem.				
Postconditie	De operator scores van TimeFrame DAY worden teruggegeven.				
Paden	2-3-4-5-6-7				
Verwijzing testboom	Incident-Operator-Scores-Timeframe				
Testactie(s)	Testacti e Id	Testactie	Verwachte reactie	Conclusi e	Opmerkin g
	1	getOperatorScoresByTimeFrame wordt uitgevoerd met maand als TimeFrame en 1000 als userId.	Het TimeFrame wordt omgezet naar DAY. De operator scores voor de huidige dag en de dag ervoor worden teruggegeven.		

Incident-Operator-Scores-Relative

Id	AT.T07-1
Doel van de test	Succesvol ophalen van Operator Scores. Per huidig kwartaal en hetzelfde kwartaal van het voorgaande jaar.
Preconditie	Er zijn operator scores aanwezig in het systeem.

Postconditie	De operator scores voor de opgevraagde periode en dezelfde periode het jaar ervoor worden teruggegeven.				
Paden	1-4-6-8-9-11-14-16				
Verwijzing testboom	Incident-Operator-Scores-Relative				
Testactie(s)	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call "/incident/stats/operator-scores/relative-to " wordt uitgevoerd met als TimeFrame Q2 en als userId 1000.	De operator scores voor Q2 van het huidige jaar en het jaar ervoor worden opgehaald.		

Id	AT.T07-2				
Doel van de test	De waarde van het opgegeven TimeFrame wordt omgezet naar MONTH. Vervolgens wordt succesvol de Operator Scores opgehaald. Per huidige maand en dezelfde maand van het voorgaande jaar.				
Preconditie	Er zijn operator scores aanwezig in het systeem.				
Postconditie	De operator scores voor de maand en dezelfde maand het jaar ervoor worden teruggegeven.				
Paden	2-3-4-5-7-8-9-10-13-17				
Verwijzing testboom	Incident-Operator-Scores-Relative				
Testactie(s)	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call "/incident/stats/operator-scores/relative-to " wordt uitgevoerd met als TimeFrame Q5 en als userId 1000.	Het TimeFrame wordt omgezet naar MONTH. De operator scores voor		

			de huidige maand van het huidige jaar en het jaar ervoor worden opgehaald.		
--	--	--	--	--	--

Id	AT.T07-3				
Doel van de test	De waarde van het opgegeven TimeFrame wordt omgezet naar MONTH. Vervolgens wordt succesvol de Operator Scores opgehaald. Per huidige maand en dezelfde maand van het voorgaande jaar.				
Preconditie	Er zijn operator scores aanwezig in het systeem.				
Postconditie	De operator scores voor de maand en dezelfde maand het jaar ervoor worden teruggegeven.				
Paden	1-4-5-7-8-9-10-13-17				
Verwijzing testboom	Incident-Operator-Scores-Relative				
Testactie(s)	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call "/incident/stats/operator-scores/relative-to " wordt uitgevoerd met als TimeFrame DAY en als userId 1000.	Het TimeFrame wordt omgezet naar MONTH. De operator scores voor de huidige maand van het huidige jaar en het jaar ervoor worden opgehaald.		

Id	AT.T07-4				
Doel van de	Succesvol ophalen van Operator Scores. Per huidig jaar en het voorgaande jaar.				

test					
Preconditie	Er zijn operator scores aanwezig in het systeem.				
Postconditie	De operator scores voor de opgevraagde periode en dezelfde periode het jaar ervoor worden teruggegeven.				
Paden	1-4-6-8-9-12-15-18				
Verwijzing testboom	Incident-Operator-Scores-Relative				
Testactie(s)	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call “/incident/stats/operator-scores/relative-to” wordt uitgevoerd met als TimeFrame YEAR en als userId 1000.	De operator scores van het huidige jaar en het jaar ervoor worden opgehaald.		

Algoritme Test : Branche Coverage

Incident-Opvragen

Id	AT.T08-1				
Doel van de test	Succesvol ophalen van Feedback voor het opgegeven valide feedbackEncId				
Preconditie	Er is een feedback aanwezig voor het valide feedbackEncId.				
Postconditie	De gegevens van de feedback worden teruggegeven.				
Paden	1-3-5				
Verwijzing testboom	Incident-Opvragen				
Testactie(s)	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call “incident/{feedbackEncId}”	De gegevens van de opgevraagde		

		“ wordt aangeroepen met als feedbackEncId “XXXXXX” en als userId 1000.	feedback worden teruggegeven		
--	--	--	------------------------------	--	--

Id	AT.T08-2				
Doel van de test	De validatie van het FeedbackEncId geeft een error, omdat het opgegeven feedbackEncId niet valide is.				
Preconditie	Er is een feedback aanwezig voor het valide feedbackEncId.				
Postconditie	Er wordt een foutmelding teruggegeven.				
Paden	2-4				
Verwijzing testboom	Incident-Opvragen				
Testactie(s)	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call “incident/{feedbackEncId}” wordt aangeroepen met als feedbackEncId “<CHAR35” en als userId 1000.	De response bevat een message met de tekst: "No valid encId supplied!"		

Incident-Versturen

Id	AT.T09-1				
Doel van de test	het succesvol verzenden van een feedback mail voor het opgegeven feedbackEncId.				
Preconditie	Er is een feedback aanwezig voor het valide feedbackEncId.				
Postconditie	De feedback mail is verzonden aan de beoordelaar en een succes message wordt teruggegeven.				
Paden	1-3-6				

Verwijzing testboom	Incident-Versturen				
Testactie(s)					
	Testacti e Id	Testactie	Verwachte reactie	Conclusi e	Opmerkin g
	1	De call “incident/{feedbackEncId}/send “ wordt aangeroepen met als feedbackEncId “XXXXXX” en als userId 1000.	De response bevat een message met de tekst: "Feedback mail send for encId: " + XXXXXX + "." En de feedback mail is verzonden aan de beoordelaar.		

Id	AT.T09-2				
Doel van de test	De validatie van het FeedbackEncId geeft een error, omdat het opgegeven feedbackEncId niet valide is.				
Preconditie	Er is een feedback aanwezig voor het valide feedbackEncId.				
Postconditie	De feedback mail is niet verzonden aan de beoordelaar en een fout message wordt teruggegeven.				
Paden	2-4				
Verwijzing testboom	Incident-Versturen				
Testactie(s)					
	Testacti e Id	Testactie	Verwachte reactie	Conclusi e	Opmerkin g
	1	De call “incident/{feedbackEncId}/send	De response		

		nd “ wordt aangeroepen met als feedbackEncId “<35CHAR” en als userId 1000.	bevat een message met de tekst: "No valid encId supplied!" En de feedback mail is niet verzonden aan de beoordelaar.		
--	--	---	---	--	--

Id	AT.T09-3				
Doel van de test	SendFeedbackMail geeft FALSE, omdat de feedback mail niet verzonden kon worden.				
Preconditie	Er is een feedback aanwezig voor het valide feedbackEncId.				
Postconditie	De feedback mail is niet verzonden aan de beoordelaar en een fout message wordt teruggegeven.				
Paden	1-5-7				
Verwijzing testboom	Incident-Versturen				
Testactie(s)					
	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call “incident/{feedbackEncId}/send “ wordt aangeroepen met als feedbackEncId “XXXXXX” en als userId 1000.	De response bevat een message met de tekst: "Feedback mail not send for encId: " + feedbackEncId + "!" En de feedback mail is verzonden aan de		

			beoordelaar.		
--	--	--	--------------	--	--

Gebruiker-Toevoegen

Id	AT.T10-1				
Doel van de test	Het succesvol toevoegen van een User.				
Preconditie	De gebruiker is nog niet aanwezig in het systeem.				
Postconditie	De gebruiker is toegevoegd aan het systeem.				
Paden	1-3-5-7-10				
Verwijzing testboom	Gebruiker-Toevoegen				
Testactie(s)	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call POST “/accesstoken” wordt uitgevoerd met de body: “XXXXXX”	De gebruiker is toegevoegd. De response bevat de gegevens van de nieuwe gebruiker.		

Id	AT.T10-2				
Doel van de test	De validatie van de User geeft een foutmelding, omdat de User niet valide is.				
Preconditie	De gebruiker is nog niet aanwezig in het systeem.				
Postconditie	De gebruiker is niet toegevoegd aan het systeem.				
Paden	2-4				

Verwijzing testboom	Gebruiker-Toevoegen				
Testactie(s)					
	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call POST “/accesstoken” wordt uitgevoerd met de body: “XXXXXX”	De gebruiker is niet toegevoegd. De response bevat een message met de tekst: "User not valid!"		

Id	AT.T10-3				
Doel van de test	Er wordt een foutmelding gegeven, omdat de gebruiker reeds bestaat.				
Preconditie	De gebruiker is reeds aanwezig in het systeem.				
Postconditie	De gebruiker is niet toegevoegd aan het systeem. De response geeft aan dat de gebruiker al bestaat.				
Paden	1-3-6-8				
Verwijzing testboom	Gebruiker-Toevoegen				
Testactie(s)					
	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call POST “/accesstoken” wordt uitgevoerd met de body: “XXXXXX”	De gebruiker is niet toegevoegd. De response bevat een message met de tekst: "User already exists!"		

Id	AT.T10-4				
Doel van de test	CreateUser geeft een exception, omdat het toevoegen niet lukt.				

Preconditie	De gebruiker is nog niet aanwezig in het systeem.				
Postconditie	De gebruiker is niet toegevoegd aan het systeem. Hiervan wordt melding gegeven in de response.				
Paden	1-3-5-9-11				
Verwijzing testboom	Gebruiker-Toevoegen				
Testactie(s)					
	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call POST “/accesstoken” wordt uitgevoerd met de body: “XXXXXX”	De gebruiker is niet toegevoegd. De response bevat een message met de tekst: "Could not create new user with accountname: " + user.getAccountName() + "!"		

Operator-Update

Id	AT.T11-1				
Doel van de test	Succesvol updaten van een Operator.				
Preconditie	De operator is reeds aanwezig in het systeem.				
Postconditie	De operator is aangepast en is niet meer gelijk aan de operator voor de test.				
Paden	1-2-4-5-9-10-11-12				
Verwijzing testboom	Operator-Update				
Testactie(s)					
	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call PUT	De operator is		

		“/operator” wordt uitgevoerd met als body: “dao1 + “nieuwe voornaam 2” en als userId 1000.	aangepast. De response bevat de gegevens van de operator.		
--	--	--	---	--	--

Id	AT.T11-2				
Doel van de test	De validatie van de Operator geeft een foutmelding, omdat de Operator niet valide is.				
Preconditie	De operator is reeds aanwezig in het systeem.				
Postconditie	De operator is niet aangepast. Hiervan wordt in de response een melding gemaakt.				
Paden	1-2-3-7				
Verwijzing testboom	Operator-Update				
Testactie(s)	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call PUT “/operator” wordt uitgevoerd met als body: “dao1 + lege achternaam” en als userId 1000.	De operator is niet aangepast. De response bevat een message met de tekst: "Operator not valid!"		

Id	AT.T11-3				
Doel van de test	Er wordt een foutmelding gegeven omdat de Operator niet bestaat.				
Preconditie	De operator is niet aanwezig in het systeem.				
Postconditie	De operator is niet aangepast. Hiervan wordt een melding gemaakt in de response.				
Paden	1-2-4-5-6-8				
Verwijzing	Operator-Update				

testboom					
Testactie(s)					
	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call PUT “/operator” wordt uitgevoerd met als body: “dao1 + 09889-abd7-7882-abd7-3209-09df-3209” en als userId 1000.	De operator is niet aangepast. De response bevat een message met de tekst: "Operator not found!"		

Operator-Toevoegen

Id	AT.T12-1				
Doel van de test	Het toevoegen van een bestaande Operator welke niet actief is, wordt weer actief gemaakt.				
Preconditie	De operator is aanwezig, maar heeft de status active=False.				
Postconditie	De operator heeft de status active=True				
Paden	1-2-4-6-7-9-13-16-17				
Verwijzing testboom	Operator-Toevoegen				
Testactie(s)					
	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call POST “/operator” wordt uitgevoerd met de body: DAO1 en active = false ” en als userId 1000.	De status van de operator is active=true. De response bevat een message met de tekst: "Operator already created!" The operator status is set to active, it was inactive."		

Id	AT.T12-2
----	----------

Doel van de test	De validatie van de Operator geeft een foutmelding, omdat de Operator niet valide is.				
Preconditie	De operator is niet aanwezig.				
Postconditie	De operator is niet toegevoegd. In de response wordt hier melding van gegeven.				
Paden	1-2-3-5				
Verwijzing testboom	Operator-Toevoegen				
Testactie (s)					
	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call POST “/operator” wordt uitgevoerd met de body: <pre> operatorDao1.setExternalId(1); operatorDao1.setLastname("Lastname"); operatorDao1.setEncId("345-0-09df-3209-09df-3209-09df-3209"); operatorDao1.setEmail("operator@tor.nl"); operatorDao1.setUserId(1100); operatorDao1.setActive(1); </pre> ” en als userId 1000.	De operator is niet toegevoegd. De response bevat een message met de tekst: "Could not add new operator!"		

Id	AT.T12-3
Doel van de test	Het toevoegen van een bestaande Operator welke ook actief is.
Preconditie	De operator is aanwezig en heeft de status active=True.
Postconditie	Er wordt een melding teruggegeven dat de operator al bestaat.
Paden	1-2-4-6-7-10-17

Verwijzing testboom	Operator-Toevoegen				
Testactie(s)					
	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call POST “/operator” wordt uitgevoerd met de body: “XXXXXX” en als userId 1000.	De response bevat een message met de tekst: "Operator already created!"		

Id	AT.T12-4				
Doel van de test	Het succesvol toevoegen van een nieuwe Operator.				
Preconditie	De operator is niet aanwezig.				
Postconditie	De operator is toegevoegd.				
Paden	1-2-4-6-8-12-15				
Verwijzing testboom	Operator-Toevoegen				
Testactie(s)					
	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call POST “/operator” wordt uitgevoerd met de body: “XXXXXX” en als userId 1000.	De nieuwe operator is toegevoegd aan het systeem. De gegevens van de operator worden teruggegeven in de response.		

Id	AT.T12-5				
Doel van de test	CreateOperator geeft een exception, omdat het toevoegen niet is gelukt.				
Preconditie	De operator is niet aanwezig.				
Postconditi	De operator is niet toegevoegd. Er wordt hiervan een melding gegeven.				

e					
Paden	1-2-4-6-8-11-14				
Verwijzing testboom	Operator-Toevoegen				
Testactie(s)	Testacti e Id	Testactie	Verwachte reactie	Conclusi e	Opmerkin g
	1	De call POST "/operator" wordt uitgevoerd met de body: "XXXXXX" en als userId 1000.	De operator is niet toegevoegd. De response bevat een message met de tekst: "Could not create operator for userId: " + operator.getUserId().toString() + "!"		

Incident-Toevoegen = Incident-Update

Id	AT.T13-1				
Doel van de test	Het succesvol toevoegen van een nieuw incident.				
Preconditie	De feedback is niet aanwezig in het systeem.				
Postconditie	De feedback is toegevoegd aan het systeem. Het feedbackEncId wordt terug gegeven.				
Paden	1-2-4-7-9-10				
Verwijzing testboom	Incident-Toevoegen = Incident-Update				
Testactie(s)	Testactie	Testactie	Verwachte reactie	Conclusie	Opmerking

	Id				
	1	De call POST “/incident” wordt aangeroepen met de body: “XXXXXX” en userId 1000.	De feedback is toegevoegd aan het systeem. Het feedbackEncId wordt teruggegeven in de response.		

Id	AT.T13-2				
Doel van de test	De validatie van het Incident geeft een foutmelding, omdat het incident niet valide is.				
Preconditie	De feedback is niet aanwezig in het systeem.				
Postconditie	De feedback is niet toegevoegd. Hiervan wordt een melding gegeven.				
Paden	1-2-3-5				
Verwijzing testboom	Incident-Toevoegen = Incident-Update				
Testactie(s)	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call POST “/incident” wordt aangeroepen met de body: “XXXXXX” en userId 1000.	De feedback is niet toegevoegd aan het systeem. De response bevat een message met de tekst: "Request is not valid!"		

Id	AT.T13-3				
Doel van de test	ProcesIncident geeft een exception, omdat het verwerken van het incident niet is gelukt.				
Preconditie	De feedback is niet aanwezig in het systeem.				
Postconditie	De feedback is niet toegevoegd aan het systeem. Hiervan wordt een melding gegeven.				
Paden	1-2-4-6-8				

Verwijzing testboom	Incident-Toevoegen = Incident-Update				
Testactie(s)					
	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call POST “/incident” wordt aangeroepen met de body: “XXXXXX” en userId 1000.	De feedback is niet toegevoegd aan het systeem. De response bevat een message met de tekst: "Could not process incident!"		

Beslissingstabellen Test

Incident-Toevoegen = Incident-Update

Id	CE.T01-1				
Doel van de test	Het request is niet valide, omdat het incident niet valide is.				
Preconditie	Er is geen feedback aanwezig in het systeem.				
Postconditie	er is geen feedback toegevoegd, omdat het incident niet valide is.				
Verwijzing testboom	Incident-Toevoegen = Incident-Update				
Testactie(s)					
	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call POST “/incident” wordt aangeroepen met de body: “XXXXXX” en userId 1000.	De body is niet valide, er wordt een message teruggegeven met de melding: "Request is not valid!"		

Id	CE.T01-2				
Doel van de	Het incident wordt verwerkt tot een feedback in de database.				

test					
Preconditie	Er is geen feedback aanwezig in het systeem.				
Postconditie	Er is een feedback toegevoegd. Het feedbackEncId wordt teruggegeven in de response.				
Verwijzing testboom	Incident-Toevoegen = Incident-Updaten				
Testactie(s)					
	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call POST “/incident” wordt aangeroepen met de body: “XXXXXX” en userId 1000.	De reponse bevat een message met het feedbackEncId.		

Id	CE.T01-4				
Doel van de test	Er treed een exception op doordat het incident niet verwerkt kan worden.				
Preconditie	Er is geen feedback aanwezig in het systeem.				
Postconditie	er is geen feedback toegevoegd, omdat het incident niet verwerkt kon worden.				
Verwijzing testboom	Incident-Toevoegen = Incident-Updaten				
Testactie(s)					
	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call POST “/incident” wordt aangeroepen met de body: “XXXXXX” en userId 1000.	Tijdens het verwerken van het incident gaat iets mis. Er wordt een exception gegeven. De response heeft een message met de melding: "Could not process incident!"		

Bijlage X: Rest-API documentatie

POST {{url}}accesstoken/

body:

```
{
  "accountName": "Test Bedrijf A.P.I"
}
```

response:

```
{
  "consumerKey": "VBXpDZPSakFQswTWG8xdwtjWwCdZ9ZUr",
  "consumerSecret": "jvT25xxfGWlr4by3NRmb75kWvypaZtPw",
  "accountName": "Test Bedrijf A.P.I",
  "role": "DASHBOARD"
}
```

GET {{url}}accesstoken/

header x-user-key

response:

```
{
  "consumerKey": "thHNsVStEyQDqj5Zh6feRYutfhwgNcgZ",
  "consumerSecret": "5x2debhYGV6WMX6bDuRz6sRmWtbyXYLW",
  "accountName": "Mailplus Dashboard",
  "role": "SUPERUSER"
}
```

POST {{url}}accesstoken/regenerate-keys

body:

```
{
  "consumerKey": "6c92b35a7f13c383ef7c32a147439927"
}
```

response:

```
{
  "consumerKey": "ckfNSymwisfHUSwV6mkfk4ipXmCh2DLZ",
  "consumerSecret": "cp9tkqcLjDhngisrl8htpt7hKUKHryPD",
  "accountName": "Test Bedrijf A.P.I",
  "role": "DASHBOARD"
}
```

POST {{url}}operator

body:

```
{
  "externalId": 3,
```

```
    "firstname": "Jos",
    "lastname": "Verhaar",
    "email": "josverhaar@gmail.com"
}
```

response:

```
{
  "externalId": "5",
  "encId": "263c7508-35e5-4a38-b72a-b3fc6936158b",
  "firstname": "Jos",
  "lastname": "Verhaar",
  "email": "josverhaar@gmail.com",
  "active": true
}
```

PUT {{url}}operator

body:

```
{
  "externalId": 3,
  "firstname": "Jos",
  "lastname": "Verhaar",
  "email": "jos@mailplus.nl"
}
```

response:

```
{
  "externalId": "5",
  "encId": "263c7508-35e5-4a38-b72a-b3fc6936158b",
  "firstname": "Jos",
  "lastname": "Verhaar",
  "email": "jos@mailplus.nl",
  "active": true
}
```

GET {{url}}operator/{{operatorId}}

response:

```
{
  "externalId": "2",
  "encId": "61987163-ca2f-4897-8cb3-e786701b41f",
  "firstname": "Jos",
  "lastname": "Verhaar",
  "email": "jos@mailplus.nl",
  "active": true
}
```

DELETE {{url}}operator/{{operatorId}}

response:

```
{
  "externalId": "2",
  "encId": "61987163-ca2f-4897-8cb3-e786701b41f",
  "firstname": "Jos",
  "lastname": "Verhaar",
  "email": "jos@mailplus.nl",
  "active": false
}
```

GET {{url}}operator/list

response:

```
[
  {
    "externalId": "2",
    "encId": "61987163-ca2f-4897-8cb3-e786701b41f",
    "firstname": "Jos",
    "lastname": "Verhaar",
    "email": "jos@mailplus.nl",
    "active": false
  },
  {
    "externalId": "5",
    "encId": "263c7508-35e5-4a38-b72a-b3fc6936158b",
    "firstname": "Jos",
    "lastname": "Verhaar",
    "email": "jos@mailplus.nl",
    "active": true
  },
  {
    "externalId": "1",
    "encId": "e54beacc-09be-4bc8-bd41-a7b255bbc22",
    "firstname": "Paul",
    "lastname": "Nederlof",
    "email": "paulne@mailplus.nl",
    "active": true
  },
  {
    "externalId": "3",
    "encId": "78222c7a-0c7d-47f8-8f1c-d3e3cfb8dd1",
    "firstname": "Ralph",
    "lastname": "van Delft",
    "email": "ralph@mailplus.nl",
    "active": true
  }
]
```

GET {{url}}operator/stats/operator-score/{{operatorId}}?timeframe=day

GET {{url}}operator/stats/operator-score/{{operatorId}}?timeframe=month

GET {{url}}operator/stats/operator-score/{{operatorId}}?timeframe=q1

```
GET {{url}}operator/stats/operator-score/{{operatorId}}?timeframe=q2
GET {{url}}operator/stats/operator-score/{{operatorId}}?timeframe=q3
GET {{url}}operator/stats/operator-score/{{operatorId}}?timeframe=q4
GET {{url}}operator/stats/operator-score/{{operatorId}}?timeframe=year
```

response:

```
{
  "operatorScore": [
    {
      "good": {
        "count": 1,
        "percentage": 100
      },
      "ok": {
        "count": 0,
        "percentage": 0
      },
      "bad": {
        "count": 0,
        "percentage": 0
      },
      "operatorFirstname": "Jos"
    }
  ]
}
```

POST {{url}}incident

body:

```
{
  "id": "I2016 007",
  "shortDescription": "Feedback verzoek via Rest API call.",
  "requestor": {
    "firstname": "Rest API",
    "lastname": "feedback/create",
    "account": "Postman",
    "email": "josverhaar@gmail.com"
  },
  "operator": {
    "firstname": "Jos",
    "lastname": "Verhaar",
    "externalId": 2
  }
}
```

Response:

```
{
```

```
"encId": "30408d8c-d48b-40d7-be5c-6a3af9b3736b"
}
```

PUT {{url}}incident

body:

```
{
  "id": "I2016 1107",
  "shortDescription": "Feedback verzoek via Rest API call.",
  "requestor": {
    "firstname": "Rest API",
    "lastname": "feedback/create",
    "account": "Postman",
    "email": "josverhaar@gmail.com"
  },
  "operator": {
    "firstname": "Jos",
    "lastname": "Verhaar",
    "externalId": 2
  }
}
```

response:

```
{
  "encId": "30408d8c-d48b-40d7-be5c-6a3af9b3736b"
}
```

GET {{url}}incident/{{feedbackEncId}}

response:

```
{
  "encId": "qwGF2nNWLS8qku6KZCJDJb59KLngAjM",
  "ticketNumber": "I2016 011",
  "shortDescription": "Short description 11",
  "operator": {
    "lastName": "Nederlof",
    "encId": "e54beacc-09be-4bc8-bd41-a7b255bbc22",
    "firstName": "Paul"
  },
  "requestor": {
    "lastName": "API 11",
    "email": "paulne+11@mailplus.nl",
    "account": "Postman 11",
    "firstName": "Rest"
  },
  "feedbackLinks": {
    "OK": {
```



```

    "url":
    "http://localhost:8080/supportfeedback/feedback/qwGF2nNWLS8qku6KZCJDJjb59KLNgAjM/2"
  },
  "GOOD": {
    "url":
    "http://localhost:8080/supportfeedback/feedback/qwGF2nNWLS8qku6KZCJDJjb59KLNgAjM/1"
  },
  "BAD": {
    "url":
    "http://localhost:8080/supportfeedback/feedback/qwGF2nNWLS8qku6KZCJDJjb59KLNgAjM/3"
  }
}
}
}

```

POST {{url}}incident/{{feedbackEncId}}/send

response:

```

{
  "message": "Feedback mail send for encId: qwGF2nNWLS8qku6KZCJDJjb59KLNgAjM."
}

```

```

GET {{url}}incident/stats/customer-satisfaction?timeframe=day
GET {{url}}incident/stats/customer-satisfaction?timeframe=month
GET {{url}}incident/stats/customer-satisfaction?timeframe=q1
GET {{url}}incident/stats/customer-satisfaction?timeframe=q2
GET {{url}}incident/stats/customer-satisfaction?timeframe=q3
GET {{url}}incident/stats/customer-satisfaction?timeframe=q4
GET {{url}}incident/stats/customer-satisfaction?timeframe=year

```

response:

```

{
  "current": {
    "customerSatisfaction": {
      "requestCount": 0,
      "responseCount": 0
    }
  },
  "previous": {
    "customerSatisfaction": {
      "good": {
        "count": 1,
        "percentage": 25
      },
      "ok": {
        "count": 2,
        "percentage": 50
      }
    }
  }
}

```

```

    },
    "bad": {
      "count": 1,
      "percentage": 25
    },
    "requestCount": 4,
    "responseCount": 4
  }
}
}

```

GET {{url}}incident/stats/customer-satisfaction/count?count=100

response:

```

{
  "current": {
    "customerSatisfaction": {
      "good": {
        "count": 2,
        "percentage": 66.667
      },
      "ok": {
        "count": 1,
        "percentage": 33.333
      },
      "bad": {
        "count": 0,
        "percentage": 0
      }
    }
  },
  "previous": {
    "customerSatisfaction": {
      "good": {
        "count": 2,
        "percentage": 66.667
      },
      "ok": {
        "count": 1,
        "percentage": 33.333
      },
      "bad": {
        "count": 0,
        "percentage": 0
      }
    }
  }
}
}

```

```

GET {{url}}incident/stats/operator-scores?timeframe=day
GET {{url}}incident/stats/operator-scores?timeframe=month
GET {{url}}incident/stats/operator-scores?timeframe=q1
GET {{url}}incident/stats/operator-scores?timeframe=q2
GET {{url}}incident/stats/operator-scores?timeframe=q3
GET {{url}}incident/stats/operator-scores?timeframe=q4
GET {{url}}incident/stats/operator-scores?timeframe=year

```

response:

```

{
  "current": {},
  "previous": {
    "operatorScores": [
      {
        "good": {
          "count": 1,
          "percentage": 50
        },
        "ok": {
          "count": 1,
          "percentage": 50
        },
        "bad": {
          "count": 0,
          "percentage": 0
        },
        "operatorFirstname": "Paul"
      },
      {
        "good": {
          "count": 0,
          "percentage": 0
        },
        "ok": {
          "count": 1,
          "percentage": 100
        },
        "bad": {
          "count": 0,
          "percentage": 0
        },
        "operatorFirstname": "Ralph"
      }
    ]
  }
}

```

```

GET {{url}}incident/stats/live-scores?timeframe=day
GET {{url}}incident/stats/live-scores?timeframe=month

```

```
GET {{url}}incident/stats/live-scores?timeframe=q1
GET {{url}}incident/stats/live-scores?timeframe=q2
GET {{url}}incident/stats/live-scores?timeframe=q3
GET {{url}}incident/stats/live-scores?timeframe=q4
GET {{url}}incident/stats/live-scores?timeframe=year
```

```
{
  "current": {},
  "previous": {
    "liveScores": [
      {
        "rating": "OK",
        "ticketNumber": "I2016 010",
        "requestor": {
          "firstname": "Rest",
          "lastname": "API 10",
          "account": "Postman 10"
        }
      },
      {
        "rating": "GOOD",
        "ticketNumber": "I2016 009",
        "requestor": {
          "firstname": "Rest",
          "lastname": "API 9",
          "account": "Postman 9"
        }
      },
      {
        "rating": "GOOD",
        "ticketNumber": "I2016 017",
        "requestor": {
          "firstname": "Rest",
          "lastname": "API 17",
          "account": "Postman 17"
        }
      },
      {
        "rating": "OK",
        "ticketNumber": "I2016 016",
        "requestor": {
          "firstname": "Rest",
          "lastname": "API 16",
          "account": "Postman 16"
        }
      }
    ]
  }
}
```

```

GET {{url}}incident/stats/remarks?timeframe=day
GET {{url}}incident/stats/remarks?timeframe=month
GET {{url}}incident/stats/remarks?timeframe=q1
GET {{url}}incident/stats/remarks?timeframe=q2
GET {{url}}incident/stats/remarks?timeframe=q3
GET {{url}}incident/stats/remarks?timeframe=q4
GET {{url}}incident/stats/remarks?timeframe=year

```

```

private String remark;
    private String requestorFirstname;
    private String requestorLastname;
    private String accountName;
    private String operatorFirstname;
    private String rating;

```

```

GET {{url}}incident/stats/customer-satisfaction/relative-to?timeframe=month
{
  "current": {
    "January": {
      "requestCount": 0,
      "responseCount": 0
    }
  },
  "previous": {
    "January": {
      "requestCount": 0,
      "responseCount": 0
    }
  }
}

```

```

GET {{url}}incident/stats/customer-satisfaction/relative-to?timeframe=q1
{
  "Q1 2016": {
    "January": {
      "requestCount": 0,
      "responseCount": 0
    },
    "February": {
      "requestCount": 0,
      "responseCount": 0
    },
    "March": {
      "requestCount": 0,
      "responseCount": 0
    }
  },
  "Q1 2017": {
    "January": {

```

```

    "requestCount": 0,
    "responseCount": 0
  },
  "February": {
    "requestCount": 0,
    "responseCount": 0
  },
  "March": {
    "requestCount": 0,
    "responseCount": 0
  }
}
}
}

```

```

GET {{url}}incident/stats/customer-satisfaction/relative-to?timeframe=q2
GET {{url}}incident/stats/customer-satisfaction/relative-to?timeframe=q3
GET {{url}}incident/stats/customer-satisfaction/relative-to?timeframe=q4
GET {{url}}incident/stats/customer-satisfaction/relative-to?timeframe=year
response:

```

```

{
  "2016": {
    "January": {
      "requestCount": 0,
      "responseCount": 0
    },
    "February": {
      "requestCount": 0,
      "responseCount": 0
    },
    "March": {
      "requestCount": 0,
      "responseCount": 0
    },
    "April": {
      "requestCount": 0,
      "responseCount": 0
    },
    "May": {
      "requestCount": 0,
      "responseCount": 0
    },
    "June": {
      "requestCount": 0,
      "responseCount": 0
    },
    "July": {
      "requestCount": 0,
      "responseCount": 0
    },
    "August": {

```

```

    "requestCount": 0,
    "responseCount": 0
  },
  "September": {
    "good": {
      "count": 1,
      "percentage": 50
    },
    "ok": {
      "count": 1,
      "percentage": 50
    },
    "bad": {
      "count": 0,
      "percentage": 0
    },
    "requestCount": 2,
    "responseCount": 2
  },
  "October": {
    "good": {
      "count": 0,
      "percentage": 0
    },
    "ok": {
      "count": 0,
      "percentage": 0
    },
    "bad": {
      "count": 1,
      "percentage": 100
    },
    "requestCount": 1,
    "responseCount": 1
  },
  "November": {
    "good": {
      "count": 0,
      "percentage": 0
    },
    "ok": {
      "count": 1,
      "percentage": 100
    },
    "bad": {
      "count": 0,
      "percentage": 0
    },
    "requestCount": 1,
    "responseCount": 1
  }

```

```

    },
    "December": {
        "requestCount": 0,
        "responseCount": 0
    }
},
"2017": {
    "January": {
        "requestCount": 0,
        "responseCount": 0
    },
    "February": {
        "requestCount": 0,
        "responseCount": 0
    },
    "March": {
        "requestCount": 0,
        "responseCount": 0
    },
    "April": {
        "requestCount": 0,
        "responseCount": 0
    },
    "May": {
        "requestCount": 0,
        "responseCount": 0
    },
    "June": {
        "requestCount": 0,
        "responseCount": 0
    },
    "July": {
        "requestCount": 0,
        "responseCount": 0
    },
    "August": {
        "requestCount": 0,
        "responseCount": 0
    },
    "September": {
        "requestCount": 0,
        "responseCount": 0
    },
    "October": {
        "requestCount": 0,
        "responseCount": 0
    },
    "November": {
        "requestCount": 0,
        "responseCount": 0
    }
}

```



```

    },
    "December": {
      "requestCount": 0,
      "responseCount": 0
    }
  }
}

```

```

GET {{url}}incident/stats/customer-satisfaction/{{operatorId}}/relative-to?timeframe=month
GET {{url}}incident/stats/customer-satisfaction/{{operatorId}}/relative-to?timeframe=q1
GET {{url}}incident/stats/customer-satisfaction/{{operatorId}}/relative-to?timeframe=q2
GET {{url}}incident/stats/customer-satisfaction/{{operatorId}}/relative-to?timeframe=q3
GET {{url}}incident/stats/customer-satisfaction/{{operatorId}}/relative-to?timeframe=q4
GET {{url}}incident/stats/customer-satisfaction/{{operatorId}}/relative-to?timeframe=year

```

```

GET {{url}}incident/stats/operator-scores/relative-to?timeframe=month
GET {{url}}incident/stats/operator-scores/relative-to?timeframe=q1
GET {{url}}incident/stats/operator-scores/relative-to?timeframe=q2
GET {{url}}incident/stats/operator-scores/relative-to?timeframe=q3
GET {{url}}incident/stats/operator-scores/relative-to?timeframe=q4
GET {{url}}incident/stats/operator-scores/relative-to?timeframe=year

```

response:

```

{
  "2016": {
    "January": [],
    "February": [],
    "March": [],
    "April": [],
    "May": [],
    "June": [],
    "July": [],
    "August": [],
    "September": [
      {
        "good": {
          "count": 1,
          "percentage": 50
        },
        "ok": {
          "count": 1,
          "percentage": 50
        },
        "bad": {
          "count": 0,
          "percentage": 0
        }
      },
      "operatorFirstname": "Paul"
    ]
  }
}

```

```

],
"October": [],
"November": [
  {
    "good": {
      "count": 0,
      "percentage": 0
    },
    "ok": {
      "count": 1,
      "percentage": 100
    },
    "bad": {
      "count": 0,
      "percentage": 0
    },
    "operatorFirstname": "Ralph"
  }
],
"December": []
},
"2017": {
  "January": [],
  "February": [],
  "March": [],
  "April": [],
  "May": [],
  "June": [],
  "July": [],
  "August": [],
  "September": [],
  "October": [],
  "November": [],
  "December": []
}
}

```

```

GET {{url}}incident/stats/operator-scores/{{operatorId}}/relative-to?timeframe=month
GET {{url}}incident/stats/operator-scores/{{operatorId}}/relative-to?timeframe=q1
GET {{url}}incident/stats/operator-scores/{{operatorId}}/relative-to?timeframe=q2
GET {{url}}incident/stats/operator-scores/{{operatorId}}/relative-to?timeframe=q3
GET {{url}}incident/stats/operator-scores/{{operatorId}}/relative-to?timeframe=q4
GET {{url}}incident/stats/operator-scores/{{operatorId}}/relative-to?timeframe=year
response:

```

```

{
  "2016": {
    "January": [],
    "February": [],
    "March": [],

```

```

    "April": [],
    "May": [],
    "June": [],
    "July": [],
    "August": [],
    "September": [],
    "October": [
      {
        "good": {
          "count": 1,
          "percentage": 100
        },
        "ok": {
          "count": 0,
          "percentage": 0
        },
        "bad": {
          "count": 0,
          "percentage": 0
        }
      },
      "operatorFirstname": "Jos"
    ]
  ],
  "November": [],
  "December": []
},
"2017": {
  "January": [],
  "February": [],
  "March": [],
  "April": [],
  "May": [],
  "June": [],
  "July": [],
  "August": [],
  "September": [],
  "October": [],
  "November": [],
  "December": []
}
}

```

Bijlage XI: Afstudeerplan

Informatie afstudeerder en gastbedrijf *(structuur niet wijzigen)*

Afstudeerblok: 2016-1.2 (start uiterlijk 9 mei 2016)

Startdatum uitvoering afstudeeropdracht: 9 mei 2016

Inleverdatum afstudeerdossier volgens jaarrooster: 10 oktober 2016

Studentnummer: 20052930

Achternaam: dhr. Verhaar

Voorletters: JJM

Roepnaam: Jos

Adres: Meidoornstraat 11

Postcode: 2313 XD

Woonplaats: Leiden

Telefoonnummer: n.v.t.

Mobiel nummer: 06 22 18 18 17

Privé emailadres: josverhaar@gmail.com

Opleiding: Informatica

Locatie: Den Haag

Variant: deeltijd c.q. avond

Naam studieloopbaanbegeleider: Merie Heijne

Naam begeleidend examiner: Arie Toet

Naam tweede examiner: Gerard Mijnares

Naam bedrijf: Mailplus B.V.

Afdeling bedrijf: Development

Bezoekadres bedrijf: Boris Pasternaklaan 18

Postcode bezoekadres: 2719 DA

Postbusnummer: n.v.t.

Postcode postbusnummer: n.v.t.

Plaats: Zoetermeer

Telefoon bedrijf: 079 363 12 10

Telefax bedrijf: n.v.t.

Internetsite bedrijf: www.mailplus.nl

Achternaam opdrachtgever: dhr. Paul Nederlof

Voorletters opdrachtgever: P.V.

Titulatuur opdrachtgever: n.v.t.

Functie opdrachtgever: Teammanager Support

Doorkiesnummer opdrachtgever: 70

Email opdrachtgever: paulne@mailplus.nl

Achternaam bedrijfsmentor: dhr. Niels Noorlander

Voorletters bedrijfsmentor: N.E.

Titulatuur bedrijfsmentor: ir.

Functie bedrijfsmentor: Team Manager Product Management

Doorkiesnummer bedrijfsmentor: 079 363 7065

Email bedrijfsmentor: niels@mailplus.nl

Doorkiesnummer afstudeerder: n.v.t.

Functie afstudeerder (deeltijd/duaal): Medewerker Support

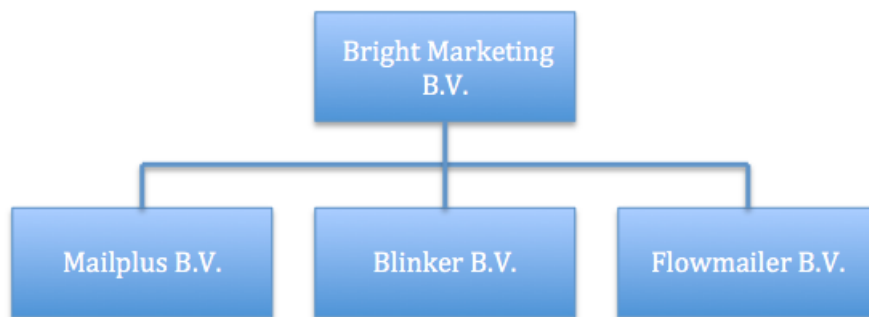
Titel afstudeeropdracht: "Ontwikkelen van een nieuw Support Feedback Systeem voor Mailplus B.V."

Opdrachtomschrijving

1. Bedrijf

Mailplus B.V. is een E-mail Service Provider (hierna: ESP). Het bedrijf levert een e-mail marketingtool met de naam MailPlus als SaaS oplossing. Het primaire doel van dit pakket is het versturen van marketing uitingen via e-mail. Hierbij is te denken aan nieuwsbrieven en serviceberichten. De software biedt verder de mogelijkheid tot het inrichten van uitgebreide campagnes. Bij deze campagnes kun je denken aan verlatenwinkelwagen campagne, verjaardag campagne, profiel verrijking campagne, enzovoort. Verder is het mogelijk om pagina's aan te maken waarop je formulieren, survey's, polls, en dergelijke kunt plaatsen. Mailplus B.V. verzorgt hiervan de verkoop, het onderhoud en uitbreiden van de software, het operationeel houden van de software en het geven van support aan de gebruikers. In de hoedanigheid van ESP zijn alle servers in eigenbeheer, maar extern gehuisvest.

Mailplus bestaat sinds 2004 en is een onderdeel van Bright Marketing Holding. Hieronder vallen eveneens Blinker B.V. en Flowmailer B.V. Binnen de Bright Marketing Holding zijn ongeveer 80 mensen werkzaam, waarvan 36 bij Mailplus B.V. In onderstaand organogram is de onderlinge verhouding inzichtelijk gemaakt van de verschillende B.V.'s.

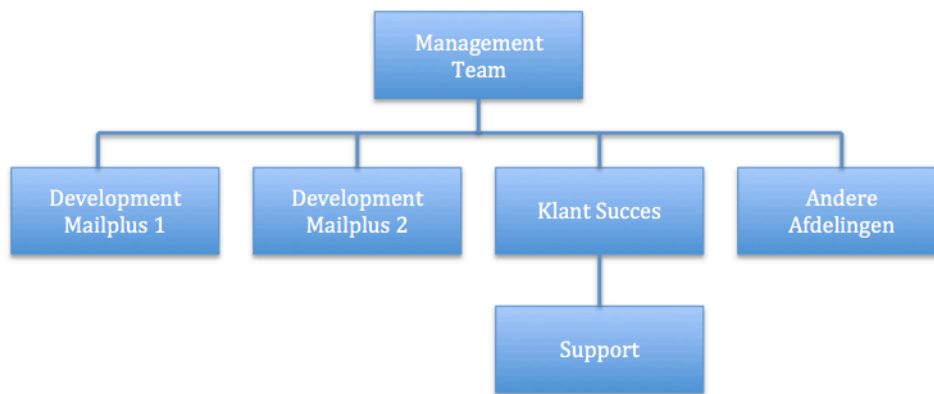


Figuur 1: Organogram Bright Marketing Holding

De afdeling support behandelt jaarlijks 20.000 support aanvragen van gebruikers. Ook collega's vallen onder de gebruikers. De afdeling support bestaat uit 8 personen. Een deel hiervan werkt parttime. De support aanvragen zijn verschillend van aard. Het kunnen eenvoudige vragen zijn over bijvoorbeeld e-mail templates, of meer ingewikkelde vragen over onze koppelingen. Hierbij kun je denken aan vragen over de werking en integratie van onze REST-API of SOAP-API.

De opdracht wordt uitgevoerd bij één van de twee product development teams. Deze teams bevinden zich direct onder het management team. Elk team is opgebouwd uit software developers en product managers. De product managers brengen klantwensen in kaart en beschrijven nieuwe features die vanuit de business units worden aangedragen. In samenwerking met de developers worden die wensen en features uitgewerkt en geïmplementeerd. Op dit moment werkt het team waar de opdracht uitgevoerd zal worden met name aan het implementeren van klantwensen. Het andere team richt zich op integraties tussen MailPlus en CRM/Webshop systemen. Beide teams werken volgens de Agile ontwikkelmethode SCRUM.

In onderstaand organogram is zichtbaar gemaakt hoe de support afdeling en beide development afdelingen zich verhouden binnen Mailplus B.V.



Afbeelding 1: Organogram Support ten opzichte van de Development teams

2. Probleemstelling

--Wat is het Support Feedback Systeem?

Het Support Feedback Systeem (hierna: SFS) is een geautomatiseerd systeem dat intern is ontwikkeld. Het doel van het systeem is het doorlopend meten van de klanttevredenheid. Het SFS wordt alleen intern gebruikt door de supportafdeling van Mailplus B.V. en haar gebruikers. Klanten geven feedback over de hulp die geleverd is door een support medewerker. De medewerkers van support worden beoordeeld op hun functioneren naar aanleiding van de statistieken die uit dit systeem worden gehaald. Een support medewerker verstuurt, na het sluiten van een support aanvraag, een feedback e-mail. De klant kan via deze mail de geleverde support beoordelen. Feedback kan gegeven worden door op één van de drie smiley-iconen te klikken. Deze smiley-iconen staan ieder voor een andere gradatie in tevredenheid. 'Ontevreden' (rood) staat voor de meest negatieve beoordeling, 'gewoon ok' (geel) staat voor voldoende support en 'zeer tevreden' (groen) staat voor zeer goede support. Desgewenst is er de mogelijkheid tot het toevoegen van een schriftelijke reactie. De gegeven feedback wordt via een scherm getoond op de afdeling.

De huidige functionaliteiten van het SFS omvatten het genereren van unieke links voor het kunnen geven van feedback. Daarnaast het genereren van een e-mail aan de klant met een vaste indeling en opmaak, het opslaan van ontvangen feedback en het opvragen van beperkte statistieken uit het systeem.

--Wat wil MailPlus B.V. bereiken?

Meerdere klanten hebben interesse getoond in het SFS. Naar aanleiding hiervan is besloten om van het SFS een product te maken, zodat klanten in staat zijn om op eenvoudige wijze de tevredenheid van hun eigen klanten doorlopend te kunnen meten. Doordat het SFS niet wordt gezien als een marketing tool wordt er met dit product een andere klantgroep aangesproken. Hierdoor is het niet wenselijk om MailPlus data te vermengen met de data van het SFS. Om deze reden is ervoor gekozen om het SFS apart op te zetten. Klanten kunnen door middel van authenticatie tokens gebruikmaken van het SFS. Zelf dienen ze een dashboard te ontwikkelen om de data te presenteren.

--Waarom voldoet de huidige implementatie niet?

Op dit moment kan het SFS niet één-op-één worden gebruikt door meerdere gebruikers. De huidige implementatie is ontwikkeld voor intern gebruik door de support afdeling van Mailplus B.V. Ook de functionaliteiten van het SFS zijn hierop gericht. Tijdens de ontwikkeling van het SFS is er geen rekening gehouden met meerdere gebruikers. Dit betekent bijvoorbeeld dat het database ontwerp enkel bedacht is voor de huidige vorm van feedback vragen (de drie smileys), dat de gegevens die je uit het systeem kunt opvragen beperkt zijn en dat de configuratie gedaan moet worden in de database. Het bericht dat verzonden wordt aan de klant is niet aan te passen en alleen beschikbaar in het Nederlands. Momenteel wordt er dan ook geen feedback gevraagd aan de Engels sprekende klanten van Mailplus B.V. Een groot probleem van het koppelen van meerdere gebruikers aan het huidige SFS is dat de gegeven feedback door elkaar gaat. Dit betekent dat iedere gebruiker, iedere feedback kan zien. Deze vermenging van gegevens is niet gewenst. Uit deze voorbeelden blijkt dat de huidige opzet niet voldoet om deze één-op-één in te kunnen zetten door de klanten van Mailplus B.V. Om dit te realiseren dient er een nieuwe opzet ontwikkeld te worden van het gehele SFS systeem.

Hierbij wordt rekening gehouden met wensen vanuit nieuwe- en bestaande gebruikers. Ook zal er worden gekeken naar wat vergelijkbare pakketten momenteel voor functionaliteit aanbieden.

3. Doelstelling van de afstudeeropdracht

Het doel van de afstudeeropdracht is het ontwikkelen van een nieuw SFS waarmee klanten hun klanttevredenheid doorlopend kunnen meten. Via een REST-API koppeling is het mogelijk om statistieken uit het systeem op te vragen en deze inzichtelijk te maken met een zelf te ontwikkelen dashboard.

4. Resultaat

Na het uitvoeren van de afstudeeropdracht heeft Mailplus B.V. een nieuw SFS. Het nieuwe SFS biedt klanten van Mailplus de mogelijkheid hun klanttevredenheid te registreren. Ook is het mogelijk om het systeem intern op meerdere afdelingen in te zetten. Het nieuwe SFS bestaat uit een backend met dataopslag en een REST-API waarmee gecommuniceerd kan worden. De front-end bestaat uit een webpagina waar gebruikers de feedback kunnen versturen en een reactie op kunnen plaatsen. Er zal een eenvoudige applicatie voor presentatie doeleinden worden ontwikkeld. De nieuwe applicatie wordt bij Mailplus gehost. Hierdoor hoeven gebruikers deze applicatie niet zelf te hosten. Door middel van een koppeling met hun eigen systeem kunnen ze gebruik maken van het nieuwe SFS.

5. Uit te voeren werkzaamheden, inclusief een globale fasering, mijlpalen en bijbehorende activiteiten

-) Opbouw afstudeerdossier
duur: 15 dagen
-) Schrijven plan van aanpak
duur: 1 dag
-) Inlezen huidige situatie (software en documentatie)
duur: 1 dag
-) Huidige situatie inzichtelijk maken (maken klassendiagram aan de hand van de aangeleverde code)
duur: 2 dagen
-) Concurrenten onderzoek
duur: 5 dagen
methoden: kwalitatief onderzoek
doel: onderzoeken wat gelijksoortige systemen doen/ hebben op het gebied van feedback systemen
-) Vaststellen functionele- en niet functionele requirements
duur: 7 dagen
methoden: interview technieken, requirements analyse
-) Ontwerp database
duur: 5 dagen
methoden:
-) Ontwerp software
duur: 14 dagen
methoden: Design Patterns, Software ontwikkelmethoden
-) Ontwerp tests
duur: 5 dagen
methoden: Test Goal, opstellen testgevallen
-) Coderen
duur: 21 dagen
-) Testen
duur: 4 dagen
methoden: test ontwerpen, TestGoal
-) Schrijven documentatie
duur: 4 dagen
-) Overdracht
duur: 1 dag

6. Op te leveren (tussen)producten

Tijdens het uitvoeren van dit afstudeerplan worden de volgende (tussen)producten opgeleverd:

- Plan van Aanpak:
Uitleg wat de opdracht inhoudt, inclusief een gedetailleerde planning
- Klassen- en Database diagram huidige implementatie
Om de huidige implementatie beter te begrijpen wordt hiervan een klassen- en database diagram opgesteld
- Uitwerking onderzoek naar gelijksoortige systemen
Het onderzoeksrapport van het gedane onderzoek naar gelijksoortige systemen op de markt
- Geïnterviewde (niet)functionele eisen
- Database ontwerp ter ondersteuning van de nieuwe applicatie
- Detail software ontwerp van de nieuwe applicatie (onder andere klassendiagram en use case diagrammen)
- Test ontwerp, -plan en -rapportage voor de nieuwe applicatie
- Geschreven code van de nieuwe applicatie
- Documentatie van de REST-API

7. Te demonstreren competenties en wijze waarop

-- 2.2 Ontwerpen, bouwen en bevragen van een database

niveau: 4 (zelfstandig, complex)

Het ontwerpen en implementeren van het gegevensmodel voor de relationele database. Het omzetten en implementeren van het fysieke ontwerp met behulp van SQL. Het uitdenken en het ontwerpen van de benodigde query's en constraints met behulp van ODL. De database draait op een Tomcat applicatie cluster met een MySQL database. Met het ontwerpen van de database structuur wordt rekening gehouden met concurrent gebruikers, performance- en beveiligingsaspecten.

-- 3.2 Ontwerpen systeemdeel

niveau: 4 (zelfstandig, complex)

Het in detail ontwerpen van de applicatie met behulp van design patterns en andere best-practices. Er zal rekening worden gehouden met de frameworks waarin de applicatie uiteindelijk wordt gebouwd.

-- 3.3 Bouwen applicatie

niveau: 3 (zelfstandig, lastig)

Het in delen implementeren en documenteren van het ontwerp tot één werkend systeem. De te ontwerpen applicatie zal geschreven worden in de programmeertaal Java. Hiervoor is gekozen omdat dit de standaard programmeertaal is binnen MailPlus. Er zal gebruik worden gemaakt van Spring MVC als web applicatie framework met Jersey (JAX-RS) voor de implementatie van de API. Voor persistence (JPA) zal gebruik worden gemaakt van EclipseLink. Als template engine wordt gebruik gemaakt van Velocity.

De user interface voor demonstratie doeleinde zal gemaakt worden in AngularJS met het Twitter Bootstrap framework.

-- 3.5 Uitvoeren van en rapporteren over het testproces

niveau: 3 (zelfstandig, lastig)

Het opstellen per testobject van de logische- en fysieke testgevallen, het bepalen van de testbasis en verwachte uitkomst. Vervolgens het uitvoeren van de tests en het noteren van de bevindingen. Aan de hand van de uitkomsten van de uitgevoerde tests zal een testrapport worden opgesteld. Als testmethode zal er gebruik worden gemaakt van TestGoal. Voor de uitvoering van geautomatiseerde testgevallen zal gebruik worden gemaakt van het pakket dat reeds in gebruik is binnen MailPlus en anders wordt JUnit gebruikt. Voor de demo interface zal voor unit tests Karma worden gebruikt en voor end-2-end tests Protractor.

Bijlage XII: Beoordeling

Faculteit IT & Design

Delft
Den Haag
Zoetermeer

Evaluatieformulier afstuderen

In te vullen door opdrachtgever c.q. bedrijfsmentor(en)

Student: Jos Verhaar

Periode:

Bedrijf c.q. instelling: MailPlus B.V.

Bedrijfsmentor: Niels Noorlander

Plaats: Zoetermeer

Datum: 6 januari 2017

1. Heeft de student zich zelf snel en goed ingewerkt in het bedrijf en de uit te voeren afstudeeropdracht?

Ja, waarbij Jos uiteraard het voordeel heeft gehad dat al bij ons bedrijf werkzaam is en dus vooral de uitdaging had om in een bestaand software project te duiken, dat matig gedocumenteerd is en buiten het dagelijks aandachtgebied van onze product development teams valt.

2. Hoe beoordeelt u de communicatieve vaardigheden van de student (in de samenwerking met collega's, in contacten met de opdrachtgever, bij mondelinge presentaties, schriftelijke rapportages)?

Op het communicatieve vlak zou ik de vaardigheden als voldoende beoordelen. In de schriftelijke communicatie komen nogal eens wat slordigheden voor die ik vooral wijt aan tijdsdruk en in veel mindere mate aan vaardigheid. In de samenwerking met collega's en opdrachtgever vind ik dat Jos heel goed zijn verhaal kan overbrengen, duidelijk formuleert en goed luistert.

3. Hoe heeft de student tijdens het uitvoeren van de opdracht gefunctioneerd?

- Qua verantwoordelijkheid ~~goed~~ / voldoende / ~~matig~~ / onvoldoende
- Qua zelfstandigheid goed / ~~voldoende~~ / ~~matig~~ / onvoldoende
- Qua planmatig werken ~~goed~~ / voldoende / ~~matig~~ / onvoldoende
- Qua creativiteit goed / ~~voldoende~~ / ~~matig~~ / onvoldoende
- Qua productiviteit ~~goed~~ / voldoende / ~~matig~~ / onvoldoende
- Qua samenwerken met collega's ~~goed~~ / voldoende / ~~matig~~ / onvoldoende
- Qua draagvlakontwikkeling ~~goed~~ / voldoende / ~~matig~~ / onvoldoende
- Qua inspelen op bedrijfscultuur goed / ~~voldoende~~ / ~~matig~~ / onvoldoende
- Qua rekening houden met de specifieke context van het bedrijf goed / ~~voldoende~~ / ~~matig~~ / onvoldoende
- Qua het op gang brengen van de nodige veranderingen ~~goed~~ / voldoende / ~~matig~~ / onvoldoende

4. Hoe beoordeelt u de kennis en kunde van de student in verhouding tot wat u verwacht van een bijna afgestudeerde?

Jos heeft wat mij betreft aangetoond bij het uitvoeren van zijn opdracht dat hij zowel de theorie als de praktijk kennis en kunde heeft die ik verwacht van een student in de laatste fase van zijn studie.

Ook het uitzoeken van en inlezen in kennis die nieuw voor hem was tijdens het uitvoeren van de afstudeeropdracht is hem goed af gegaan.

Meeste moeite had Jos bij het realistisch plannen van zijn werkzaamheden en het bewaken van de planning.

5. Hoe beoordeelt u de kwaliteit van de opgeleverde (tussen)producten?

Goed: door de gestructureerde aanpak van ontwerp, implementatie en testen in een iteratief proces denk ik dat Jos een product heeft opgeleverd dat van goede kwaliteit is en een goede basis biedt om in de toekomst als een volwaardig product te gaan verkopen.

6. Bent u tevreden over het opgeleverde (eind)product?

- **In hoeverre heeft u gekregen wat is afgesproken?**

Uitgangspunt was vooral de ombouw van een maatwerk oplossing tot een SaaS product. Dat is gelukt in het door Jos opgeleverde product. Daarnaast heeft hij een aantal toevoegingen gedaan die hoog op de wensenlijst van de opdrachtgever stonden, waardoor deze nieuwe versie van het product ook al gewenst is voor intern gebruik, dus nog ongeacht of we het als product aan derden willen leveren. Dat is dus meer dan we in eerste instantie hadden afgesproken.

- **In hoeverre voldoet het (eind)product aan uw verwachtingen?**

Het voldoet volledig aan mijn verwachtingen.

- **Wat is de bruikbaarheid en onderhoudbaarheid hiervan?**

Zoals hier boven al aangegeven is het door Jos opgeleverde resultaat sowieso al inzetbaar als vervanging van het huidige maatwerk. Om het SFS ook als product te gaan verkopen zullen er nog extra stappen gezet moeten worden, die we met opzet buiten de scope van deze opdracht hebben gehouden, met name op het gebied van tooling voor de levering en support van het product, instellen van het product naar wens van de klant, etc.

- **Wat gebeurt er met het opgeleverde (eind)product?**

We zullen het product opnemen in het portfolio van onze product development en in eerste instantie in gebruik gaan nemen als vervanging van het huidige maatwerk. Daarnaast zullen we met commercieel product management een plan moeten maken welke stappen er gezet moeten worden om het product verder te ontwikkelen (ook wat betreft documentatie, pricing, support, etc.) om het tot een volwaardig SaaS product te laten groeien.

- **Kunt u direct met het opgeleverde product aan de slag?**

Ja, dat kan en gaan we ook doen.

7. Zijn er nog aspecten voor u van belang die nog niet aan de orde zijn geweest?

In eerste instantie leek Jos het uitvoeren van een afstudeeropdracht naast zijn werk als MailPlus support medewerker wat te onderschatten, maar de afgelopen maanden heeft hij er hard aan

getrokken met ondersteuning van het thuisfront, wat niet altijd even makkelijk is met een heel jong kind. Veel respect en waardering daar voor van mijn kant.

8. Bent u bereid een volgende keer weer uw medewerking te verlenen aan het beschikbaar stellen van een afstudeerplaats (graag met toelichting)?

Ja, dat lijkt me zeker, hoewel ik dan wel zou willen gaan voor de situatie waarbij de afstudeerder het gros van zijn of haar tijd bij ons in de development teams aan de slag gaat, omdat het uitvoeren van de opdracht volledig in eigen tijd zowel zwaar is voor de student maar ook suboptimaal voor de begeleiding.

Bijlage XIII: Plan van aanpak

Plan van aanpak

Auteur: Jos Verhaar (20052930)
Bedrijf: Mailplus B.V.
Opdrachtgever: Paul Nederlof
Bedrijfsmentor: Niels Noorlander
Examinatoren: Arie Toet & Gerard Mijharends
Datum: 9 januari 2017
Instelling: Haagse Hogeschool
Stageperiode: juni 2016-februari 2017

1. Inleiding

1.1 Doel van het document

In het verslag wordt de uitvoering van de afstudeeropdracht beschreven. Het verslag is geschreven met als doel om geïnteresseerden te informeren over het proces van de afstudeeropdracht.

In het verslag wordt het proces om het Support Feedback Systeem (hierna: SFS) te ontwikkelen, beschreven. Het gehele proces en de bijbehorende keuzes die gedurende de ontwikkeling zijn gemaakt worden hierin beschreven.

1.2 Leeswijzer

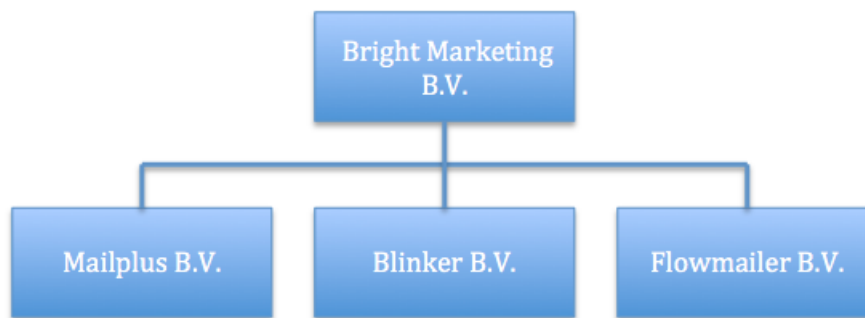
Vanwege de leesbaarheid van het document zal een “veel gebruikte termen” lijst worden opgesteld. Met name om hierin de bedrijfsterminologie en de technische termen in te verduidelijken. Voor de rest wordt aangenomen dat de lezer basiskennis bezit van software ontwikkeling.

De afstudeeropdracht is uitgevoerd bij Mailplus B.V. In hoofdstuk twee wordt een organisatieverschrijving gegeven. Alsmede een omschrijving van de werkomgeving en tot slot een overzicht van alle betrokkenen bij deze afstudeeropdracht. Vervolgens wordt in hoofdstuk drie een uitgebreide opdrachtdefinitie gegeven. In hoofdstuk vier worden de methoden en technieken, die tijdens het onderzoek zijn gebruikt, beschreven. In hoofdstuk 5 en 6 wordt zowel de aanpak als de uitvoering beschreven. Na deze hoofdstukken volgen de sprint hoofdstukken. In elk hoofdstuk wordt per sprint beschreven wat er gedaan is, hoe het gedaan is en welke keuzes zijn gemaakt. Aan het einde van elke sprint wordt er gereflecteerd op het proces. Tot slot zijn in de laatste hoofdstukken de procesevaluatie en productevaluatie opgenomen. Aan het einde van het document zijn de bijlagen opgenomen.

2. Het bedrijf

2.1 Organisatiebeschrijving

Mailplus B.V. is een E-mail Service Provider (hierna: ESP). Het bedrijf levert een e-mail marketingtool met de naam MailPlus als SaaS oplossing. Mailplus bestaat sinds 2004 en is een onderdeel van Bright Marketing Holding. Hieronder vallen eveneens Blinker B.V. en Flowmailer B.V. Binnen de Bright Marketing Holding zijn ongeveer 80 mensen werkzaam, waarvan 36 bij Mailplus B.V. In onderstaand organogram is de onderlinge verhouding inzichtelijk gemaakt van de verschillende B.V.'s.

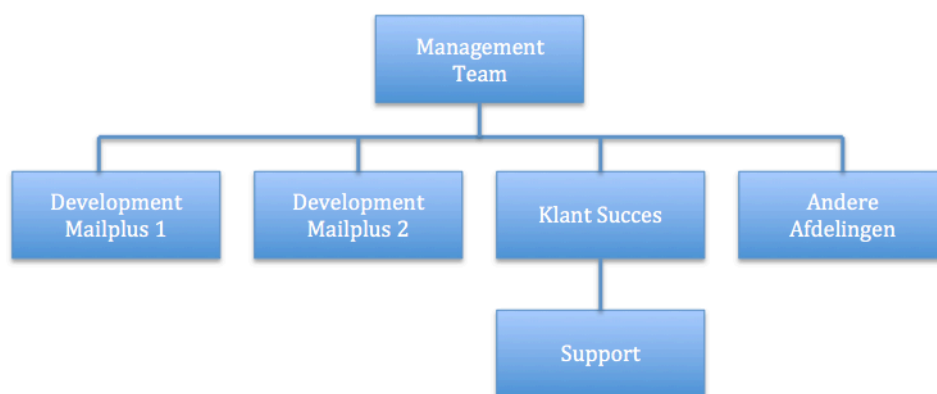


Figuur 3: Organogram Bright Marketing Holding

De afdeling support behandelt jaarlijks 20.000 support aanvragen van gebruikers. Ook collega's vallen onder de gebruikers. De afdeling support bestaat uit 8 personen. Een deel hiervan werkt parttime. De support aanvragen zijn verschillend van aard. Het kunnen eenvoudige vragen zijn over bijvoorbeeld e-mail templates, of meer ingewikkelde vragen over onze koppelingen. Hierbij kun je denken aan vragen over de werking en integratie van onze REST-API of SOAP-API.

De opdracht wordt uitgevoerd bij één van de twee product development teams. Deze teams bevinden zich direct onder het management team. Elk team is opgebouwd uit software developers en product managers. De product managers brengen klantwensen in kaart en beschrijven nieuwe features die vanuit de business units worden aangedragen. In samenwerking met de developers worden die wensen en features uitgewerkt en geïmplementeerd. Op dit moment werkt het team waar de opdracht uitgevoerd zal worden met name aan het implementeren van klantwensen. Het andere team richt zich op integraties tussen MailPlus en CRM/Webshop systemen. Beide teams werken volgens de Agile ontwikkelmethode Scrum.

In onderstaand organogram (figuur 2) is zichtbaar gemaakt hoe de support afdeling en beide development afdelingen zich verhouden binnen Mailplus B.V.



Figuur 4: Organogram Support ten opzichte van de Development teams.

2.2 Jouw werkomgeving

Het primaire doel van het MailPlus pakket is het versturen van marketing uitingen via e-mail. Hierbij is te denken aan nieuwsbrieven en serviceberichten. De software biedt verder de mogelijkheid tot het inrichten van uitgebreide campagnes. Bij deze campagnes kun je denken aan verlatenwinkelwagen campagne, verjaardag campagne, profiel verrijking campagne, enzovoort. Verder is het mogelijk om pagina's aan te maken waarop je formulieren, survey's, polls, en dergelijke kunt plaatsen. Mailplus B.V. verzorgt hiervan de verkoop, het onderhoud en uitbreiden van de software, het operationeel houden van de software en het geven van support aan de gebruikers. In de hoedanigheid van ESP zijn alle servers in eigenbeheer, maar extern gehuisvest.

2.3 Betrokkenen afstudeeropdracht

Bij de afstudeeropdracht zijn een aantal medewerkers van Mailplus B.V. betrokken. In dit document zal worden gerefereerd naar de rollen.

Naam	Email	Rol
Jos Verhaar	josverhaar@gmail.com jos@mailplus.nl	Afstudeerder
Paul Nederlof	paulne@mailplus.nl	Opdrachtgever
Niels Noorlander	niels@mailplus.nl	Bedrijfsmentor

3. Opdrachtdefinitie

3.1 Opdrachtoomschrijving

3.1.1 Wat is het Support Feedback Systeem?

Het Support Feedback Systeem (hierna: SFS) is een geautomatiseerd systeem dat intern is ontwikkeld. Het doel van het systeem is het doorlopend meten van de klanttevredenheid. Het SFS wordt alleen intern gebruikt door de supportafdeling van Mailplus B.V. en haar gebruikers. Klanten geven feedback over de hulp die geleverd is door een support medewerker. De medewerkers van support worden beoordeeld op hun functioneren naar aanleiding van de statistieken die uit dit systeem worden gehaald.

3.1.2 Huidige implementatie

Wanneer een support medewerker de vraag van een gebruiker heeft afgehandeld, wordt er door TopDesk een e-mail gegenereerd met een JSON-object. Deze e-mail is de trigger voor het SFS.



E-mail versturen voor I1508 964
Feedback support

Adres

Afzender:

Naar ->

☐ Kopie naar afzender

Bericht

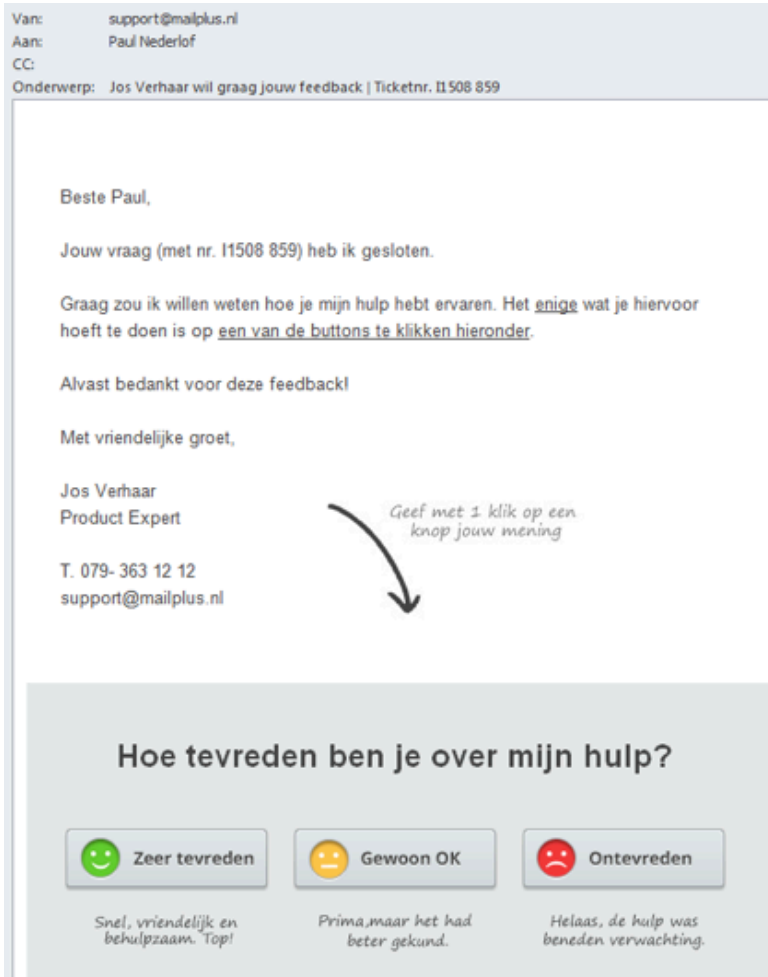
Onderwerp:

☐ HTML e-mail

```
{
  "id": "I1508 964",
  "requestor": {
    "firstname": "Paul",
    "lastname": "Nederlof",
    "email": "paulne@mailplus.nl"
  },
  "operator": {
    "lastname": "van Delft",
    "firstname": "Ralph",
    "number": "8",
    "email": "ralph@mailplus.nl"
  },
  "account": "Medewerkers"
}
```

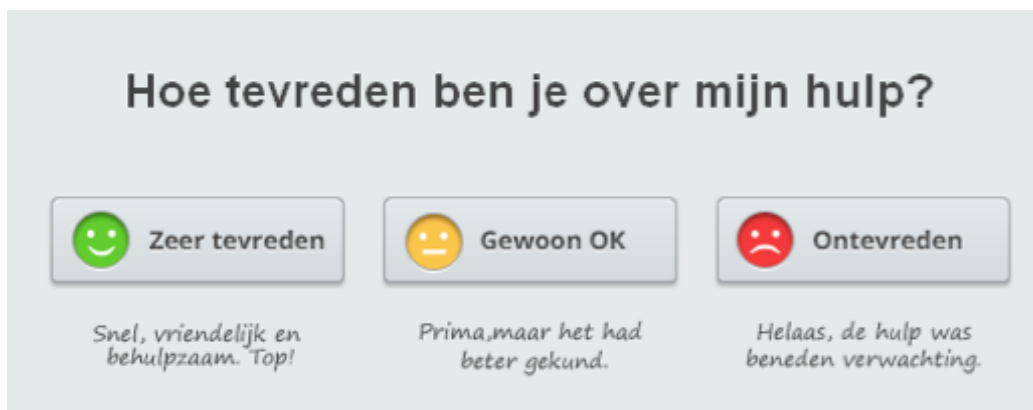
Afbeelding 54: Mail met JSON-object dat vanuit TopDesk wordt gestuurd

Het SFS leest de e-mail in en verwerkt deze tot een andere e-mail, het feedbackverzoek, welke naar de gebruiker wordt verstuurd.




Afbeelding 55: Voorbeeld van de feedbackmail

Vanuit het feedbackverzoek kan de gebruiker zijn waardering geven. Het enige dat de gebruiker hiervoor hoeft te doen is te klikken op een van de smiley-iconen onder in de mail.



Afbeelding 56: Close-up smiley's in de feedbackmail

Na het klikken heeft de gebruiker zijn waardering gegeven en wordt de gebruiker naar een dankpagina geleid. Op deze pagina is er de mogelijkheid tot het geven van een schriftelijke reactie.



Afbeelding 57: Landingspagina met mogelijkheid tot geven toelichting

Een support medewerker verstuurt, na het sluiten van een support aanvraag, een feedback e-mail. De klant kan via deze mail de geleverde support beoordelen. Feedback kan gegeven worden door op één van de drie smiley-iconen te klikken. Deze smiley-iconen staan ieder voor een andere gradatie in tevredenheid. 'Ontevreden' (rood) staat voor de meest negatieve beoordeling, 'gewoon ok' (geel) staat voor voldoende support en 'zeer tevreden' (groen) staat voor zeer goede support. Desgewenst is er de mogelijkheid tot het toevoegen van een schriftelijke reactie. De gegeven feedback wordt via een scherm getoond op de afdeling. De huidige functionaliteiten van het SFS omvatten het genereren van unieke links voor het kunnen geven van feedback. Daarnaast het genereren van een e-mail aan de klant met een vaste indeling en opmaak, het opslaan van ontvangen feedback en het opvragen van beperkte statistieken uit het systeem.

3.1.3 Wat wil MailPlus B.V. bereiken?

Meerdere klanten hebben interesse getoond in het SFS. Naar aanleiding hiervan is besloten om van het SFS een product te maken, zodat klanten in staat zijn om op eenvoudige wijze de tevredenheid van hun eigen klanten doorlopend te kunnen meten. Doordat het SFS niet wordt gezien als een marketing tool wordt er met dit product een andere klantgroep aangesproken. Hierdoor is het niet wenselijk om MailPlus data te vermengen met de data van het SFS. Om deze reden is ervoor gekozen om het SFS apart op te zetten. Klanten kunnen door middel van authenticatie tokens gebruikmaken van het SFS. Zelf dienen ze een dashboard te ontwikkelen om de data te presenteren.

3.1.4 Waarom voldoet de huidige implementatie niet?

Op dit moment kan het SFS niet één-op-één worden gebruikt door meerdere gebruikers. De huidige implementatie is ontwikkeld voor intern gebruik door de support afdeling van Mailplus B.V. Ook de functionaliteiten van het SFS zijn hierop gericht. Tijdens de ontwikkeling van het SFS is er geen rekening gehouden met meerdere gebruikers. Dit betekent bijvoorbeeld dat het database ontwerp enkel bedacht is voor de huidige vorm van feedback vragen (de drie smileys), dat de gegevens die je uit het systeem kunt opvragen beperkt zijn en dat de configuratie gedaan moet worden in de database. Het bericht dat verzonden wordt aan de klant is niet aan te passen en alleen beschikbaar in het Nederlands. Momenteel wordt er dan ook geen feedback gevraagd aan de Engels sprekende klanten van Mailplus B.V. Een groot probleem van het koppelen van meerdere gebruikers aan het huidige SFS is dat de gegeven feedback door elkaar gaat. Dit betekent dat iedere gebruiker, iedere feedback kan zien. Deze vermenging van gegevens is niet gewenst. Uit deze voorbeelden blijkt dat de huidige opzet niet voldoet om deze één-op-één in te kunnen zetten door de klanten van Mailplus B.V. Om dit te realiseren dient er een nieuwe opzet ontwikkeld te worden van het gehele SFS systeem. Hierbij wordt rekening gehouden met wensen vanuit nieuwe- en bestaande gebruikers. Ook zal er worden gekeken naar wat vergelijkbare pakketten momenteel voor functionaliteit aanbieden.

3.2 Doelstelling

Het doel van de afstudeeropdracht is het ontwikkelen van een nieuw SFS waarmee klanten hun klanttevredenheid doorlopend kunnen meten. Via een REST-API koppeling is het mogelijk om statistieken uit het systeem op te vragen en deze inzichtelijk te maken met een zelf te ontwikkelen dashboard.

3.3 Gewenste resultaat

Na het uitvoeren van de afstudeeropdracht heeft Mailplus B.V. een nieuw SFS. Het nieuwe SFS biedt klanten van Mailplus B.V. de mogelijkheid hun klanttevredenheid te registreren. Ook is het mogelijk om het systeem intern op meerdere afdelingen in te zetten. Het nieuwe SFS bestaat uit een backend met dataopslag en een REST-API waarmee gecommuniceerd kan worden. De front-end bestaat uit een webpagina waar gebruikers de feedback kunnen versturen en een reactie op kunnen plaatsen. Er zal een eenvoudige applicatie voor presentatie doeleinden worden ontwikkeld. De nieuwe applicatie wordt bij Mailplus B.V. gehost. Hierdoor hoeven gebruikers deze applicatie niet zelf te hosten. Door middel van een koppeling met hun eigen systeem kunnen ze gebruik maken van het nieuwe SFS.

3.4 Op te leveren producten

Tijdens het uitvoeren van deze afstudeeropdracht worden de volgende (tussen)producten opgeleverd:

- Plan van Aanpak: Uitleg wat de opdracht inhoudt, inclusief een gedetailleerde planning.
- Klassen- en Database diagram huidige implementatie: Om de huidige implementatie beter te begrijpen wordt hiervan een klassen- en database diagram opgesteld.
- Uitwerking onderzoek naar gelijksoortige systemen (concurrentie onderzoek). Het onderzoeksrapport van het gedane onderzoek naar gelijksoortige systemen op de markt.

- Geïntariseerde (niet)functionele eisen.
- Database ontwerp ter ondersteuning van de nieuwe applicatie.
- Detail software ontwerp van de nieuwe applicatie (onder andere klassendiagram en use case diagrammen).
- Test ontwerp, -plan en -rapportage voor de nieuwe applicatie.
- Documentatie van de REST-API.
- Het afstudeerverslag.

4. Randvoorwaarden

In dit hoofdstuk worden de afspraken rondom dit afstudeeropdracht beschreven. Tevens zal de scope en hierin de afbakening van de opdracht worden aangegeven.

4.1 Afspraken.

4.1.1 Sprint planning

Een sprint duurt twee weken. Elke week wordt de voortgang van de sprint besproken met de bedrijfsmentor. Aan het einde van de sprint wordt er, samen met de bedrijfsmentor, teruggekeken hoe de sprint is gegaan, de sprint retrospective, en welke producten er zijn opgeleverd. Tevens wordt er een planning en indeling gemaakt voor de volgende sprint.

4.1.2 Afstudeerverslag

Elke maandag zal er gewerkt worden aan het verslag. Tevens wordt hier aan gewerkt tijdens de avonden en in de weekenden.

4.1.3 Afspraken met de Haagse Hogeschool

Na ongeveer 25% van de looptijd zal de eerste stagecoördinator bij het bedrijf op bezoek komen. Na ongeveer 45% van de doorlooptijd zal een voortgangsverslag naar de eerste stagecoördinator gestuurd worden. Met het bijhorende afstudeerdossier dat alle gemaakte producten bevat. Bij ongeveer 60% van de doorlooptijd van de afstudeeropdracht zal er tijdens een tussentijds assessment het afstudeerdossier worden besproken met zowel de eerste stagecoördinator als de tweede, begeleidend stagecoördinator.

4.1.4 Afspraken met bedrijfsmentor

Vanwege de aanwezigheid van mogelijke risicofactoren die het onderzoek kunnen belemmeren (zie voor verdere informatie hoofdstuk 5 uit het plan van aanpak), zijn er vaste afspraken met de bedrijfsmentor gemaakt. Zo is er elke maandag een overleg gepland om de voortgang van de afstudeeropdracht te bespreken, maar zo ook de wensen en de eventuele wijzigingen hierin met de bedrijfsmentor te bespreken. Om ervoor te zorgen dat de bedrijfsmentor up to date is wat betreft de voortgang van het onderzoek, heeft de bedrijfsmentor toegang tot deze documenten middels een Google Drive account. De bedrijfsmentor levert hierop zijn input. Dit wordt tevens besproken tijdens de vaste overlegmomenten.

4.1.5 Scope van de afstudeeropdracht

Tijdens de besprekingen met de bedrijfsmentor zijn er afspraken gemaakt van de complexiteit van de opdracht. Er is gekozen om het nieuwe SFS als losstaand product te ontwikkelen. Hier is voor gekozen omdat het SFS niet wordt gezien als marketingtool. Door deze visie is het ook niet wenselijk dat het SFS wordt geïntegreerd in MailPlus of dat het SFS gebruik kan maken van data vanuit MailPlus.

De communicatie met het SFS gebeurt voornamelijk via de REST-API. Via deze API is het mogelijk om feedback verzoeken te versturen naar klanten, om gegeven feedback op te slaan en om statistieken uit het systeem op te vragen. Middels deze functionaliteiten is het

voor gebruikers van het systeem mogelijk om een eigen dashboard te ontwikkelen. De webpagina's voor het geven en bewerken van de feedback en schriftelijke reactie vallen binnen de opdracht. Om de complexiteit en omvang van de afstudeeropdracht beperkt te houden en om gebruikers meer vrijheid te geven is er, in overleg met de bedrijfsmentor, voor gekozen om het ontwikkelen van een dashboard buiten de scope van de opdracht te houden. Om meer dan alleen regels code te kunnen laten zien tijdens mijn verdediging zal er een eenvoudig dashboard worden ontwikkeld.

5. Risico's

Er zijn een aantal risicofactoren te benoemen die gedurende de uitvoering van de afstudeeropdracht zijn te verwachten. Allereerst het niet beschikken over voldoende kennis om de afstudeeropdracht tot een goed einde te brengen. Dit risico is te ondervangen door dit tijdig te herkennen en mijzelf in te lezen op het onderwerp en om de bedrijfsmentor in te schakelen. Het tijdig herkennen van het niet hebben van de juiste kennis kan zichtbaar worden doordat een onderdeel langer duurt dan gepland. Echter is ook mogelijk dat dit onderdeel complexer blijkt dan van tevoren is ingeschat. Een gevolg kan zijn dat het resultaat hieronder lijdt en/of de planning aangepast dient te worden zodat deze weer realistisch is.

Ten tweede tijdens de uitvoering van de afstudeeropdracht zou ik in tijdsnood kunnen komen. Dit risico kan worden beperkt door het maken van een degelijke planning en deze planning naargelang de voortgang aan te passen. Oorzaken kunnen zijn dat een onderdeel complexer is dan vooraf verwacht of de planning te optimistisch is. Door de voortgang te bewaken en hierop de planning aan te passen, verwacht ik dit risico te kunnen beperken.

Ten derde kan de afhankelijkheid van input van mijn bedrijfsmentor een knelpunt vormen tijdens de uitvoering van deze opdracht. Het is hierdoor van belang dat er vaste afspraken worden gemaakt waarop er gezamenlijk vergaderd kan worden over de voortgang en de input van de afstudeeropdracht. Om dit te ondervangen zijn er over geruime tijd vaste afspraken ingepland. Tevens heeft de bedrijfsmentor toegang tot de documenten die mijn voortgang vastleggen. Hierdoor is de bedrijfsmentor up to date over de voortgang van de opdracht.

6. Contact & rollen

Alle betrokken bij dit afstudeeronderzoek worden in onderstaande tabel genoemd.

Naam	Instantie	Email	Rol
Jos Verhaar		josverhaar@gmail.com jos@mailplus.nl	Afstudeerder
Arie Toet	De Haagse Hogeschool	a.j.toet@hhs.nl	Begeleidend examinator
Gerard Mijharends	De Haagse Hogeschool	g.a.mijnarends@hhs.nl	Tweede examiner
Paul Nederlof	Mailplus B.V.	paulne@mailplus.nl	Opdrachtgever
Niels Noorlander	Mailplus B.V.	niels@mailplus.nl	Bedrijfsmentor en product owner

7. Methodologie en planning

7.1 Methode

Als methode voor de verwerking van de afstudeeropdracht wordt gebruik gemaakt van Scrum. Op de development afdeling van Mailplus B.V. wordt gewerkt met Scrum. Om deze reden is ook tijdens deze opdracht gewerkt met de methodologie Scrum. Scrum is de manier van werken die de effectiviteit van teams verhoogt. Het voordeel van Scrum is dat projecten worden opgedeeld in korte sprints, waarin tussenproducten worden opgeleverd. Hierdoor is het zichtbaar waaraan gewerkt wordt en wat de voortgang is. Regelmatig overleg en feedback zorgt er tevens voor dat er een effectieve en intensieve samenwerking is wat resulteert in een effectievere aanpak van de opdracht (de Boer, Bruggink, Bruns et al., 2015).

Vanwege bovenstaande voordelen is daarom ook gekozen om Scrum te gebruiken tijdens deze afstudeeropdracht. Om het overzicht en de structuur te bewaren wordt door van Loon, Barendsen & Hoppenbrouwers (2015) aangeraden om gebruik te maken van een Scrum board. Dit wordt op zowel de development afdeling als tijdens deze opdracht digitaal bijgehouden middels Jira. Op dit Scrumboard worden de volgende drie kolommen bijgehouden; to do, in progress en done.

Een sprint duurt twee weken. Na deze twee weken wordt teruggekeken hoe de sprint is gegaan, de sprint retrospective, en welke producten er zijn opgeleverd. Tevens wordt er een planning en indeling gemaakt voor de volgende sprint.

7.2 Planning

	ma-w		do-z		ma-w		do-z		ma-w		do-z		ma-w		do-z		ma-w		do-z		ma-w		do-z	
Sprint #	0	0	1	1	1	1	2	2	VAK	VAK	2	2	3	3	3	3	3	3	3	3	3	3	3	3
Taak	38		39		40		41		42		43		44		45									
Inlezen huidige situatie																								
Schrijven Plan van aanpak																								
Huidige situatie inzichtelijk maken																								
Concurrenten onderzoek																								
Vaststellen (niet-) functionelen eisen																								
Ontwerp Database																								
Bedrijfsbezoek																								
Ontwerp Software																								
Ontwerp Tests																								
Coderen																								
Voortgangsverslag																								
Testen																								
Schrijven documentatie																								
Retrospective																								
Tussentijds assessment																								
Opbouw Afstudeerdossier																								
Overdracht																								
Inleveren Afstudeerdossier																								

Afbeelding 58: Planning sprint 0 t/m 3

	ma-wo		do-za		ma-wo		do-za		ma-wo		do-za		ma-wo		do-za		ma-wo		do-za		ma-wo		do-za		ma-wo		do-za	
Sprint #	4	4	4	4	5	5	5	5	6	6	6	6	VAK	VAK	VAK	VAK												
Taak	46				47				48				49				50				51				52			1
Inlezen huidige situatie																												
Schrijven Plan van aanpak																												
Huidige situatie inzichtelijk maken																												
Concurrenten onderzoek																												
Vaststellen (niet-) functionelen eisen																												
Ontwerp Database																												
Bedrijfsbezoek																												
Ontwerp Software																												
Ontwerp Tests																												
Coderen																												
Voortgangsverslag																												
Testen																												
Schrijven documentatie																												
Retrospective																												
Tussentijds assessment																												
Opbouw Afstudeerdossier																												
Overdracht																												
Inleveren Afstudeerdossier																												

Afbeelding 59: Planning sprint 4 t/m 6

8. Stakeholders

Tijdens de afstudeeropdracht zijn een aantal groepen stakeholders betrokken. Zij worden per groep gedefinieerd. In dit hoofdstuk zal worden aangegeven wie als vertegenwoordiger optreedt van deze groepen.

8.1 Groepen en hun vertegenwoordiging

8.1.1 Geïnteresseerde klanten

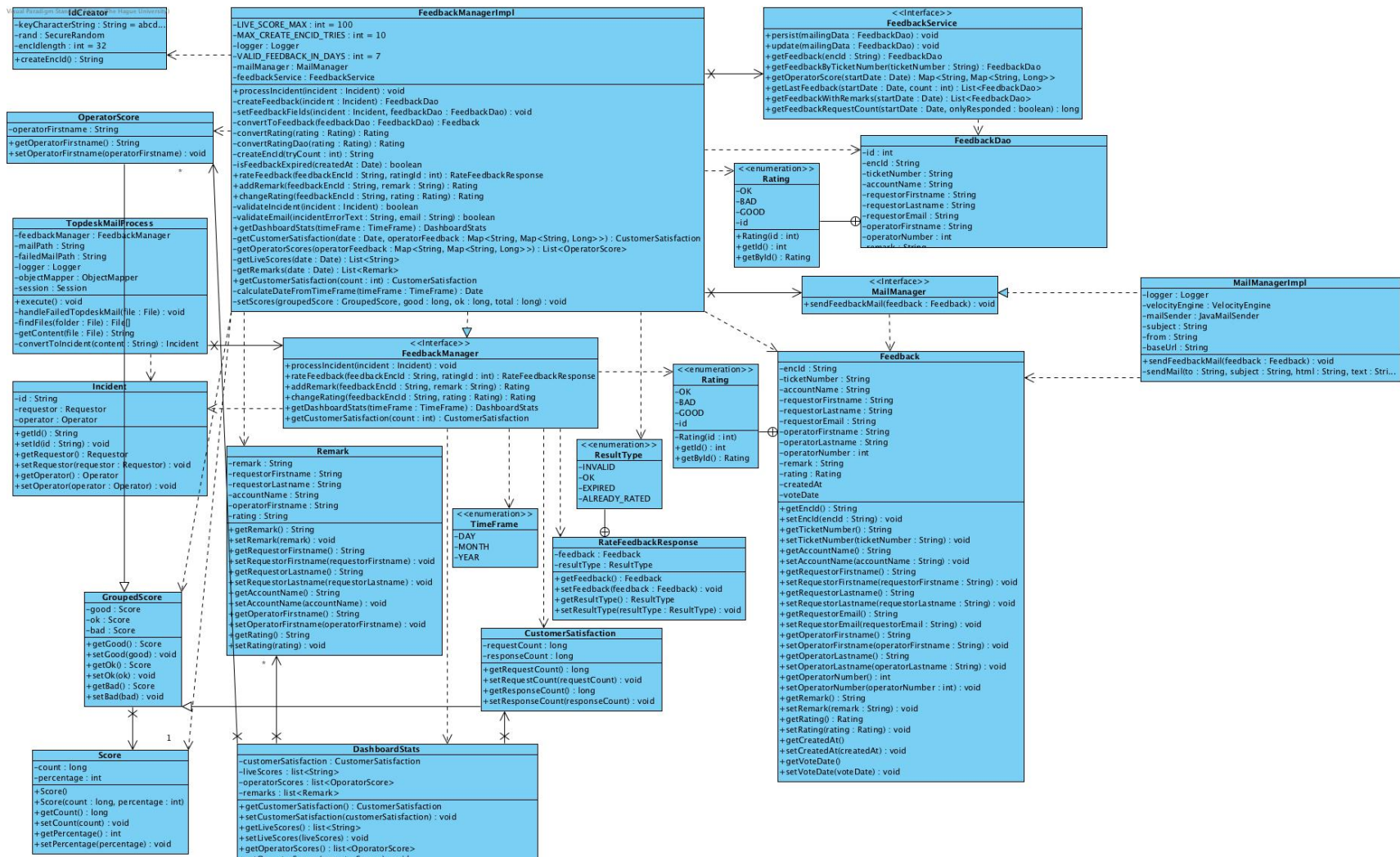
Deze groep bestaat uit klanten van Mailplus die interesse hebben getoond in het SFS. Deze personen zijn de toekomstige gebruikers van het SFS en zullen feedbackverzoeken gaan verzenden en statistieken gaan opvragen met het SFS. Deze groep zal vertegenwoordigd worden door Paul Nederlof. Hij heeft vanaf het in gebruik nemen van het SFS tot nu gesprekken gevoerd met geïnteresseerde klanten. Tijdens deze gesprekken heeft hij hun wensen en ideeën in kaart gebracht.

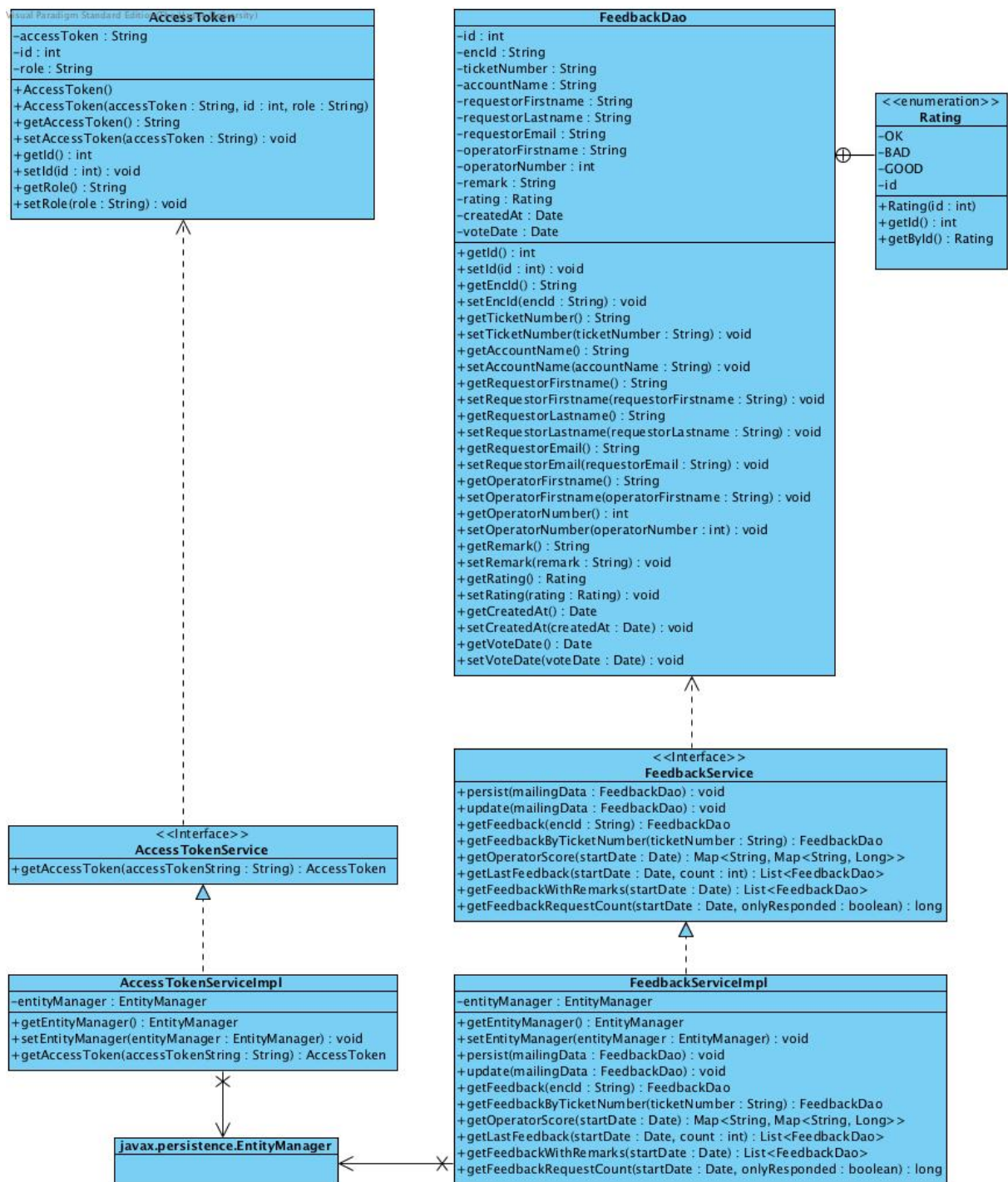
8.1.2 Huidige gebruikers

Met de huidige gebruikers worden de medewerkers van de afdeling support bij Mailplus B.V. bedoeld. Zij zijn de personen die dagelijks werken met het huidige SFS. Zij verzenden feedbackverzoeken en bekijken de statistieken die het dashboard toont. De huidige gebruikers worden vertegenwoordigd door de opdrachtgever. Hij is de direct leidinggevende van de afdeling support. Ook kunnen, wanneer nodig, andere medewerkers van de afdeling worden geraadpleegd.

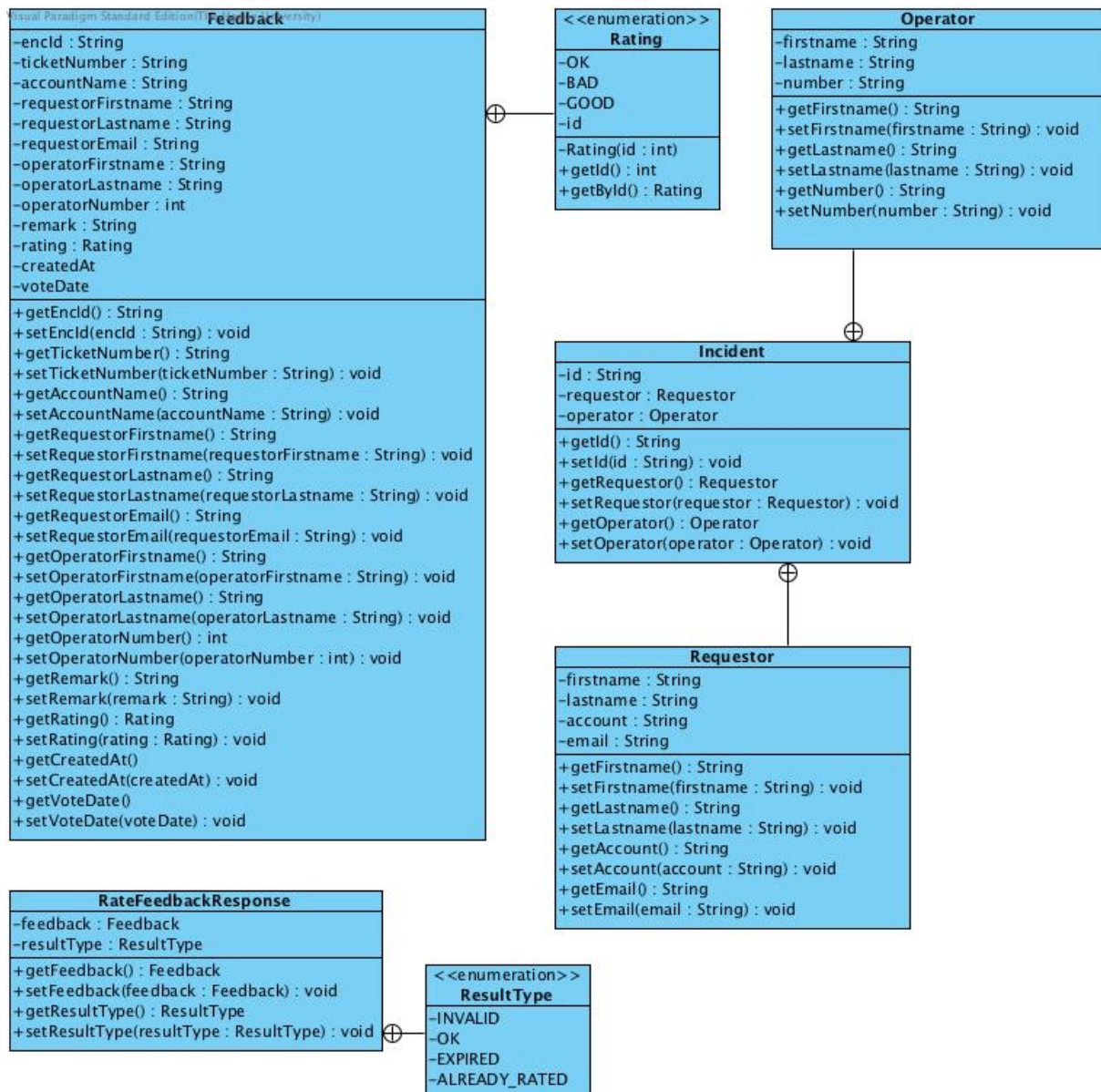
8.1.3 Product Management

Product Management is de afdeling binnen Mailplus B.V. die de requirements van het huidige SFS hebben bedacht. Zij zijn nauw betrokken geweest met het uitdenken en implementeren van de huidige versie van het SFS. Deze groep zal vertegenwoordigd worden door de bedrijfsmentor. Hij is de direct leidinggevende van deze afdeling en was zelf betrokken bij de ontwikkeling van het huidige SFS.



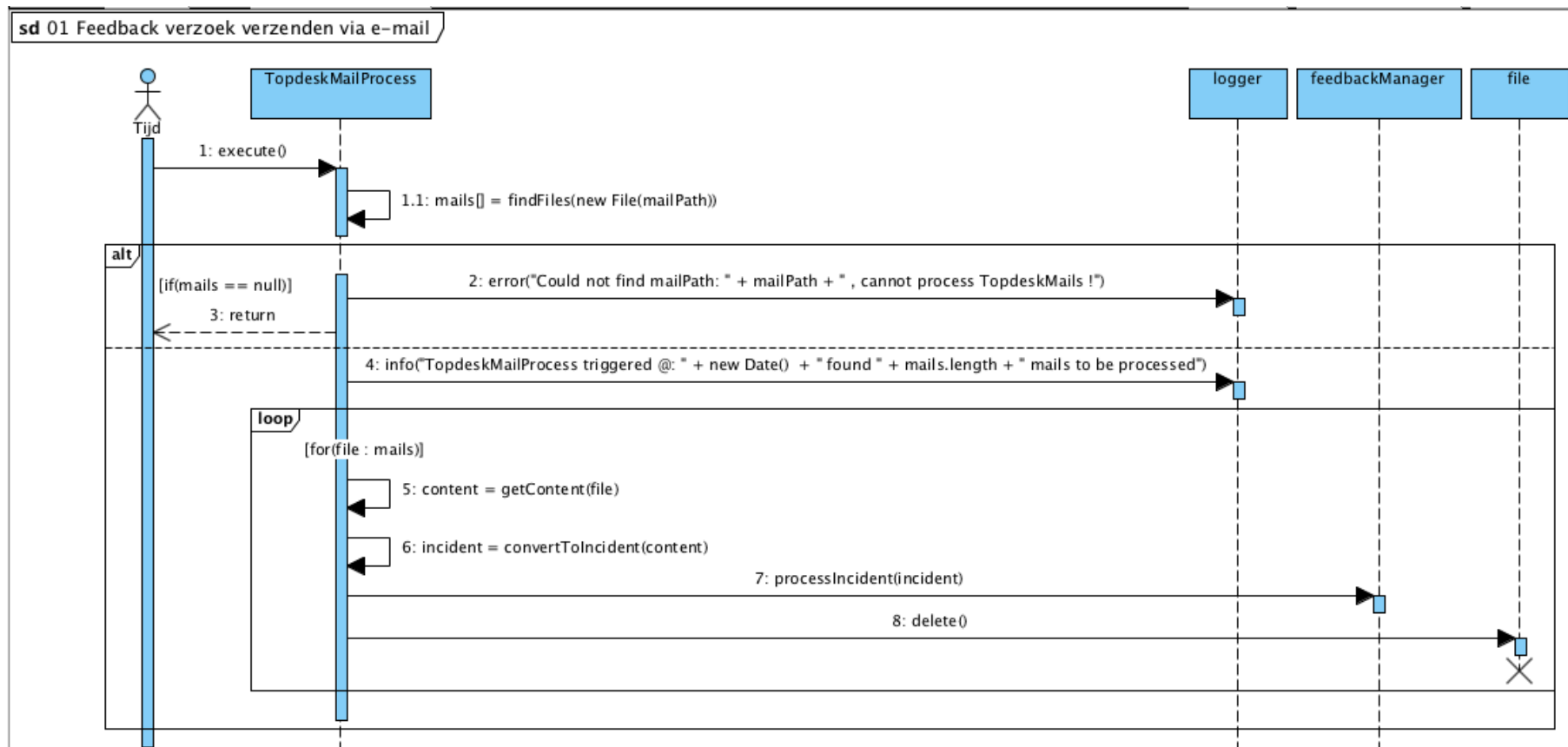


Afbeelding 45: Service klassen



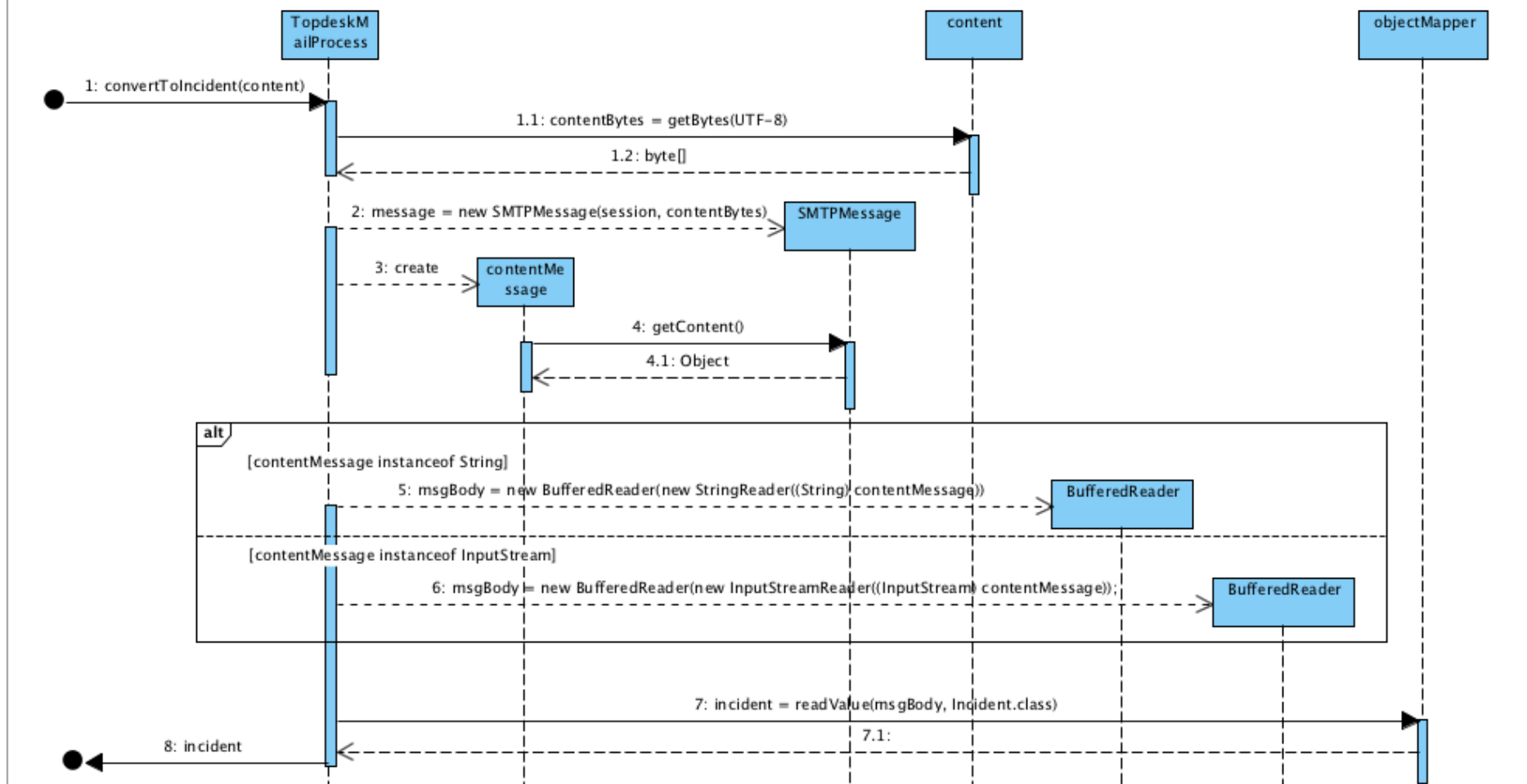
Afbeelding 46: Model klassen

Bijlage II: Sequentiediagrammen key-features huidige implementatie



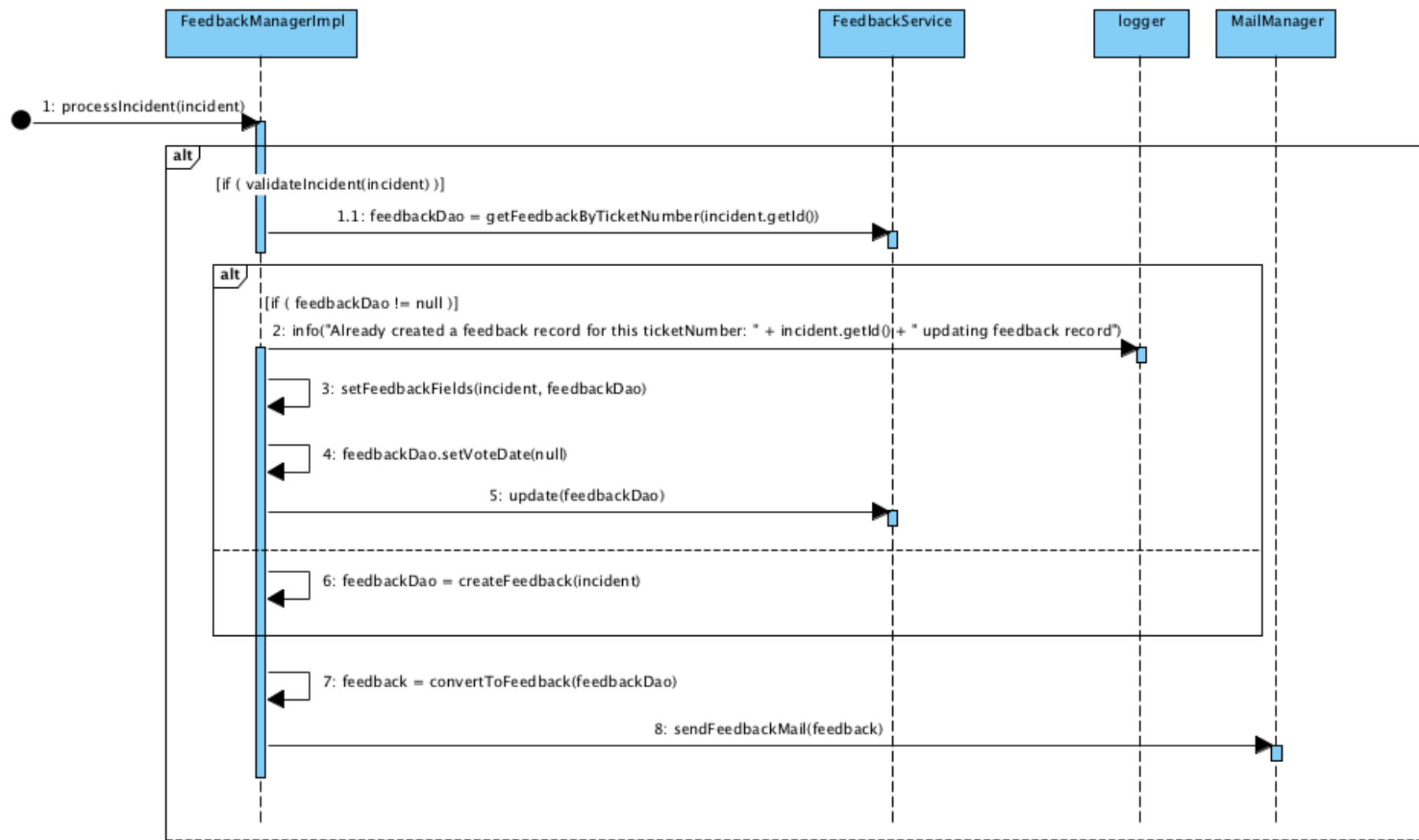
Afbeelding 47: Verwerking van feedback verzoek per mail

sd 01.1 Zet mail om naar incident



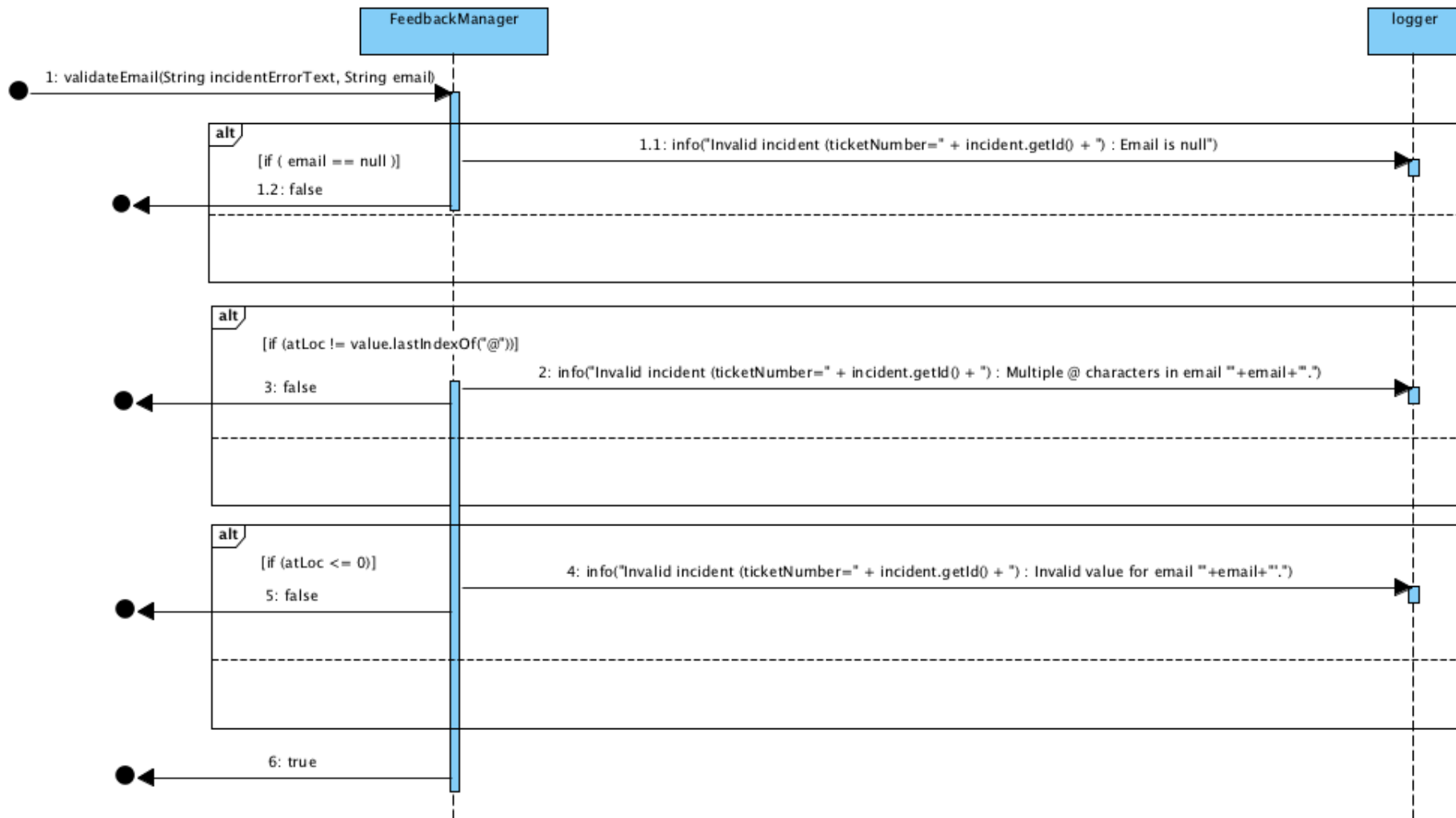
Afbeelding 48: Omzetten van e-mail naar Incident object

sd 01.2 Verwerk Incident



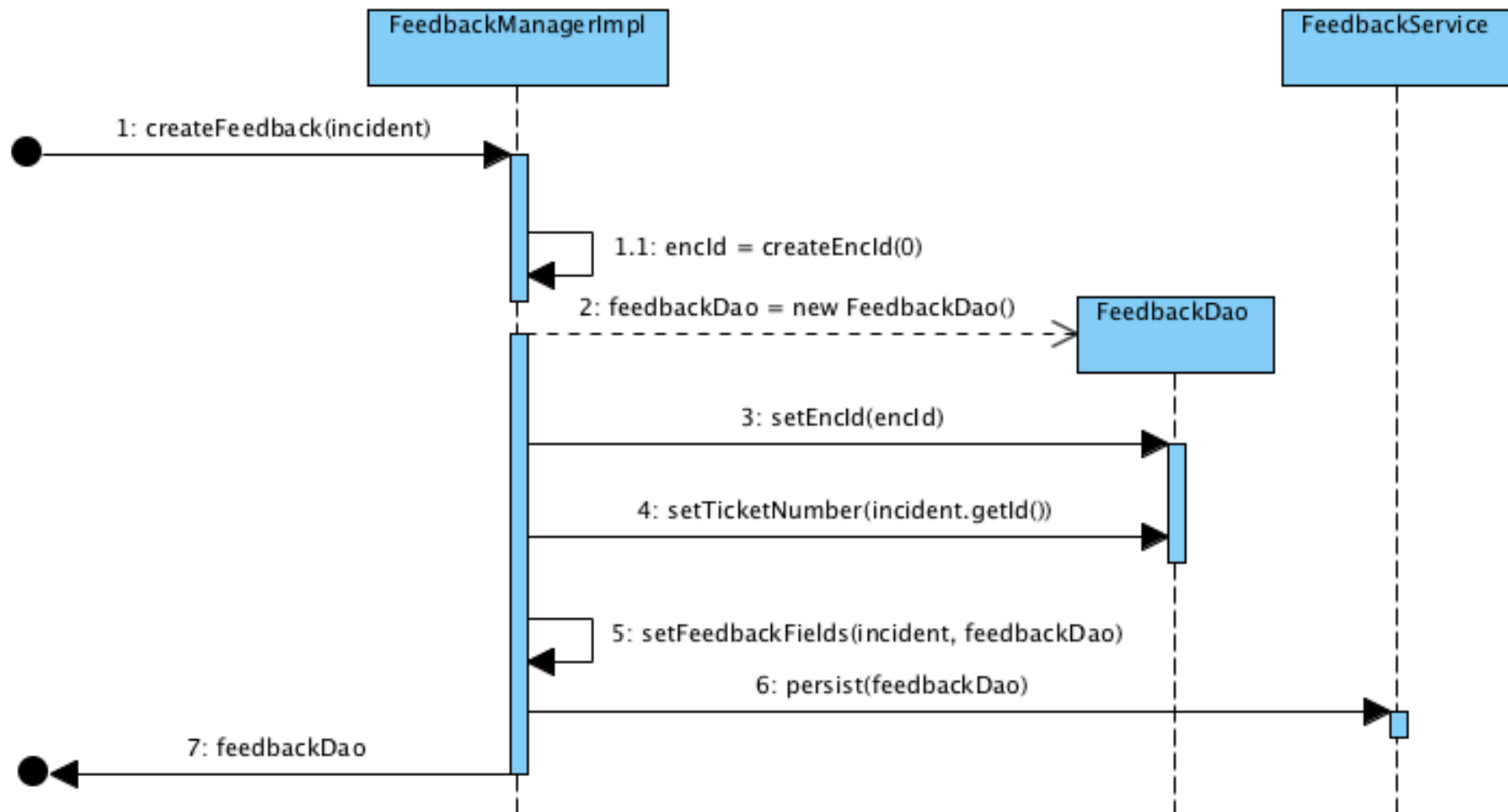
Afbeelding 49: Verwerken van Incident object tot feedbackmail

sd 01.2.1 Valideer incident



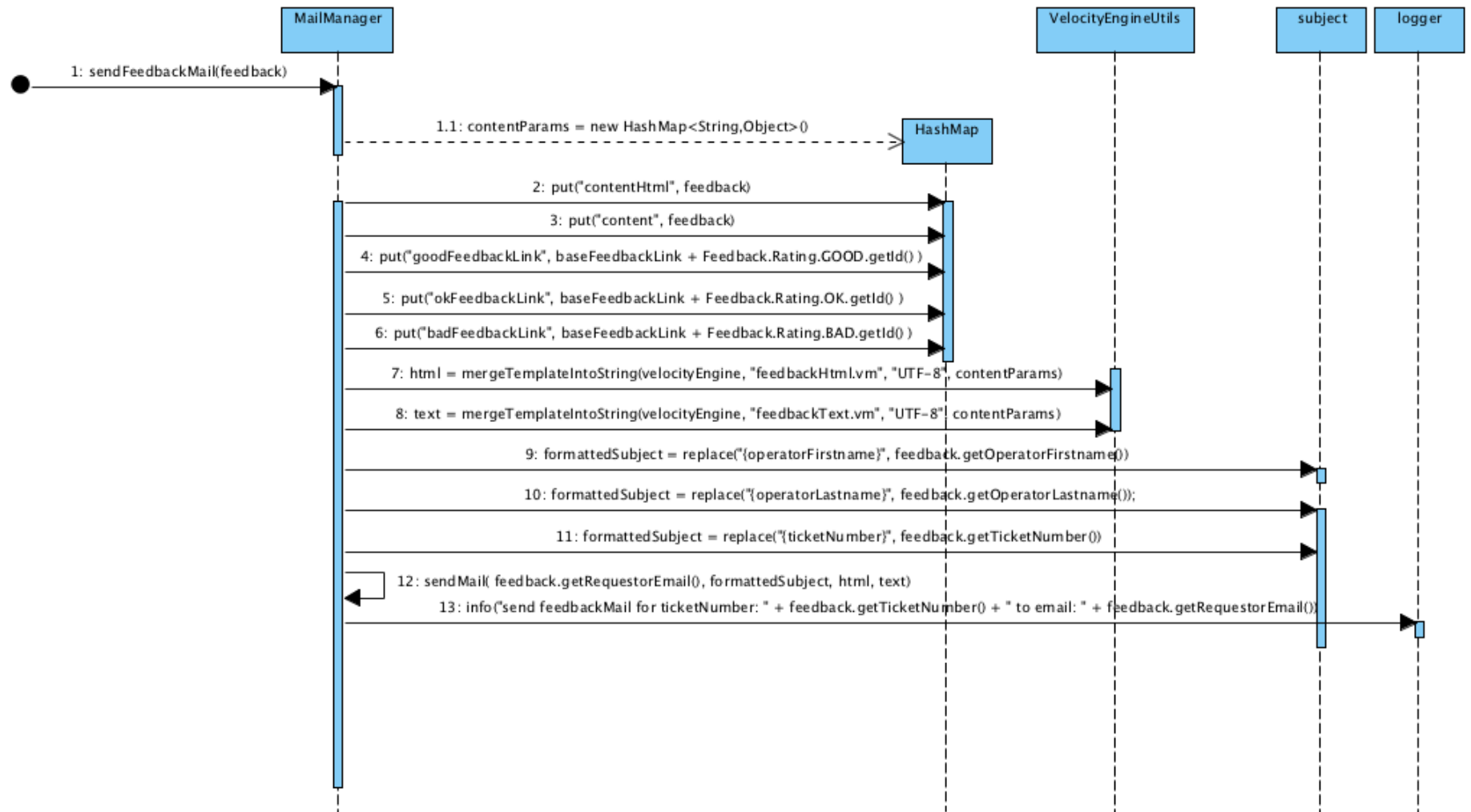
Afbeelding 50: Validatie van Incident object

sd 01.2.2 Creëer nieuw feedback record



Afbeelding 51: Omzetten van Incident object naar record in database

sd 01.2.3 Verzend feedback mail



Afbeelding 52: Detail van verzenden feedbackmail

Bijlage III: Huidige opzet

RRM huidige database

Primaire sleutel = attribuut_naam

Vreemde sleutel = *attribuut_naam*

AccessToken (id, accessToken, role)

Primaire sleutel: id

Feedback (id, accountName, createdAt, encId, operatorFirstname, operatorLastname, operatorNumber, rating, remark, requestorEmail, requestorFirstname, requestorLastname, ticketNumber, voteDate)

Primaire sleutel: id

Index: encId, ticketNumber, operatorNumber (operatorNumber, createdAt)

Bijlage IV: Het concurrentieonderzoek

Wat is het doel van het onderzoek?

Het doel van dit concurrentieonderzoek is het achterhalen welke functionaliteiten andere systemen bieden voor het uitvragen en analyseren van feedback. In het bijzonder feedback over de tevredenheid van klanten kort na een geleverde dienst.

Waarom wordt dit onderzocht?

De uitkomsten van dit onderzoeken kunnen input leveren voor de functionaliteiten van het te ontwikkelen feedback systeem. Welke functionaliteiten maken het dat klanten een bepaald systeem gebruiken? Welke goede functionaliteiten kunnen wij overnemen?

Hoe wordt dit onderzocht?

Aan de hand van vijf vragen over de werking, mogelijkheden en functionaliteiten van de feedback systemen worden de overkomsten en verschillen tussen verschillende feedback systemen op de markt in kaart gebracht.

Op welke manier wordt de feedback uitgevraagd?

Wordt dit veelal gedaan door één vraag te stellen? Of wordt er gebruik gemaakt van een vragenlijst bestaande uit meerdere vragen? Welke vraag- en antwoordtypen worden ondersteund?

Op welke manier(en) kan het feedbackverzoek worden verzonden?

Hoe komt het verzoek tot feedback bij de klant terecht? Gebeurt dit veelal via e-mail of zijn er ook concurrenten die gebruik maken van modernere technieken zoals sociale media?

Welke mogelijkheden tot analyse van de data bieden de andere systemen?

Hebben deze systemen kant en klare dashboards en is het mogelijk om deze naar wens aan te passen? Is het mogelijk om statistieken te exporteren, zo ja hoe?

Hebben de andere systemen slimme en/of extra features?

Bieden concurrenten slimme features aan waar Mailplus B.V. nog niet aan heeft gedacht?

Welke mogelijkheden tot integratie bieden de andere systemen?

Kan je het systeem koppelen met andere systemen zodat de statistieken in een ander systeem getoond kunnen worden? Is het mogelijk om je CRM-pakket er aan te koppelen zodat je gebruik kunt maken van deze data (e-mailadressen, naam, etc.)? Bieden concurrenten een API aan waarmee data opgehaald kan worden uit het systeem?

Door het beantwoorden van bovenstaande vragen kan een goed beeld worden verkregen van de andere feedback systemen op de markt. Door het in kaart brengen van de antwoorden komen er wellicht nog goede features naar boven die overgenomen kunnen worden in het te ontwikkelen feedback systeem.

Hoe worden concurrenten gevonden?

Doormiddel van het gebruik van de zoekmachine Google en verschillende zoektermen wordt verwacht eenvoudig een aantal feedback systemen te kunnen vinden. In nieuwsbrieven van grote bedrijven zoals: Conrad.nl, Bol.com, de Consumentenbond en Coolblue bevindt zich tegenwoordig de mogelijkheid tot het geven van feedback op de nieuwsbrief. Doordat deze manier van feedback gaat over de mailing die wordt verzonden en niet over een geleverde dienst, vallen deze bedrijven buiten de scope van dit onderzoek. Het gaat hier om de volgende twee bedrijven: usabilla.com en closealert.com.

Gebruikte zoekwoorden

De bedrijven en artikelen die op de eerste pagina werden genoemd en welke uniek zijn voor dit onderzoek zijn opgenomen als potentiële concurrenten. Met de volgende zoekwoorden is gezocht met Google:

- klantfeedback
- continue klantfeedback
- klant review systeem
- klant reviews beheren
- klant feedback
- customer feedback tool
- continue CES meten
- customer CES

Gevonden artikelen en bedrijven

Doormiddel van bovengenoemde zoekwoorden is een grote lijst met bedrijven en artikelen ontstaan:

- Storesupport	- Feedbackcompany
- Cysklantfeedback	- vrijdag online (artikel)
- Expoints	- Trustpilot
- Feeddex	- Custorate
- Marketingfacts (artikel)	- Asknicely
- Integron	- Clientheartbeat (artikel)
- Louwers en Partners	- Typeform
- Truqu	- Help Scout
- Groo	- Intercom
- Klantenvertellen.nl	- E-trusted
- Klantok	- Surveymonkey
- Kiyoh	- KCM
- Klantscores.nl	- Checkmarket (artikel)
- Klantenschrijven.nl	
- Thuiswinkel.nl	

Verfijnen van concurrenten

Uit de zoekactie zijn 24 systemen en 4 artikelen naar voren gekomen. De gevonden systemen zijn niet allemaal geschikt om verder te onderzoeken. Na deze inventarisatie zijn verschillende criteria opgesteld om de lijst verder te kunnen verfijnen. Deze criteria zijn:

- Het gaat om het uitvragen van klantfeedback over processen/ diensten;
- Het gaat niet over e-commerce producten/ ondersteuning;
- Het dient een eenvoudige opzet te hebben, laagdrempelig (De opdrachtgever verlangt dat de werking eenvoudig is);
- Het dient een Nederlands bedrijf te zijn;
- Het dient geen KTO te worden;
- Real time data in kunnen zien.

Deze criteria zijn tot stand gekomen door de huidige en toekomstige opzet van het te ontwikkelen systeem voor ogen te houden. Dit systeem biedt de mogelijkheid om feedback te vragen aan klanten na het leveren van een dienst, bijvoorbeeld een support vraag. Het systeem is niet bedoeld voor het geven van een waardering over producten en/of een bedrijf. Het geven van feedback dient eenvoudig te zijn en niet veel tijd in beslag te nemen. Ook hoeft het geen vragenlijst te zijn. De statistieken dienen real-time inzichtelijk te zijn, zoals nu. Omdat Mailplus B.V. zich voornamelijk richt op de Nederlandse markt dienen de systemen door Nederlandse bedrijven aangeboden te worden.

Na het toepassen van deze selectie criteria blijven de volgende bedrijven over, welke verder onderzocht zullen worden:

- Cysklantfeedback
- Expoints
- Feeddex
- Truqu
- KCM

Analyse methode

Omdat dit onderzoek minder gestructureerde data heeft opgeleverd en omdat het aantal concurrenten laag is, is er voor gekozen om de data te analyseren volgens horizontaal vergelijken.

Deze methode is uitermate geschikt voor het analyseren van minder gestructureerde data. In een tabelvorm worden alle antwoorden per feedback systeem op een rij gezet. In de laatste kolom staan de bevindingen. Deze bevindingen kunnen bestaan uit overeenkomsten en/of verschillen tussen de systemen.

	Cysklantfeedback	Expoints	Feeddex	Truqu	KCM	Bevindingen
<p><u>Vraag 1:</u> Op welke manier wordt de feedback uitgevraagd?</p> <p><u>Waarom:</u> Om te bepalen of de huidige manier van uitvragen uniek/afdoende is, is het van belang te weten wat de mogelijkheden zijn met andere systemen.</p>	<p>Uitvragen van feedback gebeurt door <u>4 simpele vragen</u>, welke ook nog eens worden getoond op onderdelen van het bedrijf. Onderdelen waar een klant niet mee te maken heeft gehad, komen ook niet terug in de vragenlijst. De vragen kunnen een cijfer zijn <u>(0-10) of open antwoorden</u></p>	<p>Expoints biedt verschillende vormen van clustering, specifieke condities, variabele antwoordcategorieën en <u>-schalen</u>. Ook kan er gebruik worden gemaakt van <u>open- en gesloten vragen</u> eventueel aangevuld met categorisering.</p>	<p>Uitvragen van feedback voor een medewerker is onderdeel van het geheel. Continue kun je de <u>CES en NPS</u> uitvragen. De CES werkt met een <u>11-punts schaal (0-10)</u> en de NPS met een <u>5-punts schaal (1-5)</u> uitvragen.</p>	<p>Het is een manier van 1-op-veel. <u>Feedback over verschillende onderwerpen</u>. Vragen kunnen onder andere bestaan uit: <u>openvragen, meerkeuze, keuze vragen, ratings.</u></p>	<p>Dit pakket kan meerdere vormen uitvragen. Ze bieden o.a. klanttevredenheid, medewerkerstevredenheid, NPS, <u>CES, Net Effort Score (NES)</u> en Customer Journey. <u>Alle mogelijke vraagtypen worden aangeboden.</u></p>	<p>-) zowel open als gesloten vragen zijn mogelijk -) Antwoordschalen worden veel gebruikt -) Tonen van Kengetallen/ KPI's -) Meerdere vragen, nooit een vraag</p>
<p><u>Vraag 2:</u> Op welke manier(en) kan het feedbackverzoek worden verzonden?</p> <p><u>Waarom:</u> Om inzicht te krijgen in andere mogelijkheden tot het versturen van een feedbackverzoek.</p>	<p>Per <u>mail</u> en via <u>SMS</u></p>	<p>Na het ontvangen van een data-file door Expoints wordt de feedback per <u>mail</u> verzonden. Op eigen initiatief van de klant is mogelijk via een extra module</p>	<p>Via online, mobile en <u>sociale media</u> methoden.</p>	<p>Continue via een <u>app</u>, via <u>links, QR-codes, sociale media</u> en per <u>mail</u> worden de verzoeken verzonden.</p>	<p>Per <u>e-mail, sms, telefonisch, touchscreens en QR-codes</u></p>	<p>-) Per e-mail -) Per sms -) Per URL -) Sociale media -) QR-codes</p>
<p><u>Vraag 3:</u> Welke</p>	<p>Uitgebreide</p>	<p>Real-time dashboard met</p>	<p>Uitgebreid</p>	<p>Analyses op</p>	<p>Niet duidelijke voorbeelden,</p>	<p>-) eigen dashboard</p>

<p>mogelijkheden tot analyse van de data bieden de andere systemen?</p> <p><u>Waarom:</u> Ik wil weten welke analyse mogelijkheden andere systemen bieden om een beeld te krijgen van de mogelijkheden.</p>	<p>analyse mogelijkheden met dashboard dat te personaliseren is. Prioriteiten Matrices die op basis van de KPI's en belang worden weergegeven. Mogelijkheid tot het tonen van rankings in het dashboard</p>	<p>verschillende tabbladen. 'Homepage' met belangrijke zaken waar de organisatie op wil sturen, 'Interactie module' waar alle feedback inzichtelijk wordt gemaakt en een 'rapportage set' op vrijwel iedere mogelijkheid (grafisch, tabel) om de feedback weer te geven.</p>	<p>real-time dashboard met kengetallen, grafieken en trends</p>	<p>verschillende niveau's. Per vragenlijst. Feedback per type, Requested Feedback, Pending Requests, Requests per type, Active Users, Requests per Group.</p>	<p>maar in het dashboard zijn hiervoor mogelijkheden. Belangrijke KPI's worden direct getoond. Per survey, onderzoek zijn overzichten en verdiepingen/ analyses mogelijk.</p>	<p>-) Tonen van kengetallen/ KPI's</p>
<p><u>Vraag 4:</u> Hebben de andere systemen slimme en/of extra features?</p> <p><u>Waarom:</u> Ik wil weten wat andere systemen voor slimme features hebben waarmee ze zich onderscheiden van de concurrenten.</p>	<p>Veel herhalende handelingen worden geautomatiseerd door slimme instellingen. <u>Rollen die op verschillend niveau naar de data kijken.</u> Automatisch verzenden van rapportages binnen de eigen organisatie. Mogelijkheid tot het ontvangen van <u>push mails</u> wanneer er</p>	<p><u>Opt-out</u> mogelijkheid voor ontvangers (niet meer feedback ontvangen) Mogelijkheid tot <u>feedback-block</u> (bijv. periode van 3mnd geen feedback versturen) Mogelijkheid tot herinnering <u>Eigen tone of voice en look and feel is mogelijk</u></p>	<p><u>E-mail alerts</u> bij lage NPS en/ of wanneer klant contact wenst</p>	<p>Real-time. Je kunt complimenten geven aan anderen (thumbs-up, fles champagne). <u>Werken met rollen en rechten systeem Nederlands en Engels</u> mogelijk Teams zijn mogelijk (geven en krijgen van</p>	<p>ISO 27001 standaard</p>	<p>-) Eigen look en feel -) Eigen tone of voice -) Beperken van versturen feedback verzoeken (time-out/ opt-out) -) E-mail alerts -) Closed loop feedback principe -) Rollen en rechten systeem -) Verschillende talen ondersteunen</p>

	<p>bepaalde antwoorden zijn gegeven.</p> <p>Ingebouwd ticket systeem op basis van ingevulde antwoorden op feedback verzoeken.</p> <p>Feedback kan na het oplossen van een <u>ontevredenheid</u> opnieuw worden verzonden (closed loop feedback principe)</p> <p><u>Eigen look and feel van de klant</u> (opdrachtgever)</p>			<p>feedabck)</p> <p>Anoniem is mogelijk. <u>Eigen look and feel is mogelijk.</u></p>		
<p><u>Vraag 5:</u> Welke mogelijkheden tot integratie bieden de andere systemen?</p> <p><u>Waarom:</u> Ik wil deze integratie mogelijkheden weten zodat wij dit kunnen meenemen in het systeem.</p>	<p>Integratie mogelijk met iOS en Android devices d.m.v. een app.</p> <p>Integraties mogelijk met verschillende review sites.</p>	<p>Er zijn mogelijkheden tot koppelingen met CRM-, rapportage- en administratiepakketten.</p> <p>Feedback is te exporteren naar Word, Excel of PDF.</p>		<p>Export naar Excel en CSV zijn mogelijk</p>	<p>Koppelen is mogelijk met veel CRM- en Klantsoftware systemen. Er zijn bestaande koppelingen voor AFAS en KANA</p>	<p>-) Export csv en PDF</p> <p>-) Koppeling met externe systemen (API)</p>

Bevindingen

Uit het concurrenten onderzoek is gebleken dat de onderzochte systemen veelal dezelfde functionaliteiten bieden. Alle systemen beschikken over de mogelijkheid om meerdere vragen te stellen en in meerdere vormen (open-, gesloten vragen en antwoordschalen). Het verzoek om feedback te geven wordt veelal via e-mail verzonden. Cysklantfeedback en KCM bieden ook de mogelijkheid tot het versturen van een link via SMS. Truqu en KCM werken ook met de optie om QR-codes in te zetten. Feeddex en KCM bieden als enige de mogelijkheid tot het gebruik van Social Media.

Alle bekeken systemen bieden een dashboard waarin statistieken worden getoond. Sommige systemen bieden meer functionaliteiten in het dashboard dan andere. Zo biedt Cysklantfeedback een geïntegreerd ticket systeem waarmee klant contact onderhouden kan worden. Truqu biedt ook statistieken over gegeven feedback per type, verzonden feedback, uitstaande feedback verzoeken, verzoeken per type en verzoeken per groep.

Alle systemen die onderzocht zijn bieden een vorm van het aanpassen van de 'look and feel' en 'tone of voice'. Verder zijn er nog een paar slimme features die opvallen. Zo biedt Cysklantfeedback het 'closed loop feedback principe' aan. Dit houdt in dat een ontevreden waardering opnieuw verzonden kan worden. Dit kan bijvoorbeeld worden gedaan nadat er contact met de klant is geweest. Expoints biedt een 'feedback lock' en een 'opt-out' mogelijkheid voor de klanten. Hiermee kan er tijdelijk (feedback lock) of permanent (opt-out) een block worden gezet op het versturen van feedback verzoeken aan klanten. Feeddex en Cysklantfeedback kunnen meldingen geven aan de gebruiker bij een bepaalde waardering of wanneer een klant contact wenst. Bij Truqu en Cysklantfeedback kun je werken met rechten en rollen. Hierbij hebben verschillende rollen andere rechten. Truqu is het enige systeem dat heeft aangegeven dat het zowel in het Nederlands als in het Engels gebruikt kan worden. Alle systemen hebben een mogelijkheid tot het koppelen met externe systemen en/of het downloaden van statistieken als Excel- (csv) of PDF bestand.

Conclusie

Uit het onderzoek blijkt dat de onderzochte systemen ieder een eigen idee hebben van het uitvragen van klant feedback. Het ene systeem is nog uitgebreider dan het andere. Het ene bedrijf gooit met mooie termen zoals NPS, CES, Customer Journey, enzovoort. Hieruit blijkt dat de onderzochte systemen uitgebreider en specialistischer zijn dan het systeem dat Mailplus B.V. voor ogen heeft.

Er kan geconcludeerd worden dat alle systemen de mogelijkheid bieden tot het uitvragen van meerdere vragen met meerdere antwoord mogelijkheden. Wanneer er gebruik wordt gemaakt van puntenschalen als antwoordmogelijkheid zien we veelal 11-punts- en 5-punts schalen. Deze bevinding uit het concurrentie onderzoek is besproken met de opdrachtgever. Hier is uitgekomen dat Mailplus B.V. graag vasthoudt aan deze ene vraag met de 3-punts schaal. Hiervoor is gekozen omdat het systeem zoals het nu is, erg laagdrempelig is voor ontvangers om te gebruiken. Het enige dat ze hoeven te doen is het klikken op één van de drie smiley iconen en optioneel een opmerking te geven. Mede hierdoor is er een hoog response gehalte. Ook is het lastig om een 11-punts schaal (0-10) in te zetten. Dit komt doordat een 7 voor de ene persoon goed is en voor een ander persoon zeer goed. Hierdoor is het lastig te classificeren wat een gemiddeld cijfer 7 nu precies betekent.

Het versturen van het feedback verzoek doen de onderzochte systemen veelal op de zelfde manier, via e-mail. Er zijn uitzonderingen die gebruik maken van andere technieken

zoals sociale media. Het huidige systeem kan alleen via e-mail verzoeken versturen. Voor het nieuwe systeem kan het wenselijk zijn te kijken naar mogelijke uitbereidingen hierop.

De systemen hebben alleen een uitgebreid dashboard tot hun beschikking waarin de statistieken op alle mogelijke manieren getoond en met elkaar vergeleken kunnen worden. Het huidige systeem heeft een eigen gemaakt dashboard dat zichtbaar is voor de afdeling support. Een dashboard is een mogelijkheid, maar in overleg met de opdrachtgever is er besloten om dit in de eerste versie achter wegen te laten. Hier is voor gekozen om de opdracht niet onnodig ingewikkeld te maken. Om statistieken aan te bieden komt er wel een REST-API waarmee klanten deze kunnen uitvragen en kunnen tonen op de manier dat hen schikt.

In het huidige systeem is het niet mogelijk om feedback verzoeken niet te laten versturen naar klanten die deze liever niet ontvangen. Wanneer het systeem een aanvraag krijgt, wordt deze altijd verzonden. Sommige klanten hebben echter aangegeven deze niet te willen ontvangen, of dat ze van mening zijn dat ze er teveel ontvangen. Uit het onderzoek is gebleken dat er systemen op de markt zijn die hier een oplossing voor hebben. Dit is een voorbeeld van een slimme feature die mogelijk overgenomen wordt.

Bijlage V: Requirements

Business Requirements

Nummer	Omschrijving
BR01	De opdrachtgever wil haar klanten de mogelijkheid bieden om doorlopend en op eenvoudige wijze de klanttevredenheid te meten van hun eigen klanten.
BR02	De opdrachtgever wil graag een nieuwe klantgroep aanboren met het nieuwe systeem.
BR03	De opdrachtgever wil dat het systeem minder afhankelijk is van releases en updates.
BR04	De opdrachtgever wil beheer mogelijkheden in de tool zelf.
BR05	De opdrachtgever wil meer statistieken uit het systeem kunnen halen.
BR06	De opdrachtgever wil meer mogelijkheden met betrekking tot het verzenden van een feedback mail.
BR07	De klant wil de mogelijkheid tot het aanpassen van de te gebruiken teksten in het systeem.
BR08	De klant wil de mogelijkheid tot het voeren van meerdere talen.

User requirements gebaseerd op geïdentificeerde belanghebbende

Nummer	Omschrijving	Belanghebbende	Bron
UR01	Dag/Maand/Jaar rapportage op monitor is niet duidelijk. De overgang van maand naar jaar is te subtiel (alleen de cijfers veranderen en er zijn nog geen grafieken). De rapportage verbeteren en uitbreiden met eerdere resultaten.	Teammanager Support	Requirements Paul Nederlof.docx
UR02	Toevoegen operator/behandelaar binnen het proces. Nieuwe medewerkers moeten ingediend worden via TOPdesk en door escalatie naar 2de lijns worden deze door development opgepakt.	Teammanager Support	Requirements Paul Nederlof.docx
UR03	Aanpassingen aan de uitnodiging (html e-mail) moet worden ingepland met de eerstvolgende release (3-4 weken). Het	Teammanager Support	Requirements Paul Nederlof.docx

	beheer van de e-mail zouden wij liever bij onszelf willen houden		
UR04	Korte omschrijving van call toevoegen in de uitnodiging. Als een klant vaker heeft gesproken met Support kan een korte omschrijving meer duidelijkheid geven waar het gesprek over ging. De korte omschrijving wordt opgehaald van het veld “Korte omschrijving (Details)” uit TOPdesk en is een verplicht veld tijdens registratie. De waarde van het veld moet ergens in de mail geplaatst worden.	Teammanager Support	Requirements Paul Nederlof.docx
UR05	<p>“Ik ontvang jullie feedback te vaak”. Soms kan het de klant ook iets teveel worden... Tussen die duizenden contacten die wij hebben uitgenodigd zitten er soms gebruikers tussen die niet zitten te wachten op een uitnodiging. Ze krijgen het te vaak zeggen ze. Dat is bij te houden in TOPdesk. Echter TOPdesk kan ons niet helpen om dan ook geen mail te sturen of die herinnering te negeren. We kunnen namelijk een melding maken bij de gebruikersgegevens maar dat valt niet altijd op. Een blacklist in de database (handmatig toegevoegd door iemand met db toegang) zou irritatie bij de klant kunnen voorkomen.</p> <p>Wij kunnen dan de uitnodiging sturen maar de laatste check vind plaats in de database. Mocht het adres voorkomen in de blacklist, dan wordt de uitnodiging niet verstuurd. Met hierbij de voorkeur dat in de rapportage ergens staat dat wij het wel zouden hebben verstuurd zodat de cijfers blijven kloppen.</p>	Teammanager Support	Requirements Paul Nederlof.docx
UR06	<p>Het tijdstip van verzenden van de getriggerde uitnodiging:</p> <p>a) moet op een vast tijdstip mogelijk zijn. Bijvoorbeeld om 20:00 's avonds naar alle contacten van die dag.</p> <p>b) met een vertraging van een xx minuten</p>	Teammanager Support	Requirements Paul Nederlof.docx
UR07	De klant kan gebruikers/behandelaars beheren. (Toevoegen, verwijderen, wijzigen)	Geïnteresseerde klanten	Requirements Paul Nederlof.docx

UR08	Het systeem kan met extra velden omgaan. M.a.w. toe te voegen in het systeem, vullen en verwerken in de mail. (Net als een databaseveld aanmaken, waarden importeren naar die velden en gebruiken in een mail als {[veldnaam]}))	Geïnteresseerde klanten	Requirements Paul Nederlof.docx
UR09	De feedback kan beheerd worden. De smileys mogen een andere naam krijgen. Zo kan je ook een andere taal kiezen (Good, Ok, Bad)	Geïnteresseerde klanten	Requirements Paul Nederlof.docx
UR10	De klant heeft de uitnodiging/mail in een omgeving waar zij zelf bij kunnen of waar een contactpersoon (of supportafdeling) bij kan voor aanpassingen	Geïnteresseerde klanten	Requirements Paul Nederlof.docx
UR11	De klant heeft de keuze om in stellen dat vooraf bepaald kan worden of de mail wel/niet verstuurd wordt en de keuze of het systeem in staat is om willekeurig de uitnodiging te versturen	Geïnteresseerde klanten	Requirements Paul Nederlof.docx
UR12	Bij een slechte beoordeling kan een mail worden gestuurd naar de behandelaar (en eventueel kopie naar de afdelingsmanager)	Geïnteresseerde klanten	Requirements Paul Nederlof.docx
UR13	Lay-out beheer (logo, kleurtjes)	Geïnteresseerde klanten	Requirements Paul Nederlof.docx
UR14	Het systeem kan een blacklist bijhouden of rekening houden met afmelders	Geïnteresseerde klanten	Requirements Paul Nederlof.docx
UR15	Rapportage vanuit het systeem of in een tool waar de standaard rapportage bijgehouden wordt. Opties zoals download naar pdf, rapportage mailen naar een e-mailadres	Geïnteresseerde klanten	Requirements Paul Nederlof.docx
UR16	Maandelijkse stats naar de beheerder en/of een lijst van e-mailadressen	Geïnteresseerde klanten	Requirements Paul Nederlof.docx
UR17	Beveiliging in het systeem om eenmalig feedback te kunnen geven en met controle om binnen (..) uur te reageren	Geïnteresseerde klanten	Requirements Paul Nederlof.docx

UR18	Landingspagina na feedback (toelichting/motivatie) is te beheren via de tool.	Geïnteresseerde klanten	Requirements Paul Nederlof.docx
UR19	Via de Rest-API is het mogelijk om de gegevens op te vragen van het feedbacksysteem	Geïnteresseerde klanten	Requirements Paul Nederlof.docx
UR20	Er is een pagina beschikbaar waarin de laatste 100 reacties bijgehouden worden. Deze kan geëxporteerd en geïmplementeerd worden op een externe pagina. Logo van Mailplus erbij zodat geïnteresseerden kunnen doorklikken naar onze website.	Geïnteresseerde klanten	Requirements Paul Nederlof.docx
UR21	Het tonen van het procentpunt verschil in good/ok/bad percentages in vergelijking met vorige periode (dag, maand, jaar)	Medewerker support	Features Sander van 't Hof
UR22	Het meesturen van de bestede tijd van een call. De waarde van de gemiddelde bestede tijd of totale bestede tijd moet getoond kunnen worden in dag/maand/jaar en hoort bij de algemene cijfers zoals Aantal feedback verstuurd en Aantal gereageerd.	Medewerker support	Features Sander van 't Hof
UR23	Het tonen van <u>Highcharts</u> grafieken met daarin percentage good/ok/bad in vergelijking met vorige periode (dag, maand, jaar)	Medewerker support	Features Sander van 't Hof
UR24	Het tonen van de naam/bedrijf/callnummer bij de betreffende smileys	Medewerker support	Features Sander van 't Hof
UR25	Totalen feedback weergeven per dag/maand/jaar. Niet alleen de percentages maar ook de absolute getallen	Medewerker support	Features Sander van 't Hof
UR26	Een beoordelaar mag één maal zijn/haar stem uitbrengen per Feedback verzoek;	Product management	Uitvragen requirements
UR27	Indien een beoordelaar opnieuw een beoordeling wil doen, wordt deze beoordeling genegeerd. Hier wordt d.m.v. een webpagina melding gemaakt;	Product management	Uitvragen requirements
UR28	Een Feedback verzoek heeft een geldigheid, bijvoorbeeld één week, wanneer er buiten deze geldigheid een	Product management	Uitvragen requirements

	beoordeling wordt verricht wordt de beoordeling genegeerd. Hier wordt d.m.v. een webpagina melding van gemaakt;		
UR29	Een Feedback verzoek kan binnen de geldigheid worden aangepast door de beoordelaar;	Product management	Uitvragen requirements
UR30	De behandelaar kan een Feedback verzoek meerdere malen versturen. Eventuele eerder gegeven beoordeling (inclusief toelichting) wordt hierdoor overschreven;	Product management	Uitvragen requirements
UR31	Indien een feedback mail bounced, wordt dit niet geregistreerd in het Feedback Systeem. Een mailer-demon melding aan de verzender is voldoende;	Product management	Uitvragen requirements
UR32	Het Feedback Systeem kan een Feedback verzoek genereren	Product management	
UR33	Het Feedback Systeem kan op de volgende manieren een Feedback verzoek genereren: per mail en via een REST-API call (voor het gebruik met Social Media).	Product management	
UR34	Een Feedback verzoek bestaat minimaal uit: <ul style="list-style-type: none"> • Gegevens van te beoordelen hulpvraag (uniek ID); • Gegevens behandelaar (Voornaam, achternaam, e-mail en nummer); • Gegevens beoordelaar (Voornaam, achternaam, e-mail en bedrijfsnaam) 	Product management	
UR35	Er kan gewaardeerd worden op de volgende manieren: 3 smiley-iconen (negatief, gewoon ok en goed) {Wellicht standaard op 3 manieren in DB 0, 1, 2 en labels in frontend, hierdoor is het mogelijk om andere namen te gebruiken}	Product management	
UR36	Het Feedback Systeem kan een gegenereerd Feedback verzoeken verzenden.	Product management	
UR37	Het Feedback Systeem kan op de volgende manieren een Feedback verzoek versturen: per mail.	Product management	

UR38	Een beoordelaar kan met één klik een beoordeling geven.	Product management	
UR39	Een beoordelaar kan een schriftelijke toelichting geven.	Product management	
UR40	De applicatie dient door meerdere gebruikers tegelijk gebruikt te kunnen worden, zonder dat er data vermenging plaats vindt.	Product management	
UR41	De webpagina's dienen responsive te zijn (wijzigen van feedback, geven van toelichting, fout pagina's, dankpagina's, geven van feedback [via link])	Medewerker support	Jos Verhaar
UR42	De API geeft de data in JSON formaat terug.	Medewerker support	Jos Verhaar
UR43	De API accepteert data in JSON formaat.	Medewerker support	Jos Verhaar

Technische beperkingen

Nummer	Omschrijving	Uitleg
TB01	De applicatie wordt geschreven in JAVA	Binnen MailPlus B.V. is dit de standaard
TB02	Als web applicatie framework wordt gebruik gemaakt van Spring MVC	Binnen MailPlus B.V. is dit de standaard
TB03	Voor de implementatie van de REST-API wordt gebruik gemaakt van Jersey (JAX-RS)	Binnen MailPlus B.V. is dit de standaard
TB04	Voor persistence (JPA) wordt gebruik gemaakt van EclipseLink	Binnen MailPlus B.V. is dit de standaard
TB05	Voor het opbouwen van de web-templates wordt Velocity gebruikt	Binnen MailPlus B.V. is dit de standaard

User requirements SMART gemaakt

Nummer	Omschrijving	UR	BR	SR
--------	--------------	----	----	----

UR01	Een beoordelaar mag één maal zijn/haar stem uitbrengen per Feedback verzoek	17,26	1	
UR02	Indien een beoordelaar opnieuw een beoordeling wil doen, wordt deze beoordeling genegeerd. Hier wordt d.m.v. een webpagina melding gemaakt	27	1	
UR03	Een Feedback verzoek heeft een geldigheid, bijvoorbeeld één week, wanneer er buiten deze geldigheid een beoordeling wordt verricht wordt de beoordeling genegeerd. Hier wordt d.m.v. een webpagina melding van gemaakt	28	1	
UR04	Een Feedback verzoek kan binnen de geldigheid worden aangepast door de beoordelaar	29	1	
UR05	De behandelaar kan een Feedback verzoek meerdere malen versturen. Eventuele eerder gegeven beoordeling (inclusief toelichting) wordt hierdoor overschreven	30	1	
UR06	Indien een feedback mail bounced, wordt dit niet geregistreerd in het Feedback Systeem. Een mailer-demon melding aan de verzender is voldoende	31	1	
UR07	De Feedback monitor toont een dag-, maand- en jaar overzicht. Er is een duidelijke overgang tussen de verschillende schermen.	1	1	
UR08	De dag-, maand- en jaar overzichten geven de cijfers van de huidige periode en die van de vorige periode (dag/ maand/ jaar).	1	5	
UR09	Het is mogelijk om behandelaren te beheren (toevoegen, bewerken en verwijderen) via het Feedback Systeem.	2, 7	6	
UR10	De tekst van een Feedback verzoek kan worden aangepast.	3, 10, 13	7	
UR11	De layout van een Feedback verzoek kan worden aangepast.	3, 10, 13	4	
UR12	Een Feedback verzoek bestaat minimaal uit: <ul style="list-style-type: none"> • Gegevens van te beoordelen hulpvraag (uniek ID en korte omschrijving); • Gegevens behandelaar (Voornaam, achternaam, e-mail en nummer); • Gegevens beoordelaar (Voornaam, achternaam, e-mail en bedrijfsnaam) 	4, 34	1	
UR13	Het is mogelijk om een Feedback verzoek te beperken voor een bepaalde beoordelaar (blacklist/ afmeldingen bijhouden (helemaal niet sturen), maximaal aantal per tijdseenheid {dag, week, maand})	5, 14	6	
UR14	Wanneer een aanvraag niet leidt tot een Feedback verzoek	5	6	

	wordt de reden hiervan opgeslagen in de database bij de bijbehorende aanvraag. (bijvoorbeeld door blacklist, missen van e-mail adres)			
UR15	Het verzendmoment van het feedback verzoek is aanpasbaar: <ul style="list-style-type: none"> • op een vasttijdstip • met een vertraging van x-aantal minuten 	6	6	
UR16	Het systeem kan met extra velden omgaan. M.a.w. toe te voegen in het systeem, vullen en verwerken in de mail. (Net als een databaseveld aanmaken, waarden importeren naar die velden en gebruiken in een mail als {[veldnaam]}).	8	7	
UR17	Het Feedback Systeem kan overweg met verschillende talen (NL en EN) { Mogelijk door het aanpasbaar maken van de teksten in de front-end}	9	8	
UR18	Het Feedback Systeem kan het versturen van een feedback verzoek wel/ niet versturen. Dit kan op basis van: <ul style="list-style-type: none"> • instellen door gebruiker • automatisch (bepaald door het systeem) 	11	6	
UR19	Wanneer er een lage feedback wordt gegeven wordt de operator (+ eventueel zijn/haar manager) hiervan per mail op de hoogte gebracht.	12	5	
UR20	Rapportage vanuit het systeem of in een tool waar de standaard rapportage bijgehouden wordt. Opties zoals download naar pdf, rapportage mailen naar een e-mailadres.	15	5	
UR21	Maandelijks statistieken naar de beheerder en/of een lijst van e-mailadressen	16	5	
UR22	De tekst van de landingspagina's kan worden aangepast.	13, 18	7	
UR23	De layout van de landingspagina's kan worden aangepast.	13, 18	4	
UR24	Via de Rest-API is het mogelijk om de statistieken op te vragen van het Feedback Systeem .	19	5	
UR25	De laatste x-aantal reacties pagina dient extern implementeerbaar te zijn als hele pagina en als (kleinere) widget.	20	5	
UR26	Vanaf deze externe implementatie kan men doorklikken naar MailPlus voor meer informatie over het Feedback Systeem .	20	5	
UR27	De dag-, maand- en jaar cijfers worden absoluut, in percentage en in (afgeronde) procentpunt getoond.	21, 25	5	
UR28	De waarde van de gemiddelde bestede- of totale bestede tijd moet getoond kunnen worden in dag/maand/jaar en hoort bij de	22	5	

	algemene cijfers zoals Aantal feedback verstuurd en Aantal gereageerd.			
UR29	De mogelijkheid tot het tonen van grafieken met daarin het percentage good/ok/bad in vergelijking met vorige periode (dag, maand en jaar)	23	5	
UR30	Bij de beoordelingen op het dag overzicht heb je de mogelijkheid tot het zien bij welk feedback verzoek deze hoort (naam beoordelaar, bedrijfsnaam en callnummer).	24	5	
UR31	Het Feedback Systeem kan een Feedback verzoek genereren	32	1	
UR32	Het Feedback Systeem kan op de volgende manieren een Feedback verzoek genereren: per mail en via een REST-API call (voor het gebruik met Social Media).	33	1	
UR33	Er kan gewaardeerd worden op de volgende manieren: 3 smiley-iconen (negatief, gewoon ok en goed) {Wellicht standaard op 3 manieren in DB 0, 1, 2 en labels in frontend, hierdoor is het mogelijk om andere namen te gebruiken}	35	1	
UR34	Het Feedback Systeem kan een gegenereerd Feedback verzoeken verzenden.	36	1	
UR35	Het Feedback Systeem kan op de volgende manieren een Feedback verzoek versturen: per mail.	37	1	
UR36	Een beoordelaar kan met één klik een beoordeling geven.	38	1	
UR37	Een beoordelaar kan een schriftelijke toelichting geven.	39	1	
UR38	De applicatie dient door meerdere gebruikers tegelijk gebruikt te kunnen worden, zonder dat er data vermenging plaats vindt.	40	2	
UR39	De webpagina's dienen responsive te zijn (wijzigen van feedback, geven van toelichting, fout pagina's, dankpagina's, geven van feedback [via link])	41	4	
UR40	De API accepteert en geeft data terug in JSON formaat.	42,43	1	

Geprioriteerde Requirements

Voor het prioriteren van de requirements wordt gebruik gemaakt van de prioriteringstechniek MoSCoW-prioritering.

Letter	Woord	Betekenis
--------	-------	-----------

M	Must have	Zonder deze requirements is het systeem onbruikbaar
S	Should have	Hoewel niet wenselijk is het mogelijk deze requirements achterwege te laten - vaak is er dan een workaround nodig
C	Could have	Deze requirements hebben een duidelijke toegevoegde waarde maar het systeem is ook bruikbaar zonder deze requirements
W	Won't have this time	Van deze requirements is afgesproken dat ze niet in de komen release geïmplementeerd zullen worden

Nummer	Omschrijving	Prioritering	Belanghebbende
UR01	Een beoordelaar mag één maal zijn/haar stem uitbrengen per Feedback verzoek	M	Product management
UR02	Indien een beoordelaar opnieuw een beoordeling wil doen, wordt deze beoordeling genegeerd. Hier wordt d.m.v. een webpagina melding gemaakt	M	Product management
UR03	Een Feedback verzoek heeft een geldigheid, bijvoorbeeld één week, wanneer er buiten deze geldigheid een beoordeling wordt verricht wordt de beoordeling genegeerd. Hier wordt d.m.v. een webpagina melding van gemaakt	M	Product management
UR04	Een Feedback verzoek kan binnen de geldigheid worden aangepast door de beoordelaar	M	Product management
UR05	De behandelaar kan een Feedback verzoek meerdere malen versturen. Eventuele eerder gegeven beoordeling (inclusief toelichting) wordt hierdoor overschreven	M	Product management
UR06	Indien een feedback mail bounced, wordt dit niet geregistreerd in het Feedback Systeem. Een mailer-demon melding aan de verzender is voldoende	M	Product management
UR07	De Feedback monitor toont een dag-, maand- en jaar overzicht. Er is een duidelijke overgang tussen de verschillende schermen.	S	Teammanager support
UR08	De dag-, maand- en jaar overzichten geven de cijfers van de huidige periode en die van de vorige periode (dag/ maand/ jaar).	S	Teammanager support

UR09	Het is mogelijk om behandelaren te beheren (toevoegen, bewerken en verwijderen) via het Feedback Systeem.	S	Teammanager support
UR10	De tekst van een Feedback verzoek kan worden aangepast.	C	Teammanager support
UR11	De layout van een Feedback verzoek kan worden aangepast.	C	Teammanager support
UR12	Een Feedback verzoek bestaat minimaal uit: <ul style="list-style-type: none"> • Gegevens van te beoordelen hulpvraag (uniek ID en korte omschrijving); • Gegevens behandelaar (Voornaam, achternaam, e-mail en nummer); • Gegevens beoordelaar (Voornaam, achternaam, e-mail en bedrijfsnaam) 	M	Teammanager support
UR13	Het is mogelijk om een Feedback verzoek te beperken voor een bepaalde beoordelaar (blacklist/ afmeldingen bijhouden (helemaal niet sturen), maximaal aantal per tijdseenheid {dag, week, maand})	C	Teammanager support
UR14	Wanneer een aanvraag niet leidt tot een Feedback verzoek wordt de reden hiervan opgeslagen in de database bij de bijbehorende aanvraag. (bijvoorbeeld door blacklist, missen van e-mail adres)	C	Teammanager support
UR15	Het verzendmoment van het feedback verzoek is aanpasbaar: <ul style="list-style-type: none"> • op een vasttijdstip • met een vertraging van x-aantal minuten 	W	Teammanager support
UR16	Het systeem kan met extra velden omgaan. M.a.w. toe te voegen in het systeem, vullen en verwerken in de mail. (Net als een databaseveld aanmaken, waarden importeren naar die velden en gebruiken in een mail als {[veldnaam]}).	W	Geïnteresseerde klanten
UR17	Het Feedback Systeem kan overweg met verschillende talen (NL en EN) { Mogelijk door het aanpasbaar maken van de teksten in de front-end}	S	Geïnteresseerde klanten
UR18	Het Feedback Systeem kan het versturen van een feedback verzoek wel/ niet versturen. Dit kan op basis van:	W	Geïnteresseerde klanten

	<ul style="list-style-type: none"> • instellen door gebruiker • automatisch (bepaald door het systeem) 		
UR19	Wanneer er een lage feedback wordt gegeven wordt de operator (+ eventueel zijn/haar manager) hiervan per mail op de hoogte gebracht.	W	Geïnteresseerde klanten
UR20	Rapportage vanuit het systeem of in een tool waar de standaard rapportage bijgehouden wordt. Opties zoals download naar pdf, rapportage mailen naar een e-mailadres.	W	Geïnteresseerde klanten
UR21	Maandelijkse statistieken naar de beheerder en/of een lijst van e-mailadressen	W	Geïnteresseerde klanten
UR22	De tekst van de landingspagina's kan worden aangepast.	C	Geïnteresseerde klanten
UR23	De layout van de landingspagina's kan worden aangepast.	C	Geïnteresseerde klanten
UR24	Via de Rest-API is het mogelijk om de statistieken op te vragen van het Feedback Systeem.	M	Geïnteresseerde klanten
UR25	De laatste x-aantal reacties pagina dient extern implementeerbaar te zijn als hele pagina en als (kleinere) widget.	S	Geïnteresseerde klanten
UR26	Vanaf deze externe implementatie kan men doorklikken naar MailPlus voor meer informatie over het Feedback Systeem.	C	Geïnteresseerde klanten
UR27	De dag-, maand- en jaar cijfers worden absoluut, in percentage en in (afgeronde) procentpunt getoond.	S	Medewerker support
UR28	De waarde van de gemiddelde bestede- of totale bestede tijd moet getoond kunnen worden in dag/maand/jaar en hoort bij de algemene cijfers zoals Aantal feedback verstuurd en Aantal gereageerd.	C	Medewerker support
UR29	De mogelijkheid tot het tonen van grafieken met daarin het percentage good/ok/bad in vergelijking met vorige periode (dag, maand en jaar).	S	Medewerker support
UR30	Bij de beoordelingen op het dag overzicht heb je de mogelijkheid tot het zien bij welk feedback verzoek deze hoort (naam	S	Medewerker support

	beoordelaar, bedrijfsnaam en callnummer).		
UR31	Het Feedback Systeem kan een Feedback verzoek genereren	M	
UR32	Het Feedback Systeem kan op de volgende manieren een Feedback verzoek genereren: per mail en via een REST-API call (voor het gebruik met Social Media).	M	
UR33	Er kan gewaardeerd worden op de volgende manieren: 3 smiley-iconen (negatief, gewoon ok en goed) {Wellicht standaard op 3 manieren in DB 0, 1, 2 en labels in frontend, hierdoor is het mogelijk om andere namen te gebruiken}	M	
UR34	Het Feedback Systeem kan een gegenereerd Feedback verzoeken verzenden.	M	
UR35	Het Feedback Systeem kan op de volgende manieren een Feedback verzoek versturen: per mail.	M	
UR36	Een beoordelaar kan met één klik een beoordeling geven.	M	
UR37	Een beoordelaar kan een schriftelijke toelichting geven.	M	
UR38	De applicatie dient door meerdere gebruikers tegelijk gebruikt te kunnen worden, zonder dat er data vermenging plaats vindt.	M	
UR39	De webpagina's dienen responsive te zijn (wijzigen van feedback, geven van toelichting, fout pagina's, dankpagina's, geven van feedback [via link])	C	
UR40	De API accepteert en geeft data terug in JSON formaat.	M	Medewerker support

Bijlage VI: Input voor requirements

Inbreng Paul Nederlof

De wensen concreet (de details volgen daaronder):

- Ik wil dat het systeem minder afhankelijk wordt van releases, updates en overige beperkingen die moeten worden doorgevoerd door o.a. development en product management. Nog steeds moeten veel verzoeken via PM ingepland worden. Dat zijn in mijn ogen onnodige verzoeken omdat deze met een beetje investering in de tool verwerkt kunnen worden en vanaf dat moment support zelf meer onderhoud kan voeren.
- Systeem als product: De klant die de uitnodiging ontvangt is enthousiast en er zijn geïnteresseerden. Momenteel zijn er 20-25 klanten geïnteresseerd. Dit aantal zal stijgen zodra we er meer over gaan vertellen en zodra het als product uitgerold kan worden.
- De laatste 100 reacties pagina op de website is nog niet beschikbaar (ook een verzoek vanuit Marketing).

Eigen wensen:

- Dag/Maand/Jaar rapportage op monitor is niet duidelijk. De overgang van maand naar jaar is te subtiel (alleen de cijfers veranderen en er zijn nog geen grafieken). De rapportage verbeteren en uitbreiden met eerdere resultaten. Sander zag dat de huidige formulieren en survey rapportage tool binnen Mailplus bijvoorbeeld hiermee gekoppeld kan worden zodat je trends kan bijhouden en dus meerdere jaren in mooie grafieken kan tonen. Is het deze maand druk ten opzichte van vorig jaar en hoeveel hebben we inmiddels aan reacties binnen in vergelijking met vorig jaar en wat was onze beste score ooit?
- Toevoegen operator/behandelaar binnen het proces. Nieuwe medewerkers moeten ingediend worden via TOPdesk en door escalatie naar 2^{de} lijns worden deze door development opgepakt.
- Aanpassingen aan de uitnodiging (html e-mail) moet worden ingepland met de eerstvolgende release (3-4 weken). Het beheer van de e-mail zouden wij liever bij onszelf willen houden
- Korte omschrijving van call toevoegen in de uitnodiging. Als een klant vaker heeft gesproken met Support kan een korte omschrijving meer duidelijkheid geven waar het gesprek over ging. De korte omschrijving wordt opgehaald van het veld “Korte omschrijving (Details)” uit TOPdesk en is een verplicht veld tijdens registratie. De waarde van het veld moet ergens in de mail geplaatst worden.
- “Ik ontvang jullie feedback te vaak”. Soms kan het de klant ook iets teveel worden... Tussen die duizenden contacten die wij hebben uitgenodigd zitten er soms gebruikers tussen die niet zitten te wachten op een uitnodiging. Ze krijgen het te vaak zeggen ze. Dat is bij te houden in TOPdesk. Echter TOPdesk kan ons niet helpen om dan ook geen mail te sturen of die herinnering te negeren. We kunnen namelijk een melding maken bij de gebruikersgegevens maar dat valt niet altijd op. Een blacklist in de database (handmatig toegevoegd door iemand met db toegang) zou irritatie bij de klant kunnen voorkomen.
Wij kunnen dan de uitnodiging sturen maar de laatste check vind plaats in de database. Mocht het adres voorkomen in de blacklist, dan wordt de uitnodiging niet verstuurd.

Met hierbij de voorkeur dat in de rapportage ergens staat dat wij het wel zouden hebben verstuurd zodat de cijfers blijven kloppen.

- Het tijdstip van verzenden van de getriggerde uitnodiging:
 - a) moet op een vast tijdstip mogelijk zijn. Bijvoorbeeld om 20:00 's avonds naar alle contacten van die dag.
 - b) met een vertraging van een xx minuten

De tool voor de klant:

Zonder dat wij hierop sturen heeft het feedbacksysteem sinds het begin bij 20+ klanten interesse gewekt en willen dus ook gebruik maken van deze tool.

Helaas voor deze klanten is de tool intern ontwikkeld en voor de Support afdeling op maat en naar wens gemaakt. Het is een lastige opgave om deze tool beschikbaar te maken voor meerdere klanten. Via deze weg wil ik toch proberen om een mogelijkheid te kunnen aanbieden voor deze klanten wat vergelijkbaar is met onze oplossing.

- De klant kan gebruikers/behandelaars beheren. (Toevoegen, verwijderen, wijzigen)
- Het systeem kan met extra velden omgaan. M.a.w. toe te voegen in het systeem, vullen en verwerken in de mail. (Net als een databaseveld aanmaken, waarden importeren naar die velden en gebruiken in een mail als `{{veldnaam}}`)
- De feedback kan beheerd worden. De smileys mogen een andere naam krijgen. Zo kan je ook een andere taal kiezen (Good, Ok, Bad)
- De klant heeft de uitnodiging/mail in een omgeving waar zij zelf bij kunnen of waar een contactpersoon (of supportafdeling) bij kan voor aanpassingen
- De klant heeft de keuze om in stellen dat vooraf bepaald kan worden of de mail wel/niet verstuurd wordt en de keuze of het systeem in staat is om willekeurig de uitnodiging te versturen
- Bij een slechte beoordeling kan een mail worden gestuurd naar de behandelaar (en eventueel kopie naar de afdelingsmanager)
- Lay-out beheer (logo, kleurtjes)
- Het systeem kan een blacklist bijhouden of rekening houden met afmelders
- Rapportage vanuit het systeem of in een tool waar de standaard rapportage bijgehouden wordt. Opties zoals download naar pdf, rapportage mailen naar een e-mailadres
- Maandelijkse stats naar de beheerder en/of een lijst van e-mailadressen
- Beveiliging in het systeem om eenmalig feedback te kunnen geven en met controle om binnen (..) uur te reageren
- Landingspagina na feedback (toelichting/motivatie) is te beheren via de tool.
- Via de Rest-API is het mogelijk om de gegevens op te vragen van het feedbacksysteem
- Er is een pagina beschikbaar waarin de laatste 100 reacties bijgehouden worden. Deze kan geëxporteerd en geïmplementeerd worden op een externe pagina. Logo van Mailplus erbij zodat geïnteresseerden kunnen doorklikken naar onze website.

Inbreng Sander van 't Hof

- Het tonen van het procentpunt verschil in good/ok/bad percentages in vergelijking met vorige periode (dag, maand, jaar) zoals in de happiness report afbeelding
- Het tonen van een omhoog/omlaag-afbeelding bij een hoger of lager percentage in vergelijking met vorige periode (dag, maand, jaar) zoals in de happiness report

afbeelding. Er zal een streepje kunnen worden getoond wanneer het verschil afgerond kleiner is dan een procentpunt.

- Het meesturen van de bestede tijd van een call. De waarde van de gemiddelde bestede tijd of totale bestede tijd moet getoond kunnen worden in dag/maand/jaar en hoort bij de algemene cijfers zoals Aantal feedback verstuurd en Aantal gereageerd
- Het tonen van Highcharts grafieken met daarin percentage good/ok/bad in vergelijking met vorige periode (dag, maand, jaar) zoals in de afbeelding Grafieken good ok bad.
- Het tonen van de naam/bedrijf/callnummer bij de betreffende smileys
- Totalen feedback weergeven per dag/maand/jaar. Niet alleen de percentages maar ook de absolute getallen

Inbreng Jos verhaar

- De API accepteert de data in JSON formaat
- De API antwoord met data in JSON formaat
- De webpagina's dienen responsive te zijn (wijzigen van feedback, geven van toelichting, fout pagina's, dankpagina's, geven van feedback [via link])
- De API calls zijn opgebouwd in een logische en voorspelbare structuur
- Het oude dashboard moet blijven werken in de nieuwe situatie
- Het nieuwe systeem dient door meerdere gebruikers gebruikt te kunnen worden, zonder dat dit problemen oplevert zoals datavermenging.
- Het is **niet** mogelijk om een operator te verwijderen, in plaats hiervan wordt deze inactief
- Voor iedere actieve operator kan er een specificatie worden getoond van de behaalde feedbacks
- Voor iedere inactieve operator wordt er geen specificatie opgehaald, maar de behaalde feedbacks worden wel mee genomen in de totalen.

Bijlage VII: Use Case beschrijvingen

01 Use Case: Verzend feedback verzoek via e-mail

naam	Verzend feedback verzoek via e-mail																
Identifier	UCD01																
Korte omschrijving	Genereren van feedback verzoek door behandelaar via e-mail. Daarna het verzenden van een feedback mail naar de beoordelaar.																
Primaire actor(en)	Gebruiker Feedback Systeem																
Secondaire actor(en)	Geen																
Precondities	De e-mail met de JSON data is goed ontvangen.																
Omschrijving	<table><tr><td>1</td><td>Als het systeem nieuwe mail detecteert.</td></tr><tr><td>1.1</td><td>Logt het systeem info met de tekst: "TopdeskMailProcess triggered @: "+new Date()+" found "+mails.length+" mails to be processed".</td></tr><tr><td>1.2</td><td>Voor iedere gevonden mail</td></tr><tr><td>1.2.1</td><td>Het systeem zet de mail om naar een Incident.</td></tr><tr><td>1.2.2</td><td>Verwerkt het systeem het incident.</td></tr><tr><td>1.2.3</td><td>Het systeem verwijdert de mail.</td></tr><tr><td>2</td><td>Anders logt het systeem een error met de tekst: "Could not find mailPath: "+mailPath+" cannot process TopdeskMails !".</td></tr></table>			1	Als het systeem nieuwe mail detecteert.	1.1	Logt het systeem info met de tekst: "TopdeskMailProcess triggered @: "+new Date()+" found "+mails.length+" mails to be processed".	1.2	Voor iedere gevonden mail	1.2.1	Het systeem zet de mail om naar een Incident.	1.2.2	Verwerkt het systeem het incident.	1.2.3	Het systeem verwijdert de mail.	2	Anders logt het systeem een error met de tekst: "Could not find mailPath: "+mailPath+" cannot process TopdeskMails !".
1	Als het systeem nieuwe mail detecteert.																
1.1	Logt het systeem info met de tekst: "TopdeskMailProcess triggered @: "+new Date()+" found "+mails.length+" mails to be processed".																
1.2	Voor iedere gevonden mail																
1.2.1	Het systeem zet de mail om naar een Incident.																
1.2.2	Verwerkt het systeem het incident.																
1.2.3	Het systeem verwijdert de mail.																
2	Anders logt het systeem een error met de tekst: "Could not find mailPath: "+mailPath+" cannot process TopdeskMails !".																
Postcondities	De feedback mail is verzonden aan de beoordelaar.																
Alternatieve flows	[01.1] Zet mail om naar incident. [01.2] Verwerk Incident.																

01.1 Use Case: Zet mail om naar incident

naam	Zet mail om naar incident										
Identifier	UCD01.1										
Korte omschrijving	Het Feedback Systeem leest de gevonden mail in en zet deze om naar een Incident Object.										
Primaire actor(en)	Het Feedback systeem										
Secondaire actor(en)	Geen										
Precondities	UCD01 is bezig en het systeem heeft een mail gevonden.										
Omschrijving	<table><tr><td>1</td><td>Deze alternatieve flow begint na stap 1.2.1 van de main flow.</td></tr><tr><td>2</td><td>Het systeem leest de mail in.</td></tr><tr><td>3</td><td>Het systeem zet de message body om naar een Incident Object.</td></tr><tr><td>4</td><td>Het systeem geeft het object terug.</td></tr></table>			1	Deze alternatieve flow begint na stap 1.2.1 van de main flow.	2	Het systeem leest de mail in.	3	Het systeem zet de message body om naar een Incident Object.	4	Het systeem geeft het object terug.
1	Deze alternatieve flow begint na stap 1.2.1 van de main flow.										
2	Het systeem leest de mail in.										
3	Het systeem zet de message body om naar een Incident Object.										
4	Het systeem geeft het object terug.										
Postcondities	Geen										
Alternatieve flows	Geen										

01.2 Use Case: Verwerk incident

naam	Verwerk incident	
Identifier	UCD01.2	
Korte omschrijving	Het Feedback Systeem verwerkt het Incident Object.	

Primaire actor(en)	Het systeem																		
Secondaire actor(en)	Geen																		
Precondities	UCD01 is bezig, het incident object dient verwerkt te worden.																		
Omschrijving	<table border="1"> <tr><td>1</td><td>Deze alternatieve flow begint na stap 1.2.2 van de main flow.</td></tr> <tr><td>2</td><td>Als het incident valide is.</td></tr> <tr><td>2.1</td><td>Als er reeds een feedback record bestaat voor het <i>unieke nummer</i>.</td></tr> <tr><td>2.1.1</td><td>Het systeem logt een info met de tekst: "Already created a feedback record for this ticketNumber: "+incident.getId()+" updating feedback record".</td></tr> <tr><td>2.1.2</td><td>Het systeem overschrijft het opgehaalde feedback record met de waarden vanuit het incident.</td></tr> <tr><td>2.1.3</td><td>Het systeem zet de vote date op NULL van het feedback record.</td></tr> <tr><td>2.1.4</td><td>Het systeem slaat het feedback record op in de database.</td></tr> <tr><td>2.2</td><td>Anders maakt het systeem een nieuw feedback record aan.</td></tr> <tr><td>2.3</td><td>De feedback mail wordt verzonden.</td></tr> </table>	1	Deze alternatieve flow begint na stap 1.2.2 van de main flow.	2	Als het incident valide is.	2.1	Als er reeds een feedback record bestaat voor het <i>unieke nummer</i> .	2.1.1	Het systeem logt een info met de tekst: "Already created a feedback record for this ticketNumber: "+incident.getId()+" updating feedback record".	2.1.2	Het systeem overschrijft het opgehaalde feedback record met de waarden vanuit het incident.	2.1.3	Het systeem zet de vote date op NULL van het feedback record.	2.1.4	Het systeem slaat het feedback record op in de database.	2.2	Anders maakt het systeem een nieuw feedback record aan.	2.3	De feedback mail wordt verzonden.
1	Deze alternatieve flow begint na stap 1.2.2 van de main flow.																		
2	Als het incident valide is.																		
2.1	Als er reeds een feedback record bestaat voor het <i>unieke nummer</i> .																		
2.1.1	Het systeem logt een info met de tekst: "Already created a feedback record for this ticketNumber: "+incident.getId()+" updating feedback record".																		
2.1.2	Het systeem overschrijft het opgehaalde feedback record met de waarden vanuit het incident.																		
2.1.3	Het systeem zet de vote date op NULL van het feedback record.																		
2.1.4	Het systeem slaat het feedback record op in de database.																		
2.2	Anders maakt het systeem een nieuw feedback record aan.																		
2.3	De feedback mail wordt verzonden.																		
Postcondities	Geen																		
Alternatieve flows	[01.2.1] Valideer incident. [01.2.2] Creëer nieuw feedback record. [01.2.3] Verzend feedback mail.																		

01.2.1 Use Case: Valideer incident

naam	Valideer incident																						
Identifier	UCD01.2.1																						
Korte omschrijving	Het systeem valideert het incident op basis van het e-mail adres van de beoordelaar.																						
Primaire actor(en)	Het systeem																						
Secondaire actor(en)	Geen																						
Precondities	UCD01.2 is bezig																						
Omschrijving	<table border="1"> <tr><td>1</td><td>Deze alternatieve flow begint na stap 2 van de main flow.</td></tr> <tr><td>2</td><td>Als er geen e-mail adres is gevonden.</td></tr> <tr><td>2.1</td><td>Logt het systeem een info met de tekst: "Invalid incident (ticketNumber="+ incident.getId() + "): Email is null"</td></tr> <tr><td>2.2</td><td>Het systeem geeft FALSE terug.</td></tr> <tr><td>3</td><td>Als er meerdere @ worden gevonden.</td></tr> <tr><td>3.1</td><td>Logt het systeem een info met de tekst: "Invalid incident (ticketNumber="+ incident.getId() + "): Multiple @ characters in email '"+email+"'"</td></tr> <tr><td>3.2</td><td>Het systeem geeft FALSE terug.</td></tr> <tr><td>4</td><td>Als er geen @ wordt gevonden.</td></tr> <tr><td>4.1</td><td>Logt het systeem een info met de tekst: "Invalid incident (ticketNumber="+ incident.getId() + "): Invalid value for email '"+email+"'"</td></tr> <tr><td>4.2</td><td>Het systeem geeft FALSE terug.</td></tr> <tr><td>5</td><td>Het systeem geeft TRUE terug.</td></tr> </table>	1	Deze alternatieve flow begint na stap 2 van de main flow.	2	Als er geen e-mail adres is gevonden.	2.1	Logt het systeem een info met de tekst: "Invalid incident (ticketNumber="+ incident.getId() + "): Email is null"	2.2	Het systeem geeft FALSE terug.	3	Als er meerdere @ worden gevonden.	3.1	Logt het systeem een info met de tekst: "Invalid incident (ticketNumber="+ incident.getId() + "): Multiple @ characters in email '"+email+"'"	3.2	Het systeem geeft FALSE terug.	4	Als er geen @ wordt gevonden.	4.1	Logt het systeem een info met de tekst: "Invalid incident (ticketNumber="+ incident.getId() + "): Invalid value for email '"+email+"'"	4.2	Het systeem geeft FALSE terug.	5	Het systeem geeft TRUE terug.
1	Deze alternatieve flow begint na stap 2 van de main flow.																						
2	Als er geen e-mail adres is gevonden.																						
2.1	Logt het systeem een info met de tekst: "Invalid incident (ticketNumber="+ incident.getId() + "): Email is null"																						
2.2	Het systeem geeft FALSE terug.																						
3	Als er meerdere @ worden gevonden.																						
3.1	Logt het systeem een info met de tekst: "Invalid incident (ticketNumber="+ incident.getId() + "): Multiple @ characters in email '"+email+"'"																						
3.2	Het systeem geeft FALSE terug.																						
4	Als er geen @ wordt gevonden.																						
4.1	Logt het systeem een info met de tekst: "Invalid incident (ticketNumber="+ incident.getId() + "): Invalid value for email '"+email+"'"																						
4.2	Het systeem geeft FALSE terug.																						
5	Het systeem geeft TRUE terug.																						
Postcondities	Geen																						
Alternatieve flows	Geen																						

01.2.2 Use Case: Creëer nieuw feedback record

naam	Creëer nieuw feedback record														
Identifier	UCD01.2.2														
Korte omschrijving	Het systeem maakt een nieuw feedback record aan en slaat deze op in de database.														
Primaire actor(en)	Het systeem														
Secondaire actor(en)	Geen														
Precondities	UCD01.2 is bezig, er dient een nieuw feedback record te worden gemaakt.														
Omschrijving	<table border="1"> <tr><td>1</td><td>Deze alternatieve flow begint na stap 2.2 van de main flow</td></tr> <tr><td>2</td><td>Het systeem genereert een <i>encl.d.</i></td></tr> <tr><td>3</td><td>Het systeem maakt een nieuw feedbackDao Object aan.</td></tr> <tr><td>4</td><td>Het systeem set het encl.d.</td></tr> <tr><td>5</td><td>Het systeem set de andere velden.</td></tr> <tr><td>6</td><td>Het systeem slaat dit feedbackDao Object op in de database.</td></tr> <tr><td>7</td><td>Het systeem geeft dit object terug</td></tr> </table>	1	Deze alternatieve flow begint na stap 2.2 van de main flow	2	Het systeem genereert een <i>encl.d.</i>	3	Het systeem maakt een nieuw feedbackDao Object aan.	4	Het systeem set het encl.d.	5	Het systeem set de andere velden.	6	Het systeem slaat dit feedbackDao Object op in de database.	7	Het systeem geeft dit object terug
1	Deze alternatieve flow begint na stap 2.2 van de main flow														
2	Het systeem genereert een <i>encl.d.</i>														
3	Het systeem maakt een nieuw feedbackDao Object aan.														
4	Het systeem set het encl.d.														
5	Het systeem set de andere velden.														
6	Het systeem slaat dit feedbackDao Object op in de database.														
7	Het systeem geeft dit object terug														
Postcondities	Er is een nieuw record in de database toegevoegd.														
Alternatieve flows	Geen														

01.2.3 Use Case: Verzend feedback mail

naam	Verzend feedback mail																		
Identifier	UCD01.2.3																		
Korte omschrijving																			
Primaire actor(en)	Het systeem																		
Secondaire actor(en)	Geen																		
Precondities	UCD01.2 is bezig, de feedback mail dient verzonden te worden.																		
Omschrijving	<table border="1"> <tr><td>1</td><td>Deze alternatieve flow begint na stap 2.3 van de main flow</td></tr> <tr><td>2</td><td>Het systeem bouwt de HTML versie van de mail op basis van parameters en de HTML template.</td></tr> <tr><td>3</td><td>Het systeem bouwt de Tekst versie van de mail op basis van parameters en de Tekst template.</td></tr> <tr><td>4</td><td>Het systeem vervangt parameters in het onderwerp met waarden uit het feedback record.</td></tr> <tr><td>5</td><td>De mail wordt verzonden naar de beoordelaar.</td></tr> <tr><td>5.1</td><td>De opgebouwde feedback mail wordt omgezet naar een echte mail.</td></tr> <tr><td>5.2</td><td>Het systeem voegt het aan- en van-adres, onderwerp en het bericht toe.</td></tr> <tr><td>5.3</td><td>De feedback mail wordt verzonden.</td></tr> <tr><td>6</td><td>Het systeem logt een info met de tekst: "send feedbackMail for ticketNumber: "+feedback.getTicketNumber()+" to email: "+feedback.getRequestorEmail().</td></tr> </table>	1	Deze alternatieve flow begint na stap 2.3 van de main flow	2	Het systeem bouwt de HTML versie van de mail op basis van parameters en de HTML template.	3	Het systeem bouwt de Tekst versie van de mail op basis van parameters en de Tekst template.	4	Het systeem vervangt parameters in het onderwerp met waarden uit het feedback record.	5	De mail wordt verzonden naar de beoordelaar.	5.1	De opgebouwde feedback mail wordt omgezet naar een echte mail.	5.2	Het systeem voegt het aan- en van-adres, onderwerp en het bericht toe.	5.3	De feedback mail wordt verzonden.	6	Het systeem logt een info met de tekst: "send feedbackMail for ticketNumber: "+feedback.getTicketNumber()+" to email: "+feedback.getRequestorEmail().
1	Deze alternatieve flow begint na stap 2.3 van de main flow																		
2	Het systeem bouwt de HTML versie van de mail op basis van parameters en de HTML template.																		
3	Het systeem bouwt de Tekst versie van de mail op basis van parameters en de Tekst template.																		
4	Het systeem vervangt parameters in het onderwerp met waarden uit het feedback record.																		
5	De mail wordt verzonden naar de beoordelaar.																		
5.1	De opgebouwde feedback mail wordt omgezet naar een echte mail.																		
5.2	Het systeem voegt het aan- en van-adres, onderwerp en het bericht toe.																		
5.3	De feedback mail wordt verzonden.																		
6	Het systeem logt een info met de tekst: "send feedbackMail for ticketNumber: "+feedback.getTicketNumber()+" to email: "+feedback.getRequestorEmail().																		
Postcondities	De feedback mail is verzonden aan de requestor.																		
Alternatieve flows	Geen																		

02 Use Case: Verzend feedback verzoek via Rest-API call

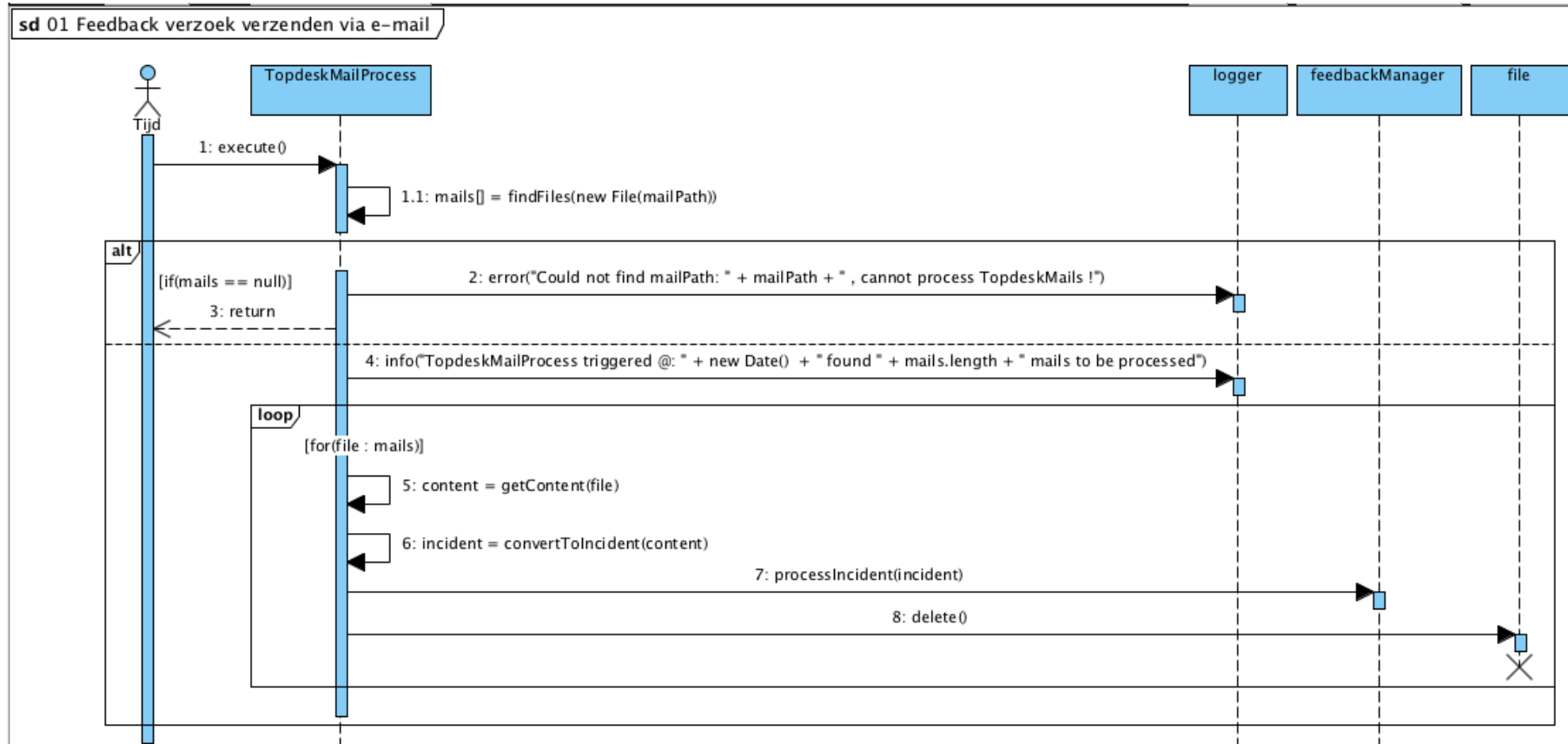
naam	Verzend feedback verzoek via Rest-API call														
Identifier	UCD02														
Korte omschrijving	Het systeem verzendt een feedback mail die is aangemaakt via een Rest-API call.														
Primaire actor(en)	Extern systeem														
Secondaire actor(en)	Geen														
Precondities	Er is via oAuth een succesvolle verbinding gemaakt.														
Omschrijving	<table border="1"> <tr> <td>1</td><td>Het externe systeem doet de Rest-API call post /feedback met de volgende opbouw in de body: <pre>{ "id": "[id]", "shortDescription": "[shortDescription]", "requestor": { "firstname": "[requestorFirstName]", "lastname": "[requestorLastName]", "account": "[requestorAccountName]", "email": "[requestorEmail]" }, "operator": { "firstname": "[operatorFirstName]", "lastname": "[operatorLastName]", "number": [operatorNumber] } }</pre> </td></tr> <tr> <td>2</td><td>Het systeem logt een info met de tekst: "post /feedback triggered @: " +new Date()+ " with id: "+body.id+"."</td></tr> <tr> <td>3</td><td>Als het request valide is.</td></tr> <tr> <td>3.2</td><td>Het systeem verwerkt het incident.</td></tr> <tr> <td>3.3</td><td>Het systeem geeft een succesmelding terug.</td></tr> <tr> <td>4</td><td>Het systeem logt een error met de tekst: "request is not valid, there is no feedback created."</td></tr> <tr> <td>4.1</td><td>Het systeem geeft een foutmelding terug met de reden.</td></tr> </table>	1	Het externe systeem doet de Rest-API call post /feedback met de volgende opbouw in de body: <pre>{ "id": "[id]", "shortDescription": "[shortDescription]", "requestor": { "firstname": "[requestorFirstName]", "lastname": "[requestorLastName]", "account": "[requestorAccountName]", "email": "[requestorEmail]" }, "operator": { "firstname": "[operatorFirstName]", "lastname": "[operatorLastName]", "number": [operatorNumber] } }</pre>	2	Het systeem logt een info met de tekst: "post /feedback triggered @: " +new Date()+ " with id: "+body.id+"."	3	Als het request valide is.	3.2	Het systeem verwerkt het incident.	3.3	Het systeem geeft een succesmelding terug.	4	Het systeem logt een error met de tekst: "request is not valid, there is no feedback created."	4.1	Het systeem geeft een foutmelding terug met de reden.
1	Het externe systeem doet de Rest-API call post /feedback met de volgende opbouw in de body: <pre>{ "id": "[id]", "shortDescription": "[shortDescription]", "requestor": { "firstname": "[requestorFirstName]", "lastname": "[requestorLastName]", "account": "[requestorAccountName]", "email": "[requestorEmail]" }, "operator": { "firstname": "[operatorFirstName]", "lastname": "[operatorLastName]", "number": [operatorNumber] } }</pre>														
2	Het systeem logt een info met de tekst: "post /feedback triggered @: " +new Date()+ " with id: "+body.id+"."														
3	Als het request valide is.														
3.2	Het systeem verwerkt het incident.														
3.3	Het systeem geeft een succesmelding terug.														
4	Het systeem logt een error met de tekst: "request is not valid, there is no feedback created."														
4.1	Het systeem geeft een foutmelding terug met de reden.														
Postcondities	<ul style="list-style-type: none"> Er is een feedback record aangemaakt; Er is een feedback mail verzonden aan de requestor 														
Alternatieve flows	[02.1] Valideer Rest-API request. [01.2] Verwerk Incident.														

02.1 Use Case: Valideer Rest-API request

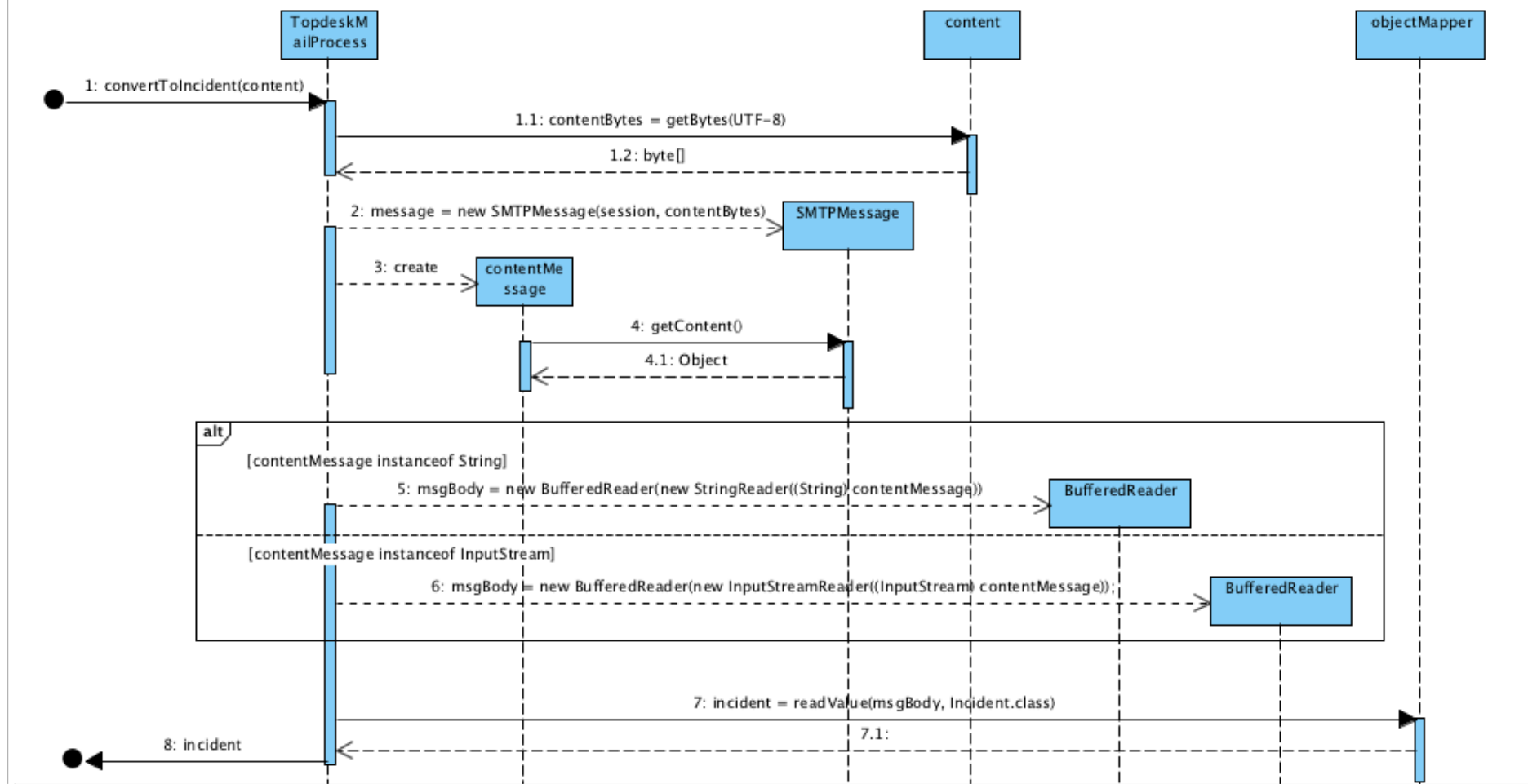
naam	Valideer Rest-API request		
Identifier	UCD02.1		
Korte omschrijving	Het systeem controleert of de verplichte velden aanwezig zijn.		
Primaire actor(en)	Het systeem		
Secondaire actor(en)	Geen		
Precondities	UCD02 is bezig en het Rest-API request dient gevalideerd te worden.		
Omschrijving	<table border="1"> <tr> <td>1</td><td>Deze alternatieve flow begint op stap 3 van de mainflow.</td></tr> </table>	1	Deze alternatieve flow begint op stap 3 van de mainflow.
1	Deze alternatieve flow begint op stap 3 van de mainflow.		

	2	Als er geen <i>id</i> is gevonden.
	2.1	Logt het systeem een info met de tekst: "Invalid incident : ticketNumber is null."
	2.2	Het systeem geeft FALSE terug.
	3	Als er geen <i>account</i> is gevonden.
	3.1	Logt het systeem een info met de tekst: "Invalid incident (ticketNumber=" + incident.getId() + ") : account is null.".
	3.2	Het systeem geeft FALSE terug.
	4	Als er geen <i>operator.number</i> is gevonden.
	4.1	Logt het systeem een info met de tekst: "Invalid incident (ticketNumber="+incident.getId()+") : operator.number is null.".
	4.2	Het systeem geeft FALSE terug.
	5	Het systeem geeft TRUE terug.
Postcondities		
Alternatieve flows		

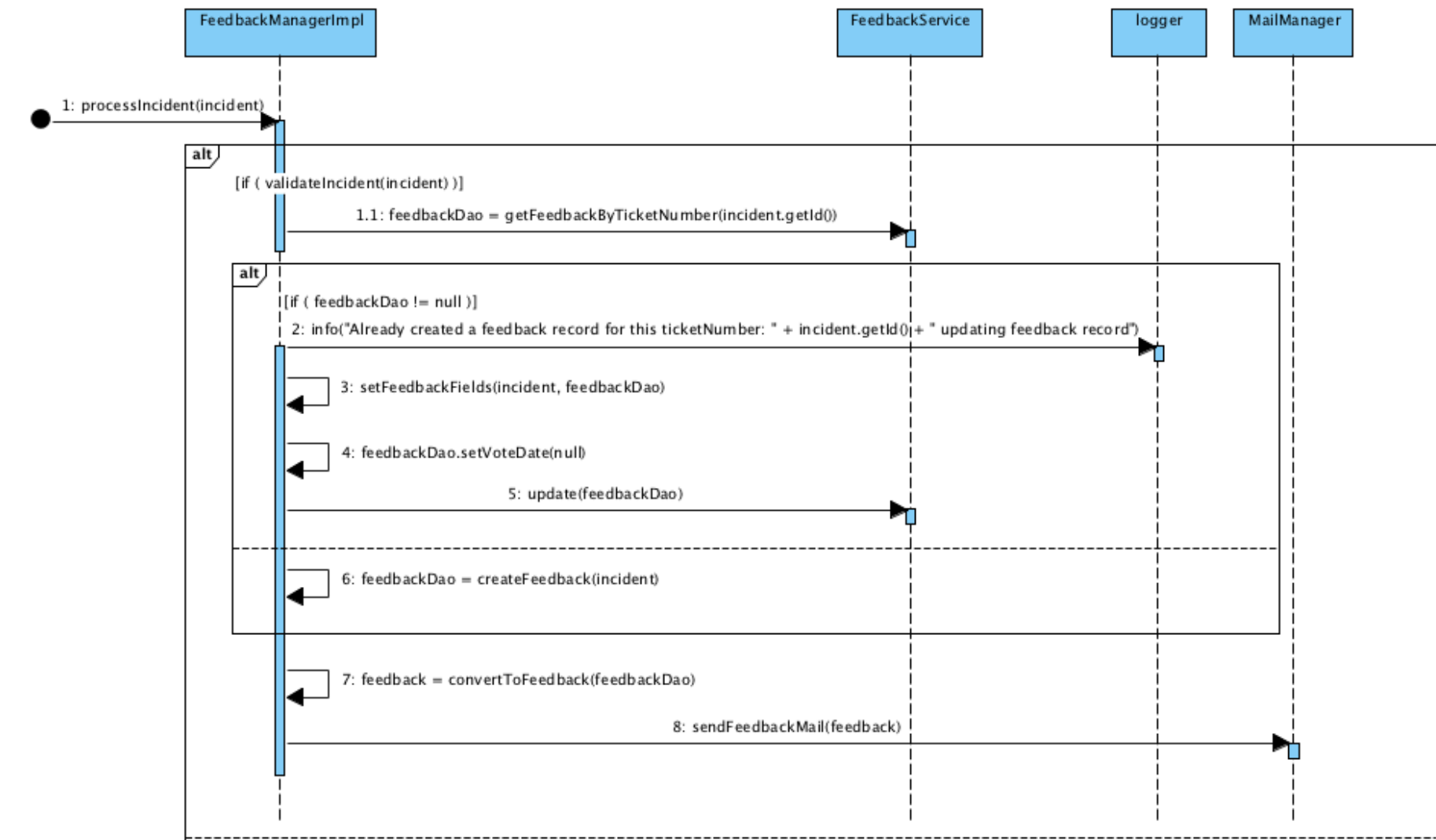
Bijlage VIII: Sequentie Diagrammen huidige implementatie



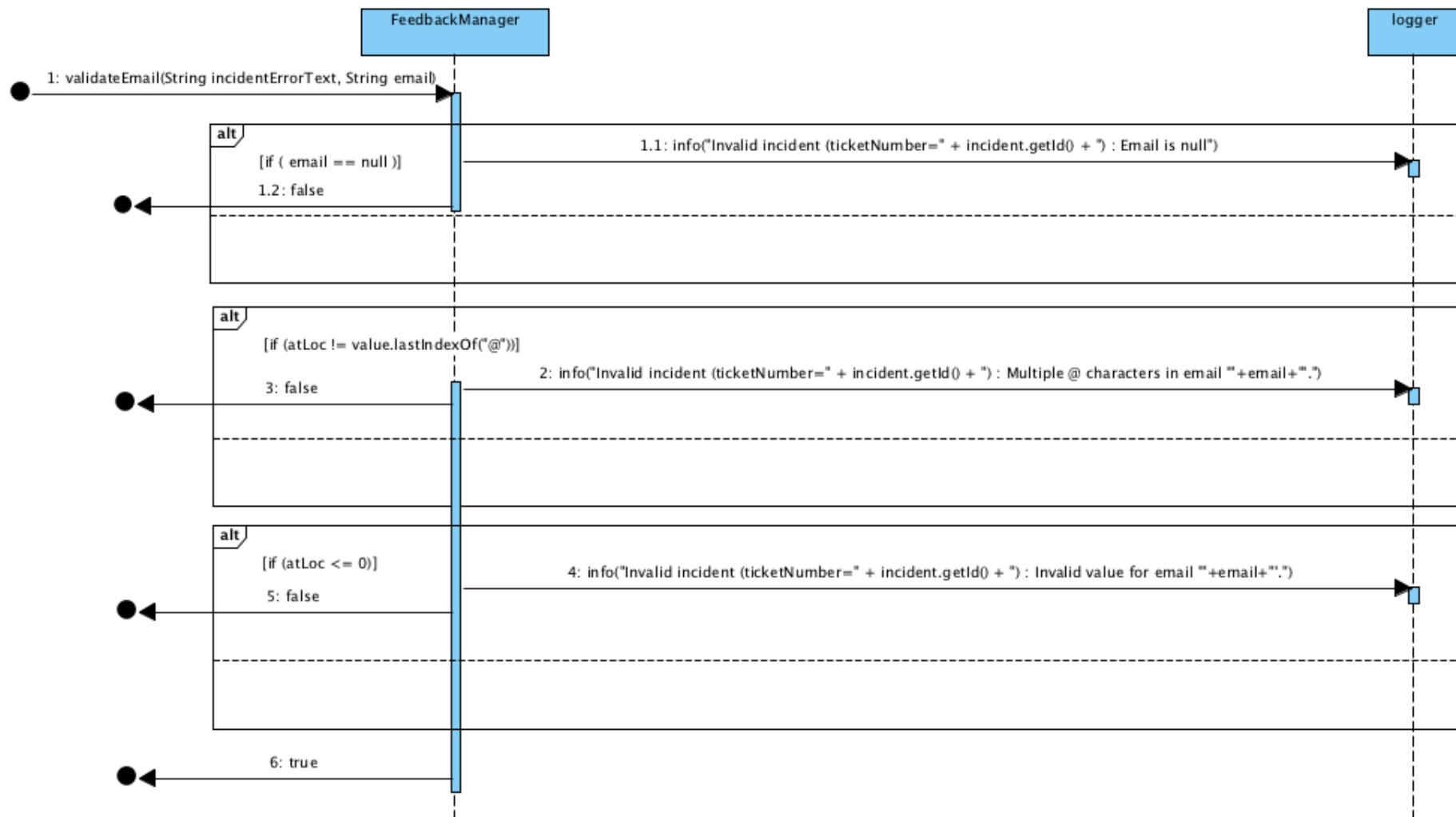
sd 01.1 Zet mail om naar incident



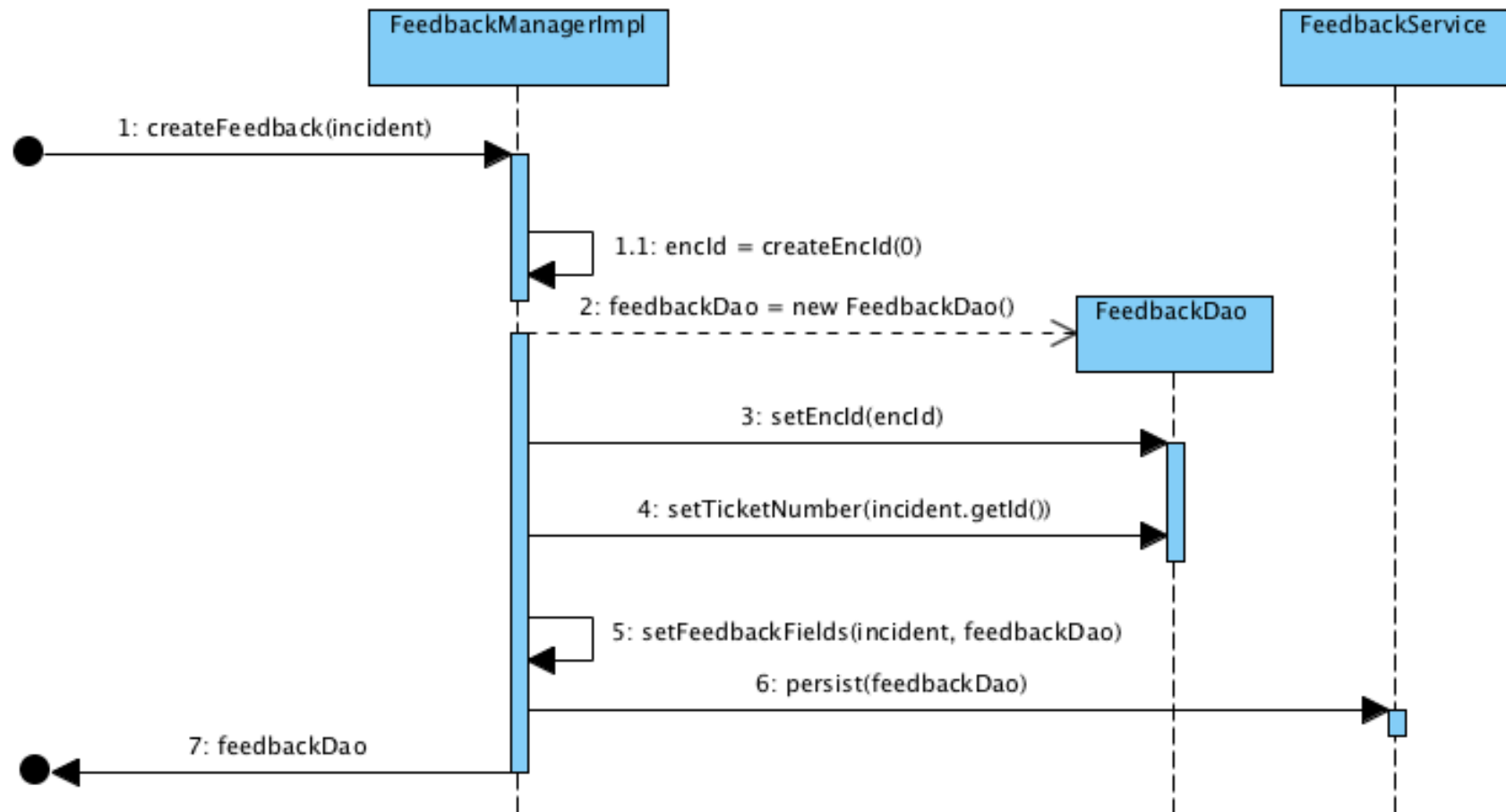
sd 01.2 Verwerk Incident



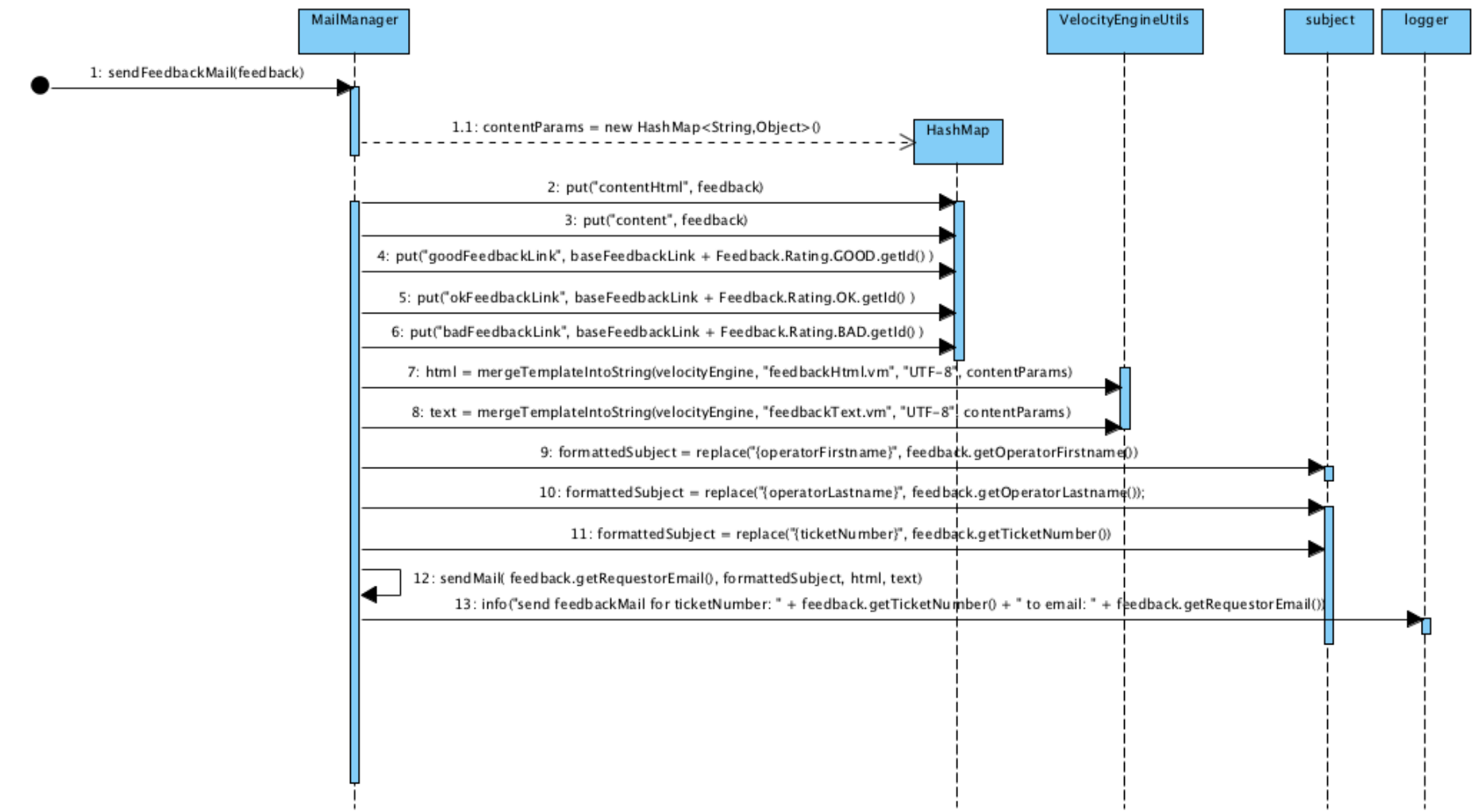
sd 01.2.1 Valideer incident



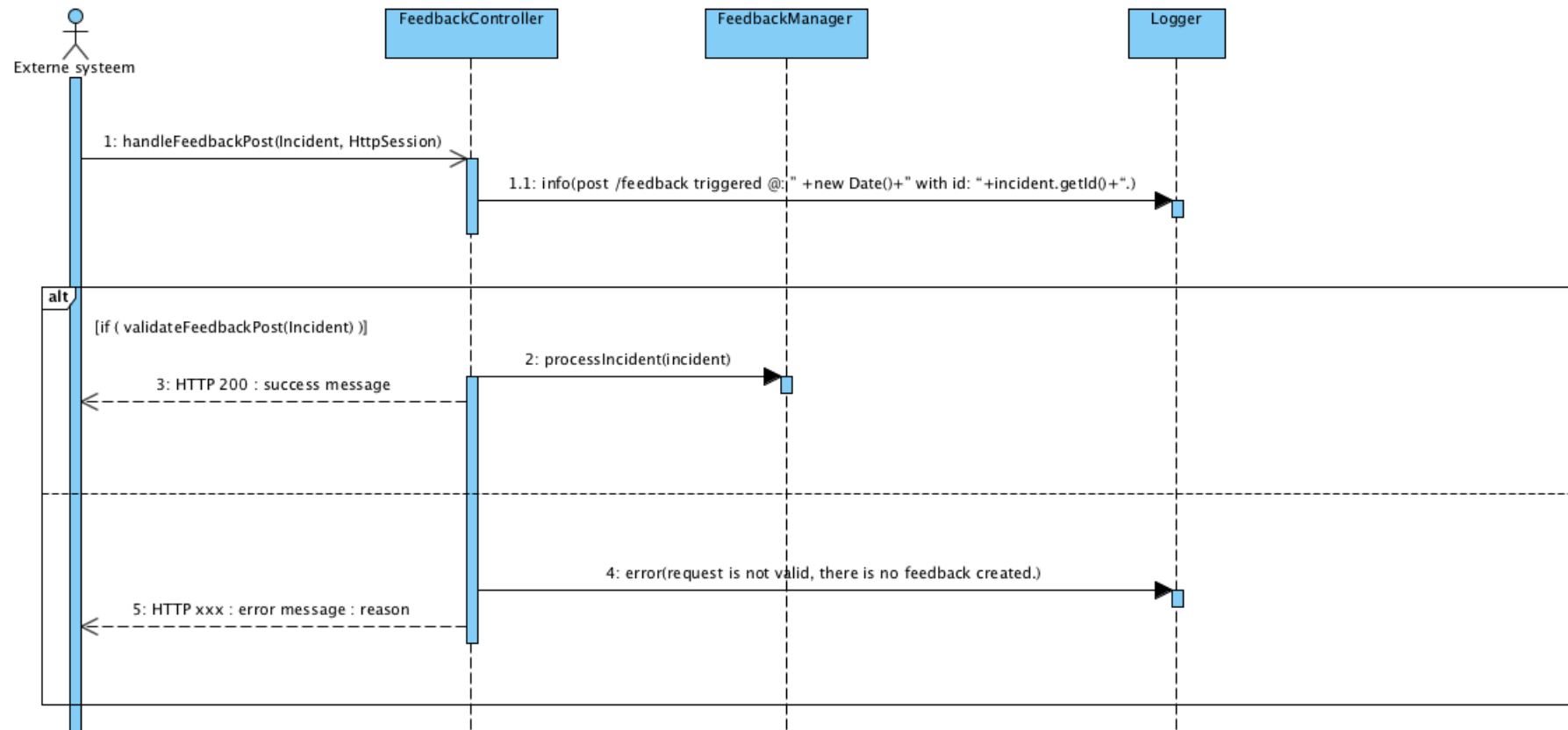
sd 01.2.2 Creëer nieuw feedback record



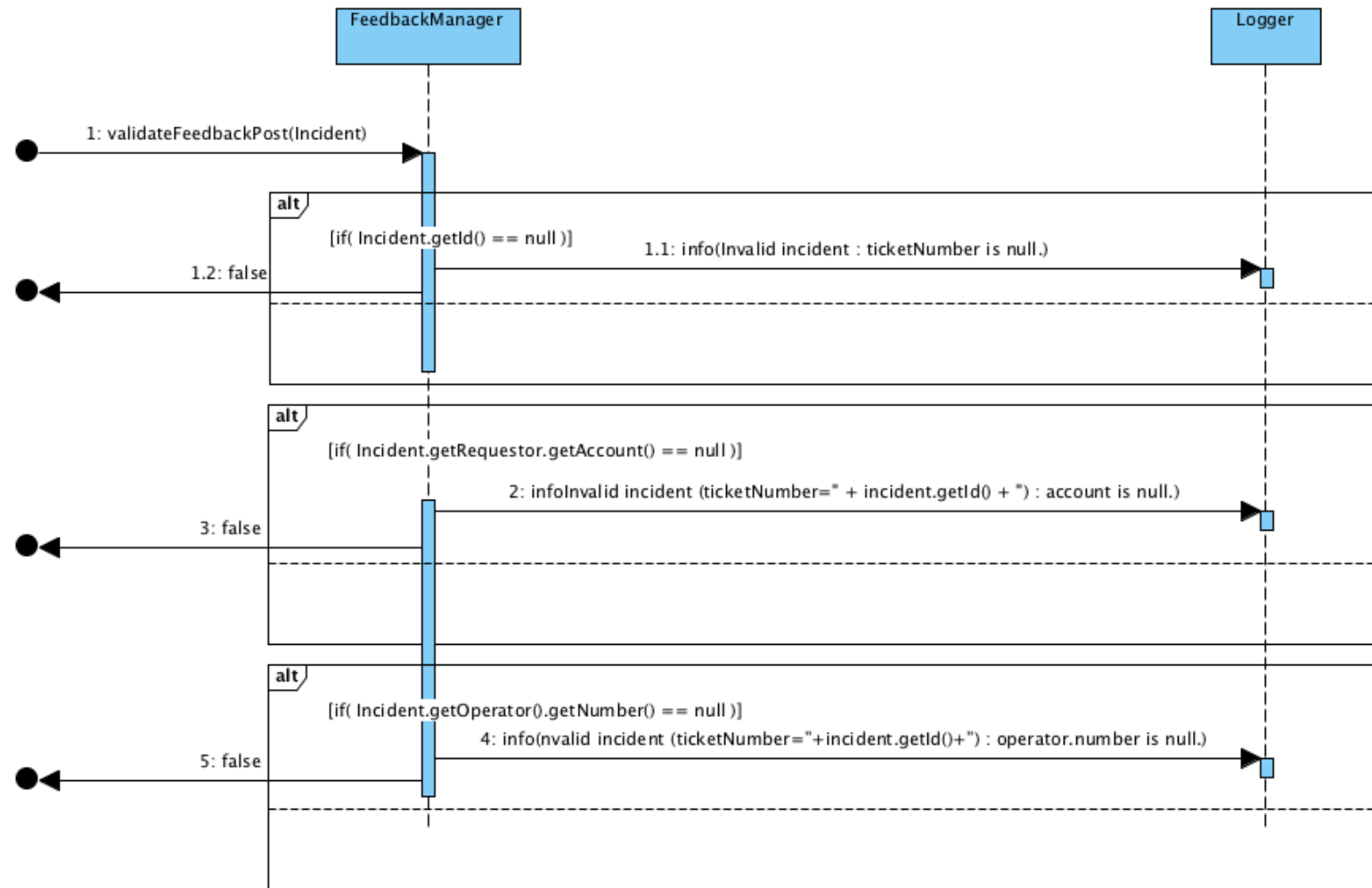
sd 01.2.3 Verzend feedback mail



sd 02 Feedback verzoek verzenden via Rest API call



sd 02.1 Valideer Rest API request



Bijlage IX: Testplan

Testbasis

System Under Test (SUT)

Tijdens het uitvoeren van de tests wordt niet het gehele systeem getest. De nadruk ligt vooral op de toegevoegde waarde. Dit is Rest-API van de applicatie. De MVC kant van het systeem, wat voornamelijk de frontend behelst, valt buiten beschouwing. Het SUT bevat alle toegevoegde functionaliteiten en de Rest-API.

De testbasis voor de uit te voeren tests bestaat uit verschillende documenten die in het afstudeerverslag aan bod komen. Je kunt hierbij denken aan:

- Use Case Beschrijvingen
- Klassendiagram huidig ontwerp
- Beslissingspunten diagrammen
- De geschreven code

Teststrategie

De doel van het testen van de ontwikkelde code is om aan te tonen dat de functionaliteit doet wat beoogd is. Door het maken van een testboom wordt in kaart gebracht welke onderdelen getest gaan worden. Deze is opgesteld tijdens de testrisicoanalyse. In afbeelding 53 wordt de testboom weergegeven.

Tijdens het opstellen van de testrisicoanalyse is gekeken naar de functionaliteiten van het SFS die inzichtelijk zijn gemaakt in de testboom. De functionaliteiten zijn ingedeeld naar belangrijkheid. Functionaliteiten die belangrijk zijn voor het systeem worden meer uitgebreid getest. Je kan hierbij denken aan meer verschillende tests die worden uitgevoerd, maar ook naar een grotere test diepte. Dit houdt in dat er meer testgevallen zijn opgesteld, waardoor er met meer zekerheid kan worden gezegd dat een functionaliteit doet wat er beoogd is.

Voor het testen van het SFS wordt gebruik gemaakt van verschillende testsoorten. Zo worden Unit Testen (hierna: UT), ook bekend als module tests, ingezet om de losse componenten van het systeem te testen. Met UT worden elementaire blokken van het SFS afzonderlijk getest. Deze test verlopen volledig geautomatiseerd zodat deze hergebruikt kunnen worden in de toekomst.

Naast UT wordt er ook gebruik gemaakt van Systeemtesten (hierna: ST). Dit zijn tests die de koppeling tussen verschillende onderdelen test. In de UT is aangetoond dat de onderdelen op zichzelf goed werken, maar werken ze ook goed samen?

Bij het opstellen van de logische testgevallen is gekeken naar de test soort waarvoor deze testgevallen gebruikt gaan worden. Aan de hand hiervan zijn verschillende testtechnieken geselecteerd. Voor de UT zijn Syntax testen geschreven. Met behulp van deze tests wordt gecontroleerd hoe het SFS reageert op valide en niet valide inputwaarden. Voor de ST zijn ook deze Syntax testen ingezet, maar zijn daarnaast ook Algoritme Testen (hierna: AT) met verschillende test diepten en Cause Effect graphing (hierna: CE) ingezet. CE testen zijn beter bekend als beslissingstabellen.

De geformuleerde logische testgevallen hebben als basis gediend voor de fysieke testgevallen. Deze zijn per testtechniek verder uitgewerkt volgens een vast stramien. Dit stramien zorgt ervoor dat de opzet voor ieder testgeval gelijk is en dat deze terug te traceren zijn tot de testboom.

1-D testrisicoanalyse

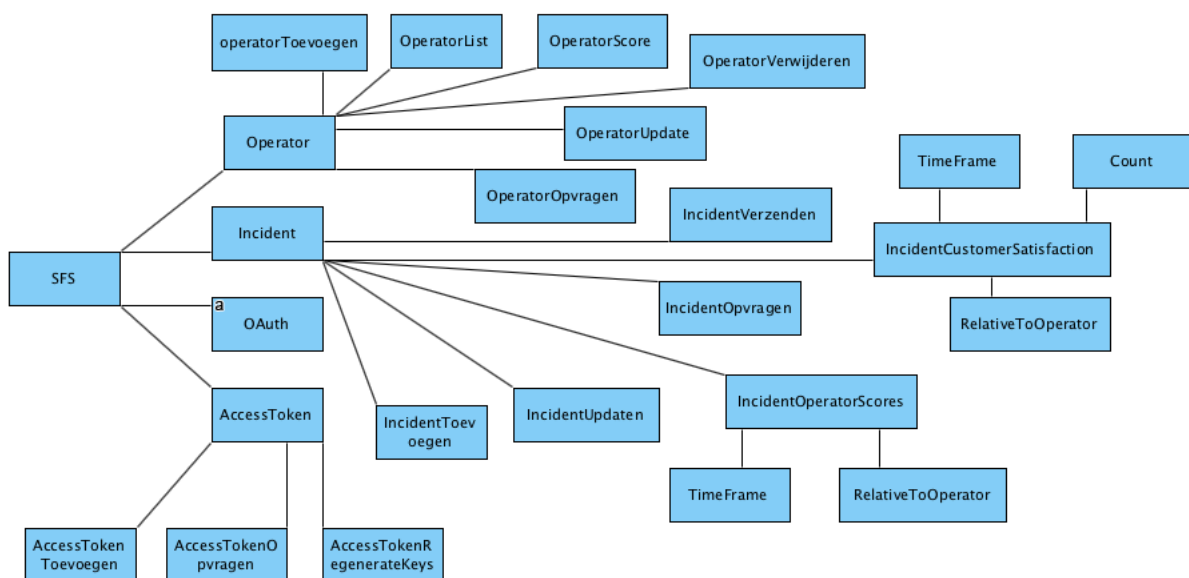
De uitvoering van de testrisicoanalyse bestaat uit een aantal stappen. Als eerste is de testboom opgesteld om de functies en aandachtspunten van het SFS in kaart te brengen.

Voor de aandachtspunten en functies die uit de testboom naar voren zijn gekomen is vervolgens het relatief belang bepaald. Hierin wordt aangegeven hoe belangrijk een bepaalde functie is voor het SFS.

Aan de hand van het relatieve belang zijn de risicocategorieën opgesteld. Hier wordt bepaald welke technieken en test diepte er voor welke functionaliteit wordt ingezet. Hiermee wordt bepaald hoe uitgebreid een functie getest gaat worden.

Testboom

Voor de testboom is het SFS onderverdeeld in hoofd- en subfuncties. Een hoofdfunctie is een verzameling van subfuncties. De hoofdfuncties in de testboom zijn: Operator, AccessToken, Incident en OAuth. Deze hoofdfuncties zijn verder onderverdeeld in de getoonde subfuncties.



Afbeelding 53: Testboom

Bepalen relatief belang

Voor de subfuncties die zijn bepaald met de testboom is het relatieve belang bepaald. Dit is gedaan door de student (verdeling 1) en de product owner (verdeling 2). Ieder heeft maximaal 69 punten gekregen om te verdelen over de verschillende subfuncties. De verdeling is te zien in onderstaande tabel.

Risicogebied	Verdeling 1	Verdeling 2	Totaal
Operator-Toevoegen	5	5	10
Operator-Updaten	5	5	10
Operator-Opvragen	1	3	4
Operator-Verwijderen	1	1	2
Operator-Get-List	1	1	2

Operator-Operator-Score	1	1	2
Gebruiker-Toevoegen	5	5	10
Gebruiker-Opvragen	1	3	4
Gebruiker-Regenerate-Keys	3	3	6
Incident-Toevoegen	5	5	10
Incident-Update	5	5	10
Incident-Opvragen	5	5	10
Incident-Versturen	5	5	10
Incident-Customer-Satisfaction-Timeframe	3	1	4
Incident-Customer-Satisfaction-Aantal	1	1	2
Incident-Customer-Satisfaction-Relative	3	1	4
Incident-Customer-Satisfaction-Relative-Operator	1	1	2
Incident-Operator-Scores-Timeframe	3	1	4
Incident-Operator-Scores-Relative	3	1	4
Incident-Operator-Scores-Relative-Operator	1	1	2
Incident-Live-Scores-Timeframe	1	1	2
Incident-Remarks-Timeframe	1	1	2
OAuth	9	5	14
Totaal	69	61	130

[tabel verdeling relatief belang]

Bepalen risicocategorieën

De verdeling van de subfuncties naar relatief belang levert de input voor het bepalen van de risicocategorieën. Deze zijn onderverdeeld naar aanleiding van de volgende verdeling.

Risicocategorie	Inhoud
Kritisch	Belangrijkste 10% van de risicogebieden
Hoog	Daaropvolgende 20% van de risicogebieden
Midden	Daaropvolgende 30% van de risicogebieden
Laag	Minst belangrijke 40% van de risicogebieden

[tabel verdeling risicocategorieën, bron: tabel 8.4, blz 100]

Het verdelen van de subfuncties gaat op aantal. Onder kritisch vallen 10% van de subfuncties. Voor het SFS zijn dit de drie functionaliteiten met de hoogste score. De vijf functionaliteiten met de hoogste score daarna, vallen in de categorie hoog. De zeven functionaliteiten met de hoogste score daarna, vallen in de categorie midden. De laatste acht functionaliteiten vallen in de categorie laag. Aan de hand van de hiervoor genoemde stappen is de volgende verdeling van de subfuncties gemaakt over de risicocategorieën.

Risicocategorie	Risicogebied	Relatief belang
Kritisch (10)	OAuth	14
	Incident-Toevoegen	10
	Incident-Update	10
Hoog (8)	Incident-Opvragen	10
	Incident-Versturen	10
	Gebruiker-Toevoegen	10
	Operator-Update	10
	Operator-Toevoegen	10
Midden (6)	Gebruiker-Regenerate-Keys	6
	Operator-Opvragen	4
	Gebruiker-Opvragen	4
	Incident-Customer-Satisfaction-Timeframe	4
	Incident-Customer-Satisfaction-Relative	4
	Incident-Operator-Scores-Timeframe	4
	Incident-Operator-Scores-Relative	4
Laag (2)	Operator-Verwijderen	2
	Operator-Get-List	2
	Operator-Operator-Score	2
	Incident-Customer-Satisfaction-Aantal	2
	Incident-Customer-Satisfaction-Relative-Operator	2
	Incident-Operator-Scores-Relative-Operator	2
	Incident-Live-Scores-Timeframe	2

	Incident-Remarks-Timeframe	2
--	----------------------------	---

[tabel verdeling (sub)functies over de risicocategorieën]

Nadat de risicocategorieën zijn verdeeld is bepaald met welke testtechnieken en test diepte de functies worden getest. Dit is terug te vinden in onderstaande tabel. Eerst wordt de risicocategorie genoemd, de kolommen ernaast geven de test domeinen: Invoer en Functionele afhandeling genoemd.

Risicocategorie	Invoer	Functionele afhandeling
Kritisch	Syntax: (niet) valide	AT branche coverage + CE
Hoog	Syntax: (niet) valide	AT branche coverage
Midden	Syntax: (niet) valide	AT statement coverage
Laag	Syntax: valide	

[Geselecteerde testtechnieken per test domein en risicocategorie]

In de tabel is te zien dat voor een laag risico functie alleen valide syntax wordt getest op de invoer. Voor een midden risico functie wordt dit uitgebreid met ook de niet valide tests. Hoe hoger in de tabel genoemd, hoe hoger de test sterkte is.

Producten

Logische testgevallen

Aan de hand van de bovenstaande tabel zijn de logische testgevallen opgesteld. Deze zijn per risicocategorie en per testtechniek doorlopen, beginnend met de risicocategorie laag. Nadat de logische gevallen voor de syntax testen waren bepaald zijn de logische testgevallen voor de AT testen van de midden risicocategorie bepaald, enz.

Syntax testen

SYN.T01: *Operator-Verwijderen* [*deleteOperator*]

Stap 1: Bepaling van de attribuutrestricties

operatorEncId: 35 Characters, Verplicht

Stap 2: Bepaal geldige en ongeldige waarden

	Geldig	Ongeldig
operatorEncId	35 Char	> 35
		< 35
		“”
		NULL

Stap 3: Bepaal testgevallen

	Geval 1
operatorEncId	35 Char
Resultaat	Geen error

SYN.T02: Operator-Get-List [*getOperatorList*]

Stap 1: Bepaling van de attribuutrestricties

userId: Integer, Verplicht

Stap 2: Bepaal geldige en ongeldige waarden

	Geldig	Ongeldig
userId	Integer	niet Integer
	≥ 1000	< 1000
		NULL

Stap 3: Bepaal testgevallen

	Geval 1
userId	Integer ≥ 1000
Resultaat	Geen error

SYN.T03: Operator-Operator-Score [*getOperatorScoreByTimeFrame*]

Stap 1: Bepaling van de attribuutrestricties

operatorEncId: 35 Characters, Verplicht

timeFrame: enum TimeFrame, Verplicht

Stap 2: Bepaal geldige en ongeldige waarden

	Geldig	Ongeldig
operatorEncId	35 Char	> 35
		< 35
		“”
		NULL

timeFrame	DAY, MONTH, Q1, Q2, Q3, Q4, YEAR	NULL
		niet DAY, MONTH, Q1, Q2, Q3, Q4, YEAR

Stap 3: Bepaal testgevallen

	Geval 1	Geval 2	Geval 3	Geval 4
operatorEncId	35 Char	35 Char	35 Char	35 Char
timeFrame	DAY	MONTH	Q1	Q2
Resultaat	Geen error	Geen error	Geen error	Geen error

	Geval 5	Geval 6	Geval 7
operatorEncId	35 Char	35 Char	35 Char
timeFrame	Q3	Q4	YEAR
Resultaat	Geen error	Geen error	Geen error

SYN.T04: Incident-Customer-Satisfaction-Aantal [getCustomerSatisfactionByCount/

Stap 1: Bepaling van de attribuutrestricties

userId: Integer, Verplicht

count: Integer, minimale waarde 1, Verplicht

Stap 2: Bepaal geldige en ongeldige waarden

	Geldig	Ongeldig
userId	Integer	niet Integer
	≥ 1000	< 1000
		NULL
count	> 0	≤ 0
		NULL
		niet Integer

Stap 3: Bepaal testgevallen

	Geval 1
--	---------

userId	>= 1000
count	> 0
Resultaat	Geen error

SYN.T05: Incident-Customer-Satisfaction-Relative-Operator
/getCustomerSatisfactionRelativeToByOperator/

Stap 1: Bepaling van de attribuurestricties

operatorEncId: 35 Characters, Verplicht
timeFrame: enum TimeFrame, Verplicht

Stap 2: Bepaal geldige en ongeldige waarden

	Geldig	Ongeldig
operatorEncId	35 Char	> 35
		< 35
		“”
		NULL
timeFrame	MONTH, Q1, Q2, Q3, Q4, YEAR	NULL
		niet MONTH, Q1, Q2, Q3, Q4, YEAR
		DAY

Stap 3: Bepaal testgevallen

	Geval 1	Geval 2	Geval 3
operatorEncId	35 Char	35 Char	35 Char
timeFrame	MONTH	Q1	Q2
Resultaat	Geen error	Geen error	Geen error

	Geval 4	Geval 5	Geval 6
operatorEncId	35 Char	35 Char	35 Char
timeFrame	Q3	Q4	YEAR

Resultaat	Geen error	Geen error	Geen error
-----------	------------	------------	------------

SYN.T06: Incident-Operator-Scores-Relative-Operator [/getOperatorScoresRelativeTo/

Stap 1: Bepaling van de attributrestricties

operatorEncId: 35 Characters, Verplicht

timeFrame: enum TimeFrame, Verplicht

Stap 2: Bepaal geldige en ongeldige waarden

	Geldig	Ongeldig
operatorEncId	35 Char	> 35
		< 35
		“”
		NULL
timeFrame	MONTH, Q1, Q2, Q3, Q4, YEAR	NULL
		niet MONTH, Q1, Q2, Q3, Q4, YEAR
		DAY

Stap 3: Bepaal testgevallen

	Geval 1	Geval 2	Geval 3
operatorEncId	35 Char	35 Char	35 Char
timeFrame	MONTH	Q1	Q2
Resultaat	Geen error	Geen error	Geen error

	Geval 4	Geval 5	Geval 6
operatorEncId	35 Char	35 Char	35 Char
timeFrame	Q3	Q4	YEAR
Resultaat	Geen error	Geen error	Geen error

SYN.T07: Incident-Live-Scores-Timeframe [/getLiveScoresByTimeFrame/

Stap 1: Bepaling van de attributrestricties

userId: Integer, Verplicht

timeFrame: enum TimeFrame, Verplicht

Stap 2: Bepaal geldige en ongeldige waarden

	Geldig	Ongeldig
userId	Integer	niet Integer
	>= 1000	<1000
		NULL
timeFrame	DAY, MONTH, Q1, Q2, Q3, Q4, YEAR	NULL
		niet DAY, MONTH, Q1, Q2, Q3, Q4, YEAR

Stap 3: Bepaal testgevallen

	Geval 1	Geval 2	Geval 3	Geval 4
userId	>= 1000	>= 1000	>= 1000	>= 1000
timeFrame	DAY	MONTH	Q1	Q2
Resultaat	Geen error	Geen error	Geen error	Geen error

	Geval 5	Geval 6	Geval 7
userId	>= 1000	>= 1000	>= 1000
timeFrame	Q3	Q4	YEAR
Resultaat	Geen error	Geen error	Geen error

SYN.T08: Incident-Remarks-Timeframe [getRemarksByTimeFrame/

Stap 1: Bepaling van de attributrestricties

userId: Integer, Verplicht

timeFrame: enum TimeFrame, Verplicht

Stap 2: Bepaal geldige en ongeldige waarden

	Geldig	Ongeldig
--	--------	----------

userId	Integer	niet Integer
	>= 1000	<1000
		NULL
timeFrame	DAY, MONTH, Q1, Q2, Q3, Q4, YEAR	NULL
		niet DAY, MONTH, Q1, Q2, Q3, Q4, YEAR

Stap 3: Bepaal testgevallen

	Geval 1	Geval 2	Geval 3	Geval 4
userId	>= 1000	>= 1000	>= 1000	>= 1000
timeFrame	DAY	MONTH	Q1	Q2
Resultaat	Geen error	Geen error	Geen error	Geen error

	Geval 5	Geval 6	Geval 7
userId	>= 1000	>= 1000	>= 1000
timeFrame	Q3	Q4	YEAR
Resultaat	Geen error	Geen error	Geen error

SYN.T09: Gebruiker-Regenerate-Keys [regenerateKeys/]

Stap 1: Bepaling van de attributrestricties

userId: Integer, Verplicht

user: AccessToken, Verplicht

Stap 2: Bepaal geldige en ongeldige waarden

	Geldig	Ongeldig
userId	Integer	niet Integer
	>= 1000	<1000
		NULL
userKey	32 Char	“”
		NULL

Stap 3: Bepaal testgevallen

	Geval 1	Geval 2	Geval 3
userId	>= 1000	>= 1000	>= 1000
userKey	32 Char	“”	NULL
Resultaat	Geen error	Error	Error

	Geval 4	Geval 5	Geval 6
userId	niet Integer	< 1000	NULL
userKey	32 Char	32 Char	32 Char
Resultaat	Error	Error	Error

Geval 4-5-6 zijn niet via bijvoorbeeld Postman te testen. Dit komt doordat de consumerKey valide moet zijn en de userId's uit de database komen...

SYN.T10: Gebruiker-Opvragen [/getUser/]

Stap 1: Bepaling van de attribuutrestricties

userId: Integer, Verplicht
userKey: 32 Char, Verplicht

Stap 2: Bepaal geldige en ongeldige waarden

	Geldig	Ongeldig
userId	Integer	niet Integer
	>= 1000	<1000
		NULL
userKey	32 Char	niet Char
		< 32 Char
		> 32 Char
		NULL

Stap 3: Bepaal testgevallen

	Geval 1	Geval 2	Geval 3	Geval 4
--	---------	---------	---------	---------

userId	>= 1000	Niet integer	< 1000	NULL
userKey	32 Char	32 Char	32 Char	32 Char
Resultaat	Geen error	Error	Error	Error

	Geval 5	Geval 6	Geval 7	Geval 8
user	>= 1000	>= 1000	>= 1000	>= 1000
userKey	niet Char	< 32	> 32	NULL
Resultaat	Error	Error	Error	Error

Geval 2-3-4 kunnen deze wel??

SYN.T11: Operator-Opvragen [/getOperator/]

Stap 1: Bepaling van de attribuurestricties

operatorEncId: 35 Characters, Verplicht

Stap 2: Bepaal geldige en ongeldige waarden

	Geldig	Ongeldig
operatorEncId	35 Char	> 35 Char
		< 35 Char
		“”
		NULL

Stap 3: Bepaal testgevallen

	Geval 1	Geval 2	Geval 3	Geval 4
operatorEncId	35 Char	> 35 Char	< 35 Char	“”
Resultaat	Geen error	Error	Error	Error

	Geval 5
operatorEncId	NULL
Resultaat	Error

SYN.T12: Incident-Customer-Satisfaction-Timeframe
/getCustomerSatisfactionByTimeFrame/

Stap 1: Bepaling van de attribuurestricties

userId: Integer, Verplicht

timeFrame: TimeFrame, Verplicht

Stap 2: Bepaal geldige en ongeldige waarden

	Geldig	Ongeldig
userId	Integer	niet Integer
	>= 1000	<1000
		NULL
timeFrame	DAY, MONTH, Q1, Q2, Q3, Q4, YEAR	NULL
		niet DAY, MONTH, Q1, Q2, Q3, Q4, YEAR

Stap 3: Bepaal testgevallen

	Geval 1	Geval 2	Geval 3	Geval 4
userId	>= 1000	>= 1000	>= 1000	>= 1000
timeFrame	DAY	MONTH	Q1	Q2
Resultaat	Geen error	Geen error	Geen error	Geen error

	Geval 5	Geval 6	Geval 7
userId	>= 1000	>= 1000	>= 1000
timeFrame	Q3	Q4	YEAR
Resultaat	Geen error	Geen error	Geen error

	Geval 8	Geval 9	Geval 10
userId	>= 1000	>= 1000	< 1000
timeFrame	NULL	2016	DAY

Resultaat	Geen error	Geen error	Error
-----------	------------	------------	-------

	Geval 11	Geval 12
userId	niet Integer	NULL
timeFrame	DAY	DAY
Resultaat	Error	Error

SYN.T13: Incident-Customer-Satisfaction-Relative [*getCustomerSatisfactionRelativeTo*]

Stap 1: Bepaling van de attribuutrestricties

userId: Integer, Verplicht

timeFrame: TimeFrame, Verplicht

Stap 2: Bepaal geldige en ongeldige waarden

	Geldig	Ongeldig
userId	Integer	niet Integer
	>= 1000	<1000
		NULL
timeFrame	MONTH, Q1, Q2, Q3, Q4, YEAR	NULL
		niet MONTH, Q1, Q2, Q3, Q4, YEAR
		DAY

Stap 3: Bepaal testgevallen

	Geval 1	Geval 2	Geval 3	Geval 4
userId	>= 1000	>= 1000	>= 1000	>= 1000
timeFrame	MONTH	Q1	Q2	Q3
Resultaat	Geen error	Geen error	Geen error	Geen error

	Geval 5	Geval 6	Geval 7
userId	>= 1000	>= 1000	>= 1000
timeFrame	Q4	YEAR	NULL

Resultaat	Geen error	Geen error	Geen error
-----------	------------	------------	------------

	Geval 8	Geval 9	Geval 10 ??
userId	>= 1000	>= 1000	< 1000
timeFrame	2016	DAY	MONTH
Resultaat	Geen error	Geen error	Error

	Geval 11 ??	Geval 12 ??
userId	Niet integer	NULL
timeFrame	MONTH	MONTH
Resultaat	Error	Error

SYN.T14: Incident-Operator-Scores-Timeframe [*getOperatorScoresByTimeFrame*]

Stap 1: Bepaling van de attribuurestricties

userId: Integer, Verplicht

Stap 2: Bepaal geldige en ongeldige waarden

	Geldig	Ongeldig
userId	Integer	niet Integer
	>= 1000	<1000
		NULL
timeFrame	DAY, MONTH, Q1, Q2, Q3, Q4, YEAR	NULL
		niet DAY, MONTH, Q1, Q2, Q3, Q4, YEAR

Stap 3: Bepaal testgevallen

	Geval 1	Geval 2	Geval 3	Geval 4
userId	>= 1000	>= 1000	>= 1000	>= 1000
timeFrame	DAY	MONTH	Q1	Q2

Resultaat	Geen error	Geen error	Geen error	Geen error
-----------	------------	------------	------------	------------

	Geval 5	Geval 6	Geval 7
userId	>= 1000	>= 1000	>= 1000
timeFrame	Q3	Q4	YEAR
Resultaat	Geen error	Geen error	Geen error

	Geval 8	Geval 9	Geval 10 ??
userId	>= 1000	>= 1000	< 1000
timeFrame	NULL	2016	DAY
Resultaat	Error	Error	Error

	Geval 11 ??	Geval 12 ??
userId	niet Integer	NULL
timeFrame	DAY	DAY
Resultaat	Error	Error

SYN.T15: Incident-Operator-Scores-Relative [*getOperatorScoresRelativeTo*]

Stap 1: Bepaling van de attributrestricties

userId: Integer, Verplicht

timeFrame: TimeFrame, Verplicht

Stap 2: Bepaal geldige en ongeldige waarden

	Geldig	Ongeldig
userId	Integer	niet Integer
	>= 1000	<1000
		NULL
timeFrame	MONTH, Q1, Q2, Q3, Q4, YEAR	NULL
		niet MONTH, Q1, Q2, Q3, Q4, YEAR

		DAY
--	--	-----

Stap 3: Bepaal testgevallen

	Geval 1	Geval 2	Geval 3	Geval 4
userId	>= 1000	>= 1000	>= 1000	>= 1000
timeFrame	MONTH	Q1	Q2	Q3
Resultaat	Geen error	Geen error	Geen error	Geen error

	Geval 5	Geval 6	Geval 7
userId	>= 1000	>= 1000	>= 1000
timeFrame	Q4	YEAR	NULL
Resultaat	Geen error	Geen error	Geen error

	Geval 8	Geval 9	Geval 10 ??
userId	>= 1000	>= 1000	< 1000
timeFrame	2016	DAY	MONTH
Resultaat	Geen error	Geen error	Error

	Geval 11 ??	Geval 12 ??
userId	Niet integer	NULL
timeFrame	MONTH	MONTH
Resultaat	Error	Error

SYN.T16: Incident-Opvragen [*getFeedback*]

Stap 1: Bepaling van de attributrestricties

feedbackEncId: 35 Char, Verplicht

Stap 2: Bepaal geldige en ongeldige waarden

	Geldig	Ongeldig
--	--------	----------

feedbackEncId	35 Char	> 35 Char
		< 35 Char
		“”
		NULL

Stap 3: Bepaal testgevallen

	Geval 1	Geval 2	Geval 3	Geval 4
feedbackEncId	35 Char	> 35 Char	< 35 Char	“”
Resultaat	Geen error	Error	Error	Error

	Geval 5
feedbackEncId	NULL
Resultaat	Error

SYN.T17: Incident-Versturen [*sendFeedbackMail*]

Stap 1: Bepaling van de attributrestricties

feedbackEncId: 35 Char, Verplicht

Stap 2: Bepaal geldige en ongeldige waarden

	Geldig	Ongeldig
feedbackEncId	35 Char	> 35 Char
		< 35 Char
		“”
		NULL

Stap 3: Bepaal testgevallen

	Geval 1	Geval 2	Geval 3	Geval 4
feedbackEncId	35 Char	> 35 Char	< 35 Char	“”
Resultaat	Geen error	Error	Error	Error

	Geval 5
--	---------

feedbackEncId	NULL
Resultaat	Error

SYN.T18: Gebruiker-Toevoegen [*addUser*]

Stap 1: Bepaling van de attributrestricties

userId: Integer, Verplicht
user: AccessToken, Verplicht

Stap 2: Bepaal geldige en ongeldige waarden

	Geldig	Ongeldig
userId	Integer	niet Integer
	>= 1000	<1000
		NULL
user	32 Char	“”
		NULL

Stap 3: Bepaal testgevallen

	Geval 1	Geval 2	Geval 3
userId	>= 1000	>= 1000	>= 1000
user	32 Char	“”	NULL
Resultaat	Geen error	Error	Error

	Geval 4 ??	Geval 5 ??	Geval 6 ??
userId	niet Integer	< 1000	NULL
user	32 Char	32 Char	32 Char
Resultaat	Error	Error	Error

SYN.T19: Operator-Updaten [*updateOperator*]

Stap 1: Bepaling van de attributrestricties

operator: Operator, Verplicht

Stap 2: Bepaal geldige en ongeldige waarden

	Geldig	Ongeldig
operator	valide operator*	NULL
		niet valide operator

*) Valide betekent verplichte velden zijn valide en gevonden in database.

Stap 3: Bepaal testgevallen

	Geval 1	Geval 2	Geval 3
operator	valide operator	NULL	niet valide operator
Resultaat	Geen error	Error	Error

SYN.T20: Operator-Toevoegen [*addOperator*]

Stap 1: Bepaling van de attributrestricties

operator: Operator, Verplicht

Stap 2: Bepaal geldige en ongeldige waarden

	Geldig	Ongeldig
operator	valide operator*	NULL
		niet valide operator

*) Valide betekent verplichte velden zijn valide, maar niet reeds aanwezig in database.

Stap 3: Bepaal testgevallen

	Geval 1	Geval 2	Geval 3
operator	valide operator	NULL	niet valide operator
Resultaat	Geen error	Error	Error

SYN.T21: Incident-Toevoegen [*addFeedback*]

Stap 1: Bepaling van de attributrestricties

incident: Incident, Verplicht

Stap 2: Bepaal geldige en ongeldige waarden

	Geldig	Ongeldig
--	--------	----------

incident	Valide Incident	NULL
		niet valide Incident

Stap 3: Bepaal testgevallen

	Geval 1	Geval 2	Geval 3
incident	valide Incident	NULL	niet valide Incident
Resultaat	Geen error	Error	Error

SYN.T22: Incident-Updaten [*addFeedback*]

Stap 1: Bepaling van de attributrestricties

incident: Incident, Verplicht

Stap 2: Bepaal geldige en ongeldige waarden

	Geldig	Ongeldig
incident	Valide Incident	NULL
		niet valide Incident

Stap 3: Bepaal testgevallen

	Geval 1	Geval 2	Geval 3
incident	valide Incident	NULL	niet valide Incident
Resultaat	Geen error	Error	Error

Algoritme Testen: statement coverage

AT.T01: Gebruiker-Regenerate-Keys

1 3 4 5 6 8

2 10

1 3 4 5 7 9

4) Definieer testgevallen

	Geval 1	Geval 2	Geval 3
Scenariobeschrijving	Succesvol genereren van nieuwe consumerKey en -	Validate consumerKey geeft een foutmelding, omdat deze niet valide	Update geeft een foutmelding, omdat de update in de database

	Secret.	is.	verkeerd gaat.
Pad	1-3-4-5-6-8	2-10	1-3-4-5-7-9

AT.T02: Gebruiker-Opvragen

1 3 4
2 5

4) Definieer testgevallen

	Geval 1	Geval 2
Scenariobeschrijving	Succesvol verkrijgen van gebruiker.	Validate consumerKey geeft een foutmelding, omdat deze niet valide is.
Pad	1-3-4	2-5

AT.T03: Operator-Opvragen

1 3 4
2 5

4) Definieer testgevallen

	Geval 1	Geval 2
Scenariobeschrijving	Succesvol verkrijgen van een Operator	Validate operatorEncId geeft een foutmelding, omdat deze niet valide is.
Pad	1-3-4	2-5

AT.T04: Incident-Customer-Satisfaction-Timeframe

1 4 5 6
2 3 4 5 6

4) Definieer testgevallen

	Geval 1	Geval 2
Scenariobeschrijving	Succesvol verkrijgen van Customer Satisfaction per TimeFrame	De waarde van het opgegeven TimeFrame wordt omgezet naar DAY en hiervan wordt de Customer Satisfaction opgehaald.
Pad	1-4-5-6	2-3-4-5-6

AT.T05: Incident-Customer-Satisfaction-Relative

1 3 6 8 9 11 14 17
 2 4 3 5 7 8 9 10 13 16
 1 3 5 7 8 9 10 13 16
 1 3 6 8 9 12 15 18

4) Definieer testgevallen

	Geval 1	Geval 2	Geval 3	Geval 4
Scenariobeschrijving	Succesvol ophalen van Customer Satisfaction. Per opgegeven kwartaal en hetzelfde kwartaal van het voorgaande jaar.	De waarde van het opgegeven TimeFrame wordt omgezet naar MONTH. Vervolgens wordt succesvol de Customer Satisfaction opgehaald. Per huidige maand en dezelfde maand van het voorgaande jaar.	De waarde van het opgegeven TimeFrame wordt omgezet naar MONTH. Vervolgens wordt succesvol de Customer Satisfaction opgehaald. Per huidige maand en dezelfde maand van het voorgaande jaar.	Succesvol ophalen van Customer Satisfaction. Per huidig jaar en het voorgaande jaar.
Pad	1-3-6-8-9-11-14-17	2-4-3-5-7-8-9-10-13 -16	1-3-5-7-8-9-10-13-16	1-3-6-8-9-12-15-18

AT.T06: Incident-Operator-Scores-Timeframe

1 4 5 6 7
 2 3 4 5 6 7

4) Definieer testgevallen

	Geval 1	Geval 2
Scenariobeschrijving	Succesvol ophalen van Operator Scores voor het opgegeven TimeFrame.	Het opgegeven TimeFrame wordt omgezet naar DAY. Daarna wordt de Operator Score succesvol opgehaald.
Pad	1-4-5-6-7	2-3-4-5-6-7

AT.T07: Incident-Operator-Scores-Relative

1-4-6-8-9-11-14-16
 2-3-4-5-7-8-9-10-13-17

1-4-5-7-8-9-10-13-17
1-4-6-8-9-12-15-18

4) Definieer testgevallen

	Geval 1	Geval 2	Geval 3	Geval 4
Scenariobeschrijving	Succesvol ophalen van Operator Scores. Per huidig kwartaal en hetzelfde kwartaal van het voorgaande jaar.	De waarde van het opgegeven TimeFrame wordt omgezet naar MONTH. Vervolgens wordt succesvol de Operator Scores opgehaald. Per huidige maand en dezelfde maand van het voorgaande jaar.	De waarde van het opgegeven TimeFrame wordt omgezet naar MONTH. Vervolgens wordt succesvol de Operator Scores opgehaald. Per huidige maand en dezelfde maand van het voorgaande jaar.	Succesvol ophalen van Operator Scores. Per huidig jaar en het voorgaande jaar.
Pad	1-4-6-8-9-11-14-16	2-3-4-5-7-8-9-10-13-17	1-4-5-7-8-9-10-13-17	1-4-6-8-9-12-15-18

Algoritme Testen: branche coverage

AT.T08: Incident-Opvragen

1-3-5
2-4

4) Definieer testgevallen

	Geval 1	Geval 2
Scenariobeschrijving	Succesvol ophalen van Feedback voor het opgegeven feedbackEncId	ValidateFeedbackEncId geeft een error, omdat het opgegeven feedbackEncId niet valide is.
Pad	1-3-5	2-4

AT.T09: Incident-Versturen

1-3-6
2-4
1-5-7

4) Definieer testgevallen

	Geval 1	Geval 2	Geval 3
Scenariobeschrijving	het succesvol verzenden van een feedback mail voor het opgegeven feedbackEncId.	ValidateFeedbackEncId geeft een error, omdat het opgegeven feedbackEncId niet valide is.	SendFeedbackMail geeft FALSE, omdat de feedback mail niet verzonden kon worden.
Pad	1-3-6	2-4	1-5-7

AT.T10: Gebruiker-Toevoegen

1-3-5-7-10

2-4

1-3-6-8

1-3-5-9-11

4) Definieer testgevallen

	Geval 1	Geval 2	Geval 3	Geval 4
Scenariobeschrijving	Het succesvol toevoegen van een User.	ValidateUser geeft een foutmelding, omdat de User niet valide is.	Er wordt een foutmelding gegeven, omdat de gebruiker reeds bestaat.	CreateUser geeft een exception, omdat het toevoegen niet lukt.
Pad	1-3-5-7-10	2-4	1-3-6-8	1-3-5-9-11

AT.T11: Operator-Updaten

1-2-4-5-9-10-11-12

1-2-3-7

1-2-4-5-6-8

4) Definieer testgevallen

	Geval 1	Geval 2	Geval 3
Scenariobeschrijving	Succesvol updaten van een Operator.	ValidateOperator geeft een foutmelding gegeven, omdat de Operator niet valide is.	Er wordt een foutmelding gegeven omdat de Operator niet bestaat.
Pad	1-2-4-5-9-10-11-12	1-2-3-7	1-2-4-5-6-8

AT.T12: Operator-Toevoegen

1-2-4-6-7-9-13-16-17

1-2-3-5

1-2-4-6-7-10-17

1-2-4-6-8-12-15

1-2-4-6-8-11-14

4) Definieer testgevallen

	Geval 1	Geval 2	Geval 3
Scenariobeschrijving	Het toevoegen van een bestaande Operator welke niet actief is, wordt weer actief gemaakt.	ValidateOperator geeft een foutmelding, omdat de Operator niet valide is.	Het toevoegen van een bestaande Operator welke ook actief is.
Pad	1-2-4-6-7-9-13-16-17	1-2-3-5	1-2-4-6-7-10-17

	Geval 4	Geval 5
Scenariobeschrijving	Het succesvol toevoegen van een nieuwe Operator.	CreateOperator geeft een exception, omdat het toevoegen niet is gelukt.
Pad	1-2-4-6-8-12-15	1-2-4-6-8-11-14

AT.T13: Incident-Toevoegen = Incident-Updaten

1-2-4-7-9-10

1-2-3-5

1-2-4-6-8

4) Definieer testgevallen

	Geval 1	Geval 2	Geval 3
Scenariobeschrijving	Het succesvol toevoegen van een nieuw incident.	ValidateIncident geeft een foutmelding, omdat het incident niet valide is.	ProcesIncident geeft een exception, omdat het verwerken van het incident niet is gelukt.
Pad	1-2-4-7-9-10	1-2-3-5	1-2-4-6-8

Cause Effect graphing

```
@Override
public IncidentResponse addFeedback(Incident incident) {

    IncidentResponse incidentResponse = new IncidentResponse();

    logger.info("Post /incident/ triggered for userId: " + incident.getUserId() + " @: " + new Date()
        + " with id: " + incident.getId() + ".");
    if( validateIncident(incident) ) {
        try {
            feedbackManager.processIncident(incident, false);
            Feedback feedback = feedbackManager.getFeedbackByIncident(incident);
            incidentResponse = getIncidentResponse(feedback, true);
        } catch(Exception e) {
            logger.error("Could not process post /incident", e);
            incidentResponse.setMessage("Could not process incident!");
        }
    } else {
        logger.error("request is not valid, there is no feedback created.");
        incidentResponse.setMessage("Request is not valid!");
    }

    return incidentResponse;
}
```

1. bepaal condities en acties

C1: valide incident

C2: throw exception

A1: processIncident

A2: setMessage => "Could not process incident!"

A3: setMessage => "Request is not valid"

2) Definieer beslissingstabel

Conditioes/ Acties	1	2	3	4
C1: Incident is valide	0	1	0	1
C2: Throw exception	0	0	1	1

A1: Verwerk incident		X		
A2: "Could not process incident"				X
A3: "Request is not valid"	X		X	

3) Reduceer het aantal testgevallen

Testgeval 3 dient verwijderd te worden. De reden is dat er geen mogelijkheid is tot het genereren van een exception wanneer het incident niet valide is. Deze exception kan voor komen wanneer het niet lukt om het incident te verwerken hiervoor dient deze wel valide te zijn. Ook is de uitkomst van dit geval gelijk aan testgeval 1.

4) Bepaal de testgevallen

	Geval 1	Geval 2	Geval 3	Geval 4
Scenariobeschrijving	Het request is niet valide,	Het incident wordt verwerkt	Is niet mogelijk	Er treed een exception op

	omdat het incident niet valide is.	tot een feedback in de database.		doordat het incident niet verwerkt kan worden.
--	------------------------------------	----------------------------------	--	--

Fysieke testgevallen

Operator-Verwijderen

Id	SYN.T01				
Doel van de test	Verifieer dat de status van een operator op FALSE staat als deze is verwijderd.				
Preconditie	Operator bestaat en heeft een status TRUE.				
Postconditie	Operator heeft een status FALSE.				
Verwijzing testboom	Operator-verwijderen				
Testactie(s)	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	deleteOperator wordt aangeroepen met operatorEncId “345-0-09df-3209-09df-3209-09df-3209	De status van de Operator in de response is FALSE		
		“			

Operator-Get-List

Id	SYN.T02				
Doel van de test	Verifieer dat er een lijst met de operators (actief en inactief) van de gebruiker wordt teruggegeven.				
Preconditie	De gebruiker is bekend en heeft een aantal operators.				
Postconditie	Er komt een lijst met operators terug als response.				
Verwijzing testboom	Operator-Get-List				
Testactie(s)					

	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	getOperatorList wordt aangeroepen met userId 1100	Een lijst met operators met de volgende encId's: e54beacc-09be-4bc8-bd41-a7b255bbc22b, 61987163-ca2f-4897-8cb3-e786701b41f9, 78222c7a-0c7d-47f8-8f1c-d3e3cfb8dd11		

Operator-Operator-Score

Id	SYN.T03-1				
Doel van de test	Verifieer dat de juiste operator score wordt teruggegeven voor de opgegeven dag en de dag ervoor.				
Preconditie	Operator is aanwezig en heeft op de dag een operator score. Timeframe is DAY				
Postconditie	Als response komen er twee operator scores terug voor de juiste operator.				
Verwijzing testboom	Operator-Operator-Score				
Testactie(s)	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call "operator/stats/operator-score/{operatorEncId}" wordt uitgevoerd met operatorEncId "DAO2"			

Id	SYN.T03-2
Doel van de test	Verifieer dat de juiste operator score wordt teruggegeven voor de opgegeven maand en de maand ervoor.

Preconditie	Operator is aanwezig en heeft in de maand een operator score. Timeframe is MONTH				
Postconditie	Als response komen er twee operator scores terug voor de juiste operator.				
Verwijzing testboom	Operator-Operator-Score				
Testactie(s)	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call “operator/stats/operator-score/{operatorEncId} ” wordt uitgevoerd met operatorEncId “78222c7a-0c7d-47f8-8f1c-d3e3cfb8dd11 ”			

Id	SYN.T03-3				
Doel van de test	Verifieer dat de juiste operator score wordt teruggegeven voor het opgegeven kwartaal en het kwartaal ervoor.				
Preconditie	Operator is aanwezig en heeft in de maand een operator score. Timeframe is Q1				
Postconditie	Als response komen er twee operator scores terug voor de juiste operator.				
Verwijzing testboom	Operator-Operator-Score				
Testactie(s)	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call “operator/stats/operator-score/{operatorEncId} ” wordt uitgevoerd met operatorEncId “78222c7a-0c7d-47f8-8f1c-d3e3cfb8dd11 ”			

Id	SYN.T03-4				
Doel van de test	Verifieer dat de juiste operator score wordt teruggeven voor het opgegeven kwartaal en het kwartaal ervoor.				
Preconditie	Operator is aanwezig en heeft in de maand een operator score. Timeframe is Q2				
Postconditie	Als response komen er twee operator scores terug voor de juiste operator.				
Verwijzing testboom	Operator-Operator-Score				
Testactie(s)	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call “operator/stats/operator-score/{operatorEncId} ” wordt uitgevoerd met operatorEncId “78222c7a-0c7d-47f8-8f1c-d3e3cfb8dd11 ”			

Id	SYN.T03-5				
Doel van de test	Verifieer dat de juiste operator score wordt teruggeven voor het opgegeven kwartaal en het kwartaal ervoor.				
Preconditie	Operator is aanwezig en heeft in de maand een operator score. Timeframe is Q3				
Postconditie	Als response komen er twee operator scores terug voor de juiste operator.				
Verwijzing testboom	Operator-Operator-Score				
Testactie(s)	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call “operator/stats/operator-score/{operatorEncId} ” wordt uitgevoerd met operatorEncId “78222c7a-			

		0c7d-47f8-8f1c-d3e3cfb8dd11 ”			
--	--	----------------------------------	--	--	--

Id	SYN.T03-6				
Doel van de test	Verifieer dat de juiste operator score wordt teruggeven voor het opgegeven kwartaal en het kwartaal ervoor.				
Preconditie	Operator is aanwezig en heeft in de maand een operator score. Timeframe is Q4				
Postconditie	Als response komen er twee operator scores terug voor de juiste operator.				
Verwijzing testboom	Operator-Operator-Score				
Testactie(s)	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call “operator/stats/operator-score/{operatorEncId}” wordt uitgevoerd met operatorEncId “78222c7a-0c7d-47f8-8f1c-d3e3cfb8dd11”			

Id	SYN.T03-7				
Doel van de test	Verifieer dat de juiste operator score wordt teruggeven voor het opgegeven jaar en het jaar ervoor.				
Preconditie	Operator is aanwezig en heeft in de maand een operator score. Timeframe is YEAR				
Postconditie	Als response komen er twee operator scores terug voor de juiste operator.				
Verwijzing testboom	Operator-Operator-Score				
Testactie(s)	Testactie	Testactie	Verwachte	Conclusie	Opmerking

	Id		reactie		
	1	De call “operator/stats/operator- score/{operatorEncId} ” wordt uitgevoerd met operatorEncId “78222c7a- 0c7d-47f8-8f1c- d3e3cfb8dd11 ”			

Incident-Customer-Satisfaction-Aantal

Id	SYN.T04				
Doel van de test	Verifieer dat het systeem de customer satisfaction terug geeft van de laatste 20 beoordeelde feedbacks en de 20 daarvoor.				
Preconditie	Voor het valide userId zijn 40 feedbacks beoordeeld.				
Postconditie	Als response wordt de customer satisfaction terug gegeven van de laatste 20 beoordeelde feedbacks en de 20 daarvoor.				
Verwijzing testboom	Incident-Customer-Satisfaction-Aantal				
Testactie(s)	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call “incident/stats/customer- satisfaction/count“ wordt uitgevoerd met een count van 20 met het userId 1000.			

Incident-Customer-Satisfaction-Relative-Operator

Id	SYN.T05-1				
Doel van de test	Controleer of er een customer satisfaction als response komt voor een valide operator voor de opgegeven maand en dezelfde maand het jaar ervoor.				
Preconditie	Er zijn voor de valide user feedbacks aanwezig voor het timeframe.				
Postconditie	De customer satisfaction voor de valide operator worden teruggegeven als response voor het huidige jaar en het jaar ervoor.				

Verwijzing testboom	Incident-Customer-Satisfaction-Relative-Operator				
Testactie(s)	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call “incident/stats/customer-satisfaction/{operatorEncId}/relative-to“ wordt uitgevoerd met timeframe MONTH en als operatorEncId “61987163-ca2f-4897-8cb3-e786701b41f9”			

Id	SYN.T05-2				
Doel van de test	Controleer of er een customer satisfaction als response komt voor een valide operator voor het opgegeven kwartaal en hetzelfde kwartaal het jaar ervoor.				
Preconditie	Er zijn voor de valide user feedbacks aanwezig voor het timeframe.				
Postconditie	De customer satisfaction voor de valide operator worden teruggegeven als response voor het huidige jaar en het jaar ervoor.				
Verwijzing testboom	Incident-Customer-Satisfaction-Relative-Operator				
Testactie(s)	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call “incident/stats/customer-satisfaction/{operatorEncId}/relative-to“ wordt uitgevoerd met timeframe Q1 en als operatorEncId “61987163-ca2f-4897-8cb3-e786701b41f9”			

Id	SYN.T05-3				
Doel van de test	Controleer of er een customer satisfaction als response komt voor een valide operator voor het opgegeven kwartaal en hetzelfde kwartaal het jaar ervoor.				

Preconditie	Er zijn voor de valide user feedbacks aanwezig voor het timeframe.				
Postconditie	De customer satisfaction voor de valide operator worden teruggegeven als response voor het huidige jaar en het jaar ervoor.				
Verwijzing testboom	Incident-Customer-Satisfaction-Relative-Operator				
Testactie(s)	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call “incident/stats/customer-satisfaction/{operatorEncId}/relative-to“ wordt uitgevoerd met timeframe Q2 en als operatorEncId “61987163-ca2f-4897-8cb3-e786701b41f9”			

Id	SYN.T05-4				
Doel van de test	Controleer of er een customer satisfaction als response komt voor een valide operator voor het opgegeven kwartaal en hetzelfde kwartaal het jaar ervoor.				
Preconditie	Er zijn voor de valide user feedbacks aanwezig voor het timeframe.				
Postconditie	De customer satisfaction voor de valide operator worden teruggegeven als response voor het huidige jaar en het jaar ervoor.				
Verwijzing testboom	Incident-Customer-Satisfaction-Relative-Operator				
Testactie(s)	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call “incident/stats/customer-satisfaction/{operatorEncId}/relative-to“ wordt uitgevoerd met timeframe Q3 en als operatorEncId “61987163-ca2f-4897-8cb3-e786701b41f9”			

Id	SYN.T05-5				
Doel van de test	Controleer of er een customer satisfaction als response komt voor een valide operator voor het opgegeven kwartaal en hetzelfde kwartaal het jaar ervoor.				
Preconditie	Er zijn voor de valide user feedbacks aanwezig voor het timeframe.				
Postconditie	De customer satisfaction voor de valide operator worden teruggegeven als response voor het huidige jaar en het jaar ervoor.				
Verwijzing testboom	Incident-Customer-Satisfaction-Relative-Operator				
Testactie(s)					
	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call “incident/stats/customer-satisfaction/{operatorEncId}/relative-to“ wordt uitgevoerd met timeframe Q4 en als operatorEncId “61987163-ca2f-4897-8cb3-e786701b41f9”			

Id	SYN.T05-6				
Doel van de test	Controleer of er een customer satisfaction als response komt voor een valide operator voor het opgegeven jaar en het jaar ervoor.				
Preconditie	Er zijn voor de valide user feedbacks aanwezig voor het timeframe.				
Postconditie	De customer satisfaction voor de valide operator worden teruggegeven als response voor het huidige jaar en het jaar ervoor.				
Verwijzing testboom	Incident-Customer-Satisfaction-Relative-Operator				
Testactie(s)					
	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call “incident/stats/customer-satisfaction/{operatorEncId}/relative-to“ wordt uitgevoerd met			

		timeframe YEAR en als operatorEncId “61987163-ca2f-4897-8cb3-e786701b41f9”			
--	--	--	--	--	--

Incident-Operator-Scores-Relative-Operator

Id	SYN.T06-1				
Doel van de test	Controleer of er een operator score als response komt voor een valide operator voor de opgegeven maand en dezelfde maand het jaar ervoor.				
Preconditie	Er zijn voor de valide user feedbacks aanwezig voor het timeframe.				
Postconditie	De operator score voor de valide operator worden teruggegeven als response voor het huidige jaar en het jaar ervoor.				
Verwijzing testboom	Incident-Operator-Scores-Relative-Operator				
Testactie(s)	Testacti e Id	Testactie	Verwacht e reactie	Conclusi e	Opmerkin g
	1	De call “incident/stats/operator-scores/{operatorEncId}/relative-to” wordt uitgevoerd met timeframe YEAR en als operatorEncId “61987163-ca2f-4897-8cb3-e786701b41f9”			

Id	SYN.T06-2				
Doel van de test	Controleer of er een operator score als response komt voor een valide operator voor het opgegeven kwartaal en hetzelfde kwartaal het jaar ervoor.				
Preconditie	Er zijn voor de valide user feedbacks aanwezig voor het timeframe.				
Postconditie	De operator score voor de valide operator worden teruggegeven als response voor het huidige jaar en het jaar ervoor.				
Verwijzing testboom	Incident-Operator-Scores-Relative-Operator				
Testactie(s)					

	Testacti e Id	Testactie	Verwacht e reactie	Conclusi e	Opmerkin g
	1	De call “incident/stats/operator- scores/{operatorEncId}/relativ e-to“ wordt uitgevoerd met timeframe Q1 en als operatorEncId “61987163- ca2f-4897-8cb3- e786701b41f9”			

Id	SYN.T06-3				
Doel van de test	Controleer of er een operator score als response komt voor een valide operator voor het opgegeven kwartaal en hetzelfde kwartaal het jaar ervoor.				
Preconditie	Er zijn voor de valide user feedbacks aanwezig voor het timeframe.				
Postconditi e	De operator score voor de valide operator worden teruggegeven als response voor het huidige jaar en het jaar ervoor.				
Verwijzing testboom	Incident-Operator-Scores-Relative-Operator				
Testactie(s)	Testacti e Id	Testactie	Verwacht e reactie	Conclusi e	Opmerkin g
	1	De call “incident/stats/operator- scores/{operatorEncId}/relativ e-to“ wordt uitgevoerd met timeframe Q2 en als operatorEncId “61987163- ca2f-4897-8cb3- e786701b41f9”			

Id	SYN.T06-4				
Doel van de test	Controleer of er een operator score als response komt voor een valide operator voor het opgegeven kwartaal en hetzelfde kwartaal het jaar ervoor.				
Preconditie	Er zijn voor de valide user feedbacks aanwezig voor het timeframe.				
Postconditi	De operator score voor de valide operator worden teruggegeven als response				

e	voor het huidige jaar en het jaar ervoor.				
Verwijzing testboom	Incident-Operator-Scores-Relative-Operator				
Testactie(s)	Testacti e Id	Testactie	Verwacht e reactie	Conclusi e	Opmerkin g
	1	De call “incident/stats/operator- scores/{operatorEncId}/relativ e-to“ wordt uitgevoerd met timeframe Q3 en als operatorEncId “61987163- ca2f-4897-8cb3- e786701b41f9”			

Id	SYN.T06-5				
Doel van de test	Controleer of er een operator score als response komt voor een valide operator voor het opgegeven kwartaal en hetzelfde kwartaal het jaar ervoor.				
Preconditie	Er zijn voor de valide user feedbacks aanwezig voor het timeframe.				
Postconditie	De operator score voor de valide operator worden teruggegeven als response voor het huidige jaar en het jaar ervoor.				
Verwijzing testboom	Incident-Operator-Scores-Relative-Operator				
Testactie(s)	Testacti e Id	Testactie	Verwacht e reactie	Conclusi e	Opmerkin g
	1	De call “incident/stats/operator- scores/{operatorEncId}/relativ e-to“ wordt uitgevoerd met timeframe Q4 en als operatorEncId “61987163- ca2f-4897-8cb3- e786701b41f9”			

Id	SYN.T06-6				
----	-----------	--	--	--	--

Doel van de test	Controleer of er een operator score als response komt voor een valide operator voor het opgegeven jaar en het jaar ervoor.				
Preconditie	Er zijn voor de valide user feedbacks aanwezig voor het timeframe.				
Postconditie	De operator score voor de valide operator worden teruggegeven als response voor het huidige jaar en het jaar ervoor.				
Verwijzing testboom	Incident-Operator-Scores-Relative-Operator				
Testactie(s)					
	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call “incident/stats/operator-scores/{operatorEncId}/relative-to” wordt uitgevoerd met timeframe YEAR en als operatorEncId “61987163-ca2f-4897-8cb3-e786701b41f9”			

Incident-Live-Scores-Timeframe

Id	SYN.T07-1				
Doel van de test	Controller of er een live scores als response komt voor het opgegeven valide timeframe.				
Preconditie	Er zijn voor het valide timeframe beoordeelde feedbacks aanwezig.				
Postconditie	De live scores voor het valide timeframe worden teruggegeven als response voor de huidige periode en de periode ervoor.				
Verwijzing testboom	Incident-Live-Scores-Timeframe				
Testactie(s)					
	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call “incident/stats/live-scores” wordt uitgevoerd voor het timeframe DAY en het userId 1000			

Id	SYN.T07-2				
Doel van de test	Controller of er een live scores als response komt voor het opgegeven valide timeframe.				
Preconditie	Er zijn voor het valide timeframe beoordeelde feedbacks aanwezig.				
Postconditie	De live scores voor het valide timeframe worden teruggegeven als response voor de huidige periode en de periode ervoor.				
Verwijzing testboom	Incident-Live-Scores-Timeframe				
Testactie(s)	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call “incident/stats/live-scores“ wordt uitgevoerd voor het timeframe MONTH en het userId 1000			

Id	SYN.T07-3				
Doel van de test	Controller of er een live scores als response komt voor het opgegeven valide timeframe.				
Preconditie	Er zijn voor het valide timeframe beoordeelde feedbacks aanwezig.				
Postconditie	De live scores voor het valide timeframe worden teruggegeven als response voor de huidige periode en de periode ervoor.				
Verwijzing testboom	Incident-Live-Scores-Timeframe				
Testactie(s)	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call “incident/stats/live-scores“ wordt uitgevoerd voor het timeframe Q1 en het userId 1000			

Id	SYN.T07-4				
Doel van de test	Controller of er een live scores als response komt voor het opgegeven valide timeframe.				
Preconditie	Er zijn voor het valide timeframe beoordeelde feedbacks aanwezig.				
Postconditie	De live scores voor het valide timeframe worden teruggegeven als response voor de huidige periode en de periode ervoor.				
Verwijzing testboom	Incident-Live-Scores-Timeframe				
Testactie(s)	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call “incident/stats/live-scores“ wordt uitgevoerd voor het timeframe Q2 en het userId 1000			

Id	SYN.T07-5				
Doel van de test	Controller of er een live scores als response komt voor het opgegeven valide timeframe.				
Preconditie	Er zijn voor het valide timeframe beoordeelde feedbacks aanwezig.				
Postconditie	De live scores voor het valide timeframe worden teruggegeven als response voor de huidige periode en de periode ervoor.				
Verwijzing testboom	Incident-Live-Scores-Timeframe				
Testactie(s)	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call “incident/stats/live-scores“ wordt uitgevoerd voor het timeframe Q3 en het userId 1000			

Id	SYN.T07-6				
Doel van de test	Controller of er een live scores als response komt voor het opgegeven valide timeframe.				
Preconditie	Er zijn voor het valide timeframe beoordeelde feedbacks aanwezig.				
Postconditie	De live scores voor het valide timeframe worden teruggegeven als response voor de huidige periode en de periode ervoor.				
Verwijzing testboom	Incident-Live-Scores-Timeframe				
Testactie(s)	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call “incident/stats/live-scores“ wordt uitgevoerd voor het timeframe Q4 en het userId 1000			

Id	SYN.T07-7				
Doel van de test	Controller of er een live scores als response komt voor het opgegeven valide timeframe.				
Preconditie	Er zijn voor het valide timeframe beoordeelde feedbacks aanwezig.				
Postconditie	De live scores voor het valide timeframe worden teruggegeven als response voor de huidige periode en de periode ervoor.				
Verwijzing testboom	Incident-Live-Scores-Timeframe				
Testactie(s)	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call “incident/stats/live-scores“ wordt uitgevoerd voor het timeframe YEAR en het userId 1000			

Incident-Remarks-Timeframe

Id	SYN.T08-1				
Doel van de test	Controller of er remarks als response komen voor het opgegeven valide timeframe.				
Preconditie	Er zijn voor het valide timeframe remarks aanwezig.				
Postconditie	De remarks voor het valide timeframe worden teruggegeven als response voor de huidige periode.				
Verwijzing testboom	Incident-Remarks-Timeframe				
Testactie(s)	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call “incident/stats/remarks “ wordt uitgevoerd met het timeframe DAY voor het userId 1000.			

Id	SYN.T08-2				
Doel van de test	Controller of er remarks als response komen voor het opgegeven valide timeframe.				
Preconditie	Er zijn voor het valide timeframe remarks aanwezig.				
Postconditie	De remarks voor het valide timeframe worden teruggegeven als response voor de huidige periode.				
Verwijzing testboom	Incident-Remarks-Timeframe				
Testactie(s)	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call “incident/stats/remarks “ wordt uitgevoerd met het timeframe MONTH voor het userId 1000.			

Id	SYN.T08-3				
Doel van de test	Controller of er remarks als response komen voor het opgegeven valide timeframe.				
Preconditie	Er zijn voor het valide timeframe remarks aanwezig.				
Postconditie	De remarks voor het valide timeframe worden teruggegeven als response voor de huidige periode.				
Verwijzing testboom	Incident-Remarks-Timeframe				
Testactie(s)	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call “incident/stats/remarks “ wordt uitgevoerd met het timeframe Q1 voor het userId 1000.			

Id	SYN.T08-4				
Doel van de test	Controller of er remarks als response komen voor het opgegeven valide timeframe.				
Preconditie	Er zijn voor het valide timeframe remarks aanwezig.				
Postconditie	De remarks voor het valide timeframe worden teruggegeven als response voor de huidige periode.				
Verwijzing testboom	Incident-Remarks-Timeframe				
Testactie(s)	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call “incident/stats/remarks “ wordt uitgevoerd met het timeframe Q2 voor het userId 1000.			

Id	SYN.T08-5				
Doel van de test	Controller of er remarks als response komen voor het opgegeven valide timeframe.				
Preconditie	Er zijn voor het valide timeframe remarks aanwezig.				
Postconditie	De remarks voor het valide timeframe worden teruggegeven als response voor de huidige periode.				
Verwijzing testboom	Incident-Remarks-Timeframe				
Testactie(s)	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call “incident/stats/remarks “ wordt uitgevoerd met het timeframe Q3 voor het userId 1000.			

Id	SYN.T08-6				
Doel van de test	Controller of er remarks als response komen voor het opgegeven valide timeframe.				
Preconditie	Er zijn voor het valide timeframe remarks aanwezig.				
Postconditie	De remarks voor het valide timeframe worden teruggegeven als response voor de huidige periode.				
Verwijzing testboom	Incident-Remarks-Timeframe				
Testactie(s)	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call “incident/stats/remarks “ wordt uitgevoerd met het timeframe Q4 voor het userId 1000.			

Id	SYN.T08-7				
Doel van de test	Controller of er remarks als response komen voor het opgegeven valide timeframe.				
Preconditie	Er zijn voor het valide timeframe remarks aanwezig.				
Postconditie	De remarks voor het valide timeframe worden teruggegeven als response voor de huidige periode.				
Verwijzing testboom	Incident-Remarks-Timeframe				
Testactie(s)	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call “incident/stats/remarks” wordt uitgevoerd met het timeframe YEAR voor het userId 1000.			

=====

=====

Gebruiker-Regenerate-Keys

Id	SYN.T09-1				
Doel van de test	Controleer of de consumerkey en -secret anders zijn dan voordat de test werd uitgevoerd.				
Preconditie	Een valide gebruiker is aanwezig in het systeem.				
Postconditie	De valide gebruiker heeft een andere consumerkey en -secret.				
Verwijzing testboom	Gebruiker-Regenerate-Keys				
Testactie(s)	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call “accesstoken/regenerate-keys” wordt uitgevoerd met consumerkey “XXXXXXXX” en userId 1000			

Id	SYN.T09-2				
Doel van de test	Verifieer dat er geen consumerkey en -secret worden aangepast wanneer er geen consumerkey wordt meegegeven in de aanvraag				
Preconditie	Een valide gebruiker is aanwezig in het systeem.				
Postconditie	De valide gebruiker heeft dezelfde consumerkey en -secret.				
Verwijzing testboom	Gebruiker-Regenerate-Keys				
Testactie(s)	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call “accesstoken/regenerate-keys” wordt uitgevoerd met een lege consumerkey en userId 1000	Het response object bevat een message met volgende tekst: "No valid userkey supplied!"		

Id	SYN.T09-3				
Doel van de test	Verifieer dat er geen consumerkey en -secret worden aangepast wanneer er geen valide consumerkey wordt meegegeven in de aanvraag				
Preconditie	Een valide gebruiker is aanwezig in het systeem.				
Postconditie	De valide gebruiker heeft dezelfde consumerkey en -secret.				
Verwijzing testboom	Gebruiker-Regenerate-Keys				
Testactie(s)	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call “accesstoken/regenerate-keys” wordt uitgevoerd	Het response object		

		met NULL als consumerkey en userId 1000	bevat een message met volgende tekst: "No valid userkey supplied!"		
--	--	---	--	--	--

Id	SYN.T09-4				
Doel van de test	Verifieer dat er geen consumerkey en -secret worden aangepast wanneer er geen valide userId wordt meegegeven in de aanvraag				
Preconditie	Een valide gebruiker is aanwezig in het systeem.				
Postconditie	De valide gebruiker heeft dezelfde consumerkey en -secret.				
Verwijzing testboom	Gebruiker-Regenerate-Keys				
Testactie(s)	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call "accesstoken/regenerate-keys" wordt uitgevoerd met de consumerkey "XXXXXX" en userId ABCDF	???? Kan dit eigenlijk wel??		

Id	SYN.T09-5				
Doel van de test	Verifieer dat er geen consumerkey en -secret worden aangepast wanneer er geen valide userId wordt meegegeven in de aanvraag				
Preconditie	Een valide gebruiker is aanwezig in het systeem.				
Postconditie	De valide gebruiker heeft dezelfde consumerkey en -secret.				
Verwijzing testboom	Gebruiker-Regenerate-Keys				
Testactie(s)	Testactie	Testactie	Verwachte	Conclusie	Opmerking

	Id		reactie		
	1	De call “accesstoken/regenerate-keys” wordt uitgevoerd met een lege consumerkey en userId 1000	Kan dit wel ??		

Id	SYN.T09-6				
Doel van de test	Verifieer dat er geen consumerkey en -secret worden aangepast wanneer er geen valide userId wordt meegegeven in de aanvraag				
Preconditie	Een valide gebruiker is aanwezig in het systeem.				
Postconditie	De valide gebruiker heeft dezelfde consumerkey en -secret.				
Verwijzing testboom	Gebruiker-Regenerate-Keys				
Testactie(s)	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call “accesstoken/regenerate-keys” wordt uitgevoerd met een lege consumerkey en userId 1000	kan dit wel??		

Gebruiker-Opvragen

Id	SYN.T10-1				
Doel van de test	Verifieer dat de gegevens van een valide gebruiker kunnen worden opgehaald met een valide consumerkey.				
Preconditie	De gebruiker is aanwezig in het systeem.				
Postconditie	De gegevens van de gebruiker worden teruggegeven in de response.				
Verwijzing testboom	Gebruiker-Opvragen				
Testactie(s)	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking

	1	De call GET “accesstoken/” wordt uitgevoerd met de consumerkey “XXXXXX” en userId 1000	De gegevens worden teruggegeven in de response		
--	---	--	--	--	--

Id	SYN.T10-2 KAN DEZE WEL?				
Doel van de test	Verifieer dat de gegevens van een valide gebruiker kunnen worden opgehaald met een valide consumerkey.				
Preconditie	De gebruiker is aanwezig in het systeem.				
Postconditie	De gegevens van de gebruiker worden teruggegeven in de response.				
Verwijzing testboom	Gebruiker-Opvragen				
Testactie(s)	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call GET “accesstoken/” wordt uitgevoerd met de consumerkey “XXXXXX” en userId 1000	De gegevens worden teruggegeven in de response		

Id	SYN.T10-3 KAN DEZE WEL??				
Doel van de test	Verifieer dat de gegevens van een valide gebruiker kunnen worden opgehaald met een valide consumerkey.				
Preconditie	De gebruiker is aanwezig in het systeem.				
Postconditie	De gegevens van de gebruiker worden teruggegeven in de response.				
Verwijzing testboom	Gebruiker-Opvragen				
Testactie(s)	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call GET	De gegevens		

		“accesstoken/” wordt uitgevoerd met de consumerkey “XXXXXX” en userId 1000	worden teruggegeven in de response		
--	--	--	--	--	--

Id	SYN.T10-4 KAN DEZE WEL??				
Doel van de test	Verifieer dat de gegevens van een valide gebruiker kunnen worden opgehaald met een valide consumerkey.				
Preconditie	De gebruiker is aanwezig in het systeem.				
Postconditie	De gegevens van de gebruiker worden teruggegeven in de response.				
Verwijzing testboom	Gebruiker-Opvragen				
Testactie(s)	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call GET “accesstoken/” wordt uitgevoerd met de consumerkey “XXXXXX” en userId 1000	De gegevens worden teruggegeven in de response		

Id	SYN.T10-5				
Doel van de test	Verifieer dat de er geen gegevens van een valide gebruiker kunnen worden opgehaald met een invalide consumerkey.				
Preconditie	De gebruiker is aanwezig in het systeem.				
Postconditie	Er worden geen gegevens van de gebruiker teruggegeven in de response.				
Verwijzing testboom	Gebruiker-Opvragen				
Testactie(s)	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call GET “accesstoken/”	De response bevat een		

		wordt uitgevoerd met de consumerkey 123456 en userId 1000	message met de tekst: "No valid userkey supplied!"		
--	--	---	--	--	--

Id	SYN.T10-6				
Doel van de test	Verifieer dat de er geen gegevens van een valide gebruiker kunnen worden opgehaald met een invalide consumerkey.				
Preconditie	De gebruiker is aanwezig in het systeem.				
Postconditie	Er worden geen gegevens van de gebruiker teruggegeven in de response.				
Verwijzing testboom	Gebruiker-Opvragen				
Testactie(s)	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call GET "accesstoken/" wordt uitgevoerd met de consumerkey <32CHAR en userId 1000	De response bevat een message met de tekst: "No valid userkey supplied!"		

Id	SYN.T10-7				
Doel van de test	Verifieer dat de er geen gegevens van een valide gebruiker kunnen worden opgehaald met een invalide consumerkey.				
Preconditie	De gebruiker is aanwezig in het systeem.				
Postconditie	Er worden geen gegevens van de gebruiker teruggegeven in de response.				
Verwijzing testboom	Gebruiker-Opvragen				
Testactie(s)	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call GET	De response		

		“accesstoken/” wordt uitgevoerd met de consumerkey >32CHAR en userId 1000	bevat een message met de tekst: "No valid userkey supplied!"		
--	--	---	--	--	--

Id	SYN.T10-8				
Doel van de test	Verifieer dat de er geen gegevens van een valide gebruiker kunnen worden opgehaald met een invalide consumerkey.				
Preconditie	De gebruiker is aanwezig in het systeem.				
Postconditie	Er worden geen gegevens van de gebruiker teruggegeven in de response.				
Verwijzing testboom	Gebruiker-Opvragen				
Testactie(s)	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call GET “accesstoken/” wordt uitgevoerd met de consumerkey NULL en userId 1000	De response bevat een message met de tekst: "No valid userkey supplied!"		

Operator-Opvragen

Id	SYN.T11-1
Doel van de test	Verifieer dat de gegevens van een valide operator worden opgehaald.
Preconditie	Een valide operator is aanwezig in het systeem.
Postconditie	De response bevat de gegevens van de operator.
Verwijzing testboom	Operator-Opvragen

m					
Testacties(s)	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call GET “/operator/{operatorEncId}” wordt gedaan met “09889-abd7-7882-abd7-7882” als operatorEncId	De response bevat de gegevens van de opgevraagde valide operator. <pre> operatorDao2 = new OperatorDao(); operatorDao2.setExternalId(2); operatorDao2.setFirstname("Operator 2"); operatorDao2.setLastname("Lastname 2"); operatorDao2.setEncId("09889-abd7-7882-abd7-7882"); operatorDao2.setEmail("oper@tor2.nl"); operatorDao2.setUserId(1200); operatorDao2.setActive(0); </pre>		

Id	SYN.T11-2
Doel van de test	Verifieer dat er geen gegevens van een valide operator worden opgehaald.
Preconditie	Een valide operator is aanwezig in het systeem.
Postconditie	De response bevat geen gegevens van de operator.
Verwijzing testboom	Operator-Opvragen

Testactie(s))					
	Testacti e Id	Testactie	Verwachte reactie	Conclusi e	Opmerkin g
	1	De call GET "/operator/{operatorEncId }" wordt gedaan met "dfg-67j-456-dfg-67j- 456-dfg-67j-456-dfg-67j " als operatorEncId	De response bevat de message met de tekst "OperatorEncl d is not valid!"		

Id	SYN.T11-3				
Doel van de test	Verifieer dat er geen gegevens van een valide operator worden opgehaald.				
Preconditie	Een valide operator is aanwezig in het systeem.				
Postconditi e	De response bevat geen gegevens van de operator.				
Verwijzing testboom	Operator-Opvragen				
Testactie(s))					
	Testacti e Id	Testactie	Verwachte reactie	Conclusi e	Opmerkin g
	1	De call GET "/operator/{operatorEncId }" wordt gedaan met "456-dfg-67j" als operatorEncId	De response bevat de message met de tekst "OperatorEncl d is not valid!"		

Id	SYN.T11-4				
Doel van de test	Verifieer dat er geen gegevens van een valide operator worden opgehaald.				
Preconditie	Een valide operator is aanwezig in het systeem.				
Postconditi e	De response bevat geen gegevens van de operator.				
Verwijzing	Operator-Opvragen				

testboom					
Testactie(s)					
	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call GET “/operator/{operatorEncId}” wordt gedaan met “” als operatorEncId	De response bevat de message met de tekst “OperatorEncId is not valid!”		

Id	SYN.T11-5				
Doel van de test	Verifieer dat er geen gegevens van een valide operator worden opgehaald.				
Preconditie	Een valide operator is aanwezig in het systeem.				
Postconditie	De response bevat geen gegevens van de operator.				
Verwijzing testboom	Operator-Opvragen				
Testactie(s)					
	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call GET “/operator/{operatorEncId}” wordt gedaan met NULL als operatorEncId	De response bevat de message met de tekst “OperatorEncId is not valid!”	<pre> if(operatorEncId == null) { logger.error(errorText + "EncId is NULL!"); return false; } </pre>	

Incident-Customer-Satisfaction-Timeframe

Id	SYN.T12-1
Doel van de test	Verifieer dat de customer satisfaction kan worden opgehaald met een valide TimeFrame

Preconditie	Er zijn customer satisfactions aanwezig in het systeem.				
Postconditie	De customer satisfaction van de huidige periode wordt teruggegeven en van de periode ervoor.				
Verwijzing testboom	Incident-Customer-Satisfaction-Timeframe				
Testactie(s)					
	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call “incident/stats/customer-satisfaction” wordt uitgevoerd met als TimeFrame DAY en als userId 1000.			

Id	SYN.T12-2				
Doel van de test	Verifieer dat de customer satisfaction kan worden opgehaald met een valide TimeFrame				
Preconditie	Er zijn customer satisfactions aanwezig in het systeem.				
Postconditie	De customer satisfaction van de huidige periode wordt teruggegeven en van de periode ervoor.				
Verwijzing testboom	Incident-Customer-Satisfaction-Timeframe				
Testactie(s)					
	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call “incident/stats/customer-satisfaction” wordt uitgevoerd met als TimeFrame MONTH en als userId 1000.			

Id	SYN.T12-3				
Doel van de test	Verifieer dat de customer satisfaction kan worden opgehaald met een valide TimeFrame				

Preconditie	Er zijn customer satisfactions aanwezig in het systeem.				
Postconditie	De customer satisfaction van de huidige periode wordt teruggegeven en van de periode ervoor.				
Verwijzing testboom	Incident-Customer-Satisfaction-Timeframe				
Testactie(s)	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call “incident/ stats/customer-satisfaction ” wordt uitgevoerd met als TimeFrame Q1 en als userId 1000.			

Id	SYN.T12-4				
Doel van de test	Verifieer dat de customer satisfaction kan worden opgehaald met een valide TimeFrame				
Preconditie	Er zijn customer satisfactions aanwezig in het systeem.				
Postconditie	De customer satisfaction van de huidige periode wordt teruggegeven en van de periode ervoor.				
Verwijzing testboom	Incident-Customer-Satisfaction-Timeframe				
Testactie(s)	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call “incident/ stats/customer-satisfaction ” wordt uitgevoerd met als TimeFrame Q2 en als userId 1000.			

Id	SYN.T12-5				
Doel van de test	Verifieer dat de customer satisfaction kan worden opgehaald met een valide TimeFrame				

Preconditie	Er zijn customer satisfactions aanwezig in het systeem.				
Postconditie	De customer satisfaction van de huidige periode wordt teruggegeven en van de periode ervoor.				
Verwijzing testboom	Incident-Customer-Satisfaction-Timeframe				
Testactie(s)	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call “incident/stats/customer-satisfaction” wordt uitgevoerd met als TimeFrame Q3 en als userId 1000.			

Id	SYN.T12-6				
Doel van de test	Verifieer dat de customer satisfaction kan worden opgehaald met een valide TimeFrame				
Preconditie	Er zijn customer satisfactions aanwezig in het systeem.				
Postconditie	De customer satisfaction van de huidige periode wordt teruggegeven en van de periode ervoor.				
Verwijzing testboom	Incident-Customer-Satisfaction-Timeframe				
Testactie(s)	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call “incident/stats/customer-satisfaction” wordt uitgevoerd met als TimeFrame Q4 en als userId 1000.			

Id	SYN.T12-7				
Doel van de test	Verifieer dat de customer satisfaction kan worden opgehaald met een valide TimeFrame				

Preconditie	Er zijn customer satisfactions aanwezig in het systeem.				
Postconditie	De customer satisfaction van de huidige periode wordt teruggegeven en van de periode ervoor.				
Verwijzing testboom	Incident-Customer-Satisfaction-Timeframe				
Testactie(s)	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call “incident/stats/customer-satisfaction” wordt uitgevoerd met als TimeFrame YEAR en als userId 1000.			

Id	SYN.T12-8				
Doel van de test	Verifieer dat er customer satisfaction kan worden opgehaald met een niet valide TimeFrame				
Preconditie	Er zijn customer satisfactions aanwezig in het systeem.				
Postconditie	Er wordt customer satisfaction van de huidige periode teruggegeven en van de periode ervoor met als TimeFrame DAY.				
Verwijzing testboom	Incident-Customer-Satisfaction-Timeframe				
Testactie(s)	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call “incident/stats/customer-satisfaction” wordt uitgevoerd met als TimeFrame NULL en als userId 1000.	De customer satisfaction voor TimeFrame DAY worden teruggegeven		

Id	SYN.T12-9				
Doel van de test	Verifieer dat er customer satisfaction kan worden opgehaald met een niet valide TimeFrame				

Preconditie	Er zijn customer satisfactions aanwezig in het systeem.				
Postconditie	Er wordt customer satisfaction van de huidige periode teruggegeven en van de periode ervoor met als TimeFrame DAY.				
Verwijzing testboom	Incident-Customer-Satisfaction-Timeframe				
Testactie(s)					
	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call “incident/stats/customer-satisfaction” wordt uitgevoerd met als TimeFrame 2016 en als userId 1000.	De customer satisfaction voor TimeFrame DAY worden teruggegeven.		

Id	SYN.T12-10 KAN DEZE WEL??				
Doel van de test					
Preconditie					
Postconditie					
Verwijzing testboom	Incident-Customer-Satisfaction-Timeframe				
Testactie(s)					
	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call “incident/stats/customer-satisfaction” wordt uitgevoerd met als TimeFrame Q1 en als userId 1000.			

Id	SYN.T12-11 KAN DEZE WEL??				
Doel van de test					
Preconditie					

Postconditie					
Verwijzing testboom	Incident-Customer-Satisfaction-Timeframe				
Testactie(s)					
	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1				

Id	SYN.T12-12 KAN DEZE WEL??				
Doel van de test					
Preconditie					
Postconditie					
Verwijzing testboom	Incident-Customer-Satisfaction-Timeframe				
Testactie(s)					
	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1				

Incident-Customer-Satisfaction-Relative

Id	SYN.T13-1				
Doel van de test	Verifieer dat er customer satisfaction kan worden opgehaald met een valide TimeFrame				
Preconditie	Er zijn customer satisfactions aanwezig in het systeem.				
Postconditie	Er wordt customer satisfaction van de huidige periode teruggegeven en van dezelfde periode het jaar ervoor.				
Verwijzing testboom	Incident-Customer-Satisfaction-Relative				
Testactie(s)					
	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call			

		“incident/stats/customer-satisfaction/relative-to“ wordt uitgevoerd met TimeFrame MONTH en userId 1000			
--	--	---	--	--	--

Id	SYN.T13-2				
Doel van de test	Verifieer dat er customer satisfaction kan worden opgehaald met een valide TimeFrame				
Preconditie	Er zijn customer satisfactions aanwezig in het systeem.				
Postconditie	Er wordt customer satisfaction van de huidige periode teruggegeven en van dezelfde periode het jaar ervoor.				
Verwijzing testboom	Incident-Customer-Satisfaction-Relative				
Testactie(s)	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call “incident/stats/customer-satisfaction/relative-to“ wordt uitgevoerd met TimeFrame Q1 en userId 1000			

Id	SYN.T13-3				
Doel van de test	Verifieer dat er customer satisfaction kan worden opgehaald met een valide TimeFrame				
Preconditie	Er zijn customer satisfactions aanwezig in het systeem.				
Postconditie	Er wordt customer satisfaction van de huidige periode teruggegeven en van dezelfde periode het jaar ervoor.				
Verwijzing testboom	Incident-Customer-Satisfaction-Relative				
Testactie(s)	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking

	1	De call “incident/stats/customer-satisfaction/relative-to“ wordt uitgevoerd met TimeFrame Q2 en userId 1000			
--	---	---	--	--	--

Id	SYN.T13-4				
Doel van de test	Verifieer dat er customer satisfaction kan worden opgehaald met een valide TimeFrame				
Preconditie	Er zijn customer satisfactions aanwezig in het systeem.				
Postconditie	Er wordt customer satisfaction van de huidige periode teruggegeven en van dezelfde periode het jaar ervoor.				
Verwijzing testboom	Incident-Customer-Satisfaction-Relative				
Testactie(s)	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call “incident/stats/customer-satisfaction/relative-to“ wordt uitgevoerd met TimeFrame Q3 en userId 1000			

Id	SYN.T13-5				
Doel van de test	Verifieer dat er customer satisfaction kan worden opgehaald met een valide TimeFrame				
Preconditie	Er zijn customer satisfactions aanwezig in het systeem.				
Postconditie	Er wordt customer satisfaction van de huidige periode teruggegeven en van dezelfde periode het jaar ervoor.				
Verwijzing testboom	Incident-Customer-Satisfaction-Relative				
Testactie(s)	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking

	1	De call “incident/stats/customer-satisfaction/relative-to“ wordt uitgevoerd met TimeFrame Q4 en userId 1000			
--	---	---	--	--	--

Id	SYN.T13-6				
Doel van de test	Verifieer dat er customer satisfaction kan worden opgehaald met een valide TimeFrame				
Preconditie	Er zijn customer satisfactions aanwezig in het systeem.				
Postconditie	Er wordt customer satisfaction van de huidige periode teruggegeven en van dezelfde periode het jaar ervoor.				
Verwijzing testboom	Incident-Customer-Satisfaction-Relative				
Testactie(s)	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call “incident/stats/customer-satisfaction/relative-to“ wordt uitgevoerd met TimeFrame YEAR en userId 1000			

Id	SYN.T13-7				
Doel van de test	Verifieer dat er customer satisfaction kan worden opgehaald met een niet valide TimeFrame				
Preconditie	Er zijn customer satisfactions aanwezig in het systeem.				
Postconditie	Er wordt customer satisfaction van de huidige Maand teruggegeven en van dezelfde maand het jaar ervoor.				
Verwijzing testboom	Incident-Customer-Satisfaction-Relative				
Testactie(s)	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking

	1	De call “incident/stats/customer-satisfaction/relative-to” wordt uitgevoerd met TimeFrame NULL en userId 1000			
--	---	---	--	--	--

Id	SYN.T13-8				
Doel van de test	Verifieer dat er customer satisfaction kan worden opgehaald met een niet valide TimeFrame				
Preconditie	Er zijn customer satisfactions aanwezig in het systeem.				
Postconditie	Er wordt customer satisfaction van de huidige Maand teruggegeven en van dezelfde maand het jaar ervoor.				
Verwijzing testboom	Incident-Customer-Satisfaction-Relative				
Testactie(s)	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call “incident/stats/customer-satisfaction/relative-to” wordt uitgevoerd met TimeFrame 2016 en userId 1000			

Id	SYN.T13-9				
Doel van de test	Verifieer dat er customer satisfaction kan worden opgehaald met een niet valide TimeFrame				
Preconditie	Er zijn customer satisfactions aanwezig in het systeem.				
Postconditie	Er wordt customer satisfaction van de huidige Maand teruggegeven en van dezelfde maand het jaar ervoor.				
Verwijzing testboom	Incident-Customer-Satisfaction-Relative				
Testactie(s)	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking

	1	De call “incident/stats/customer- satisfaction/relative-to” wordt uitgevoerd met TimeFrame DAY en userId 1000			
--	---	--	--	--	--

Id	SYN.T13-10 KAN DEZE WEL??				
Doel van de test	Verifieer dat er customer satisfaction kan worden opgehaald met een valide TimeFrame				
Preconditie	Er zijn customer satisfactions aanwezig in het systeem.				
Postconditie	Er wordt customer satisfaction van de huidige periode teruggegeven en van dezelfde periode het jaar ervoor.				
Verwijzing testboom	Incident-Customer-Satisfaction-Relative				
Testactie(s)					
	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1				

Id	SYN.T13-11 KAN DEZE WEL??				
Doel van de test	Verifieer dat er customer satisfaction kan worden opgehaald met een valide TimeFrame				
Preconditie	Er zijn customer satisfactions aanwezig in het systeem.				
Postconditie	Er wordt customer satisfaction van de huidige periode teruggegeven en van dezelfde periode het jaar ervoor.				
Verwijzing testboom	Incident-Customer-Satisfaction-Relative				
Testactie(s)					
	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1				

Id	SYN.T13-12 KAN DEZE WEL??				
Doel van de test	Verifieer dat er customer satisfaction kan worden opgehaald met een valide TimeFrame				

Preconditie	Er zijn customer satisfactions aanwezig in het systeem.				
Postconditie	Er wordt customer satisfaction van de huidige periode teruggegeven en van dezelfde periode het jaar ervoor.				
Verwijzing testboom	Incident-Customer-Satisfaction-Relative				
Testactie(s)					
	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1				

Incident-Operator-Scores-Timeframe

Id	SYN.T14-1				
Doel van de test	Verifieer dat er operator scores kunnen worden opgehaald met een valide TimeFrame.				
Preconditie	Er zijn operator scores aanwezig in het systeem.				
Postconditie	Er worden operator scores van de huidige periode en dezelfde periode ervoor teruggegeven.				
Verwijzing testboom	Incident-Operator-Scores-Timeframe				
Testactie(s)					
	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call “incident/stats/operator-scores” wordt gedaan met TimeFrame DAY en userId 1000			

Id	SYN.T14-2				
Doel van de test	Verifieer dat er operator scores kunnen worden opgehaald met een valide TimeFrame.				
Preconditie	Er zijn operator scores aanwezig in het systeem.				
Postconditie	Er worden operator scores van de huidige periode en dezelfde periode ervoor teruggegeven.				
Verwijzing	Incident-Operator-Scores-Timeframe				

testboom					
Testactie(s)					
	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call “incident/stats/operator-scores” wordt gedaan met TimeFrame MONTH en userId 1000			

Id	SYN.T14-3				
Doel van de test	Verifieer dat er operator scores kunnen worden opgehaald met een valide TimeFrame.				
Preconditie	Er zijn operator scores aanwezig in het systeem.				
Postconditie	Er worden operator scores van de huidige periode en dezelfde periode ervoor teruggegeven.				
Verwijzing testboom	Incident-Operator-Scores-Timeframe				
Testactie(s)					
	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call “incident/stats/operator-scores” wordt gedaan met TimeFrame Q1 en userId 1000			

Id	SYN.T14-4				
Doel van de test	Verifieer dat er operator scores kunnen worden opgehaald met een valide TimeFrame.				
Preconditie	Er zijn operator scores aanwezig in het systeem.				
Postconditie	Er worden operator scores van de huidige periode en dezelfde periode ervoor teruggegeven.				
Verwijzing testboom	Incident-Operator-Scores-Timeframe				

Testactie(s)					
	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call “incident/ stats/operator-scores ” wordt gedaan met TimeFrame Q2 en userId 1000			

Id	SYN.T14-5				
Doel van de test	Verifieer dat er operator scores kunnen worden opgehaald met een valide TimeFrame.				
Preconditie	Er zijn operator scores aanwezig in het systeem.				
Postconditie	Er worden operator scores van de huidige periode en dezelfde periode ervoor teruggegeven.				
Verwijzing testboom	Incident-Operator-Scores-Timeframe				
Testactie(s)					
	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call “incident/ stats/operator-scores ” wordt gedaan met TimeFrame Q3 en userId 1000			

Id	SYN.T14-6				
Doel van de test	Verifieer dat er operator scores kunnen worden opgehaald met een valide TimeFrame.				
Preconditie	Er zijn operator scores aanwezig in het systeem.				
Postconditie	Er worden operator scores van de huidige periode en dezelfde periode ervoor teruggegeven.				
Verwijzing testboom	Incident-Operator-Scores-Timeframe				
Testactie(s)					

	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call “incident/ stats/operator-scores ” wordt gedaan met TimeFrame Q4 en userId 1000			

Id	SYN.T14-7				
Doel van de test	Verifieer dat er operator scores kunnen worden opgehaald met een valide TimeFrame.				
Preconditie	Er zijn operator scores aanwezig in het systeem.				
Postconditie	Er worden operator scores van de huidige periode en dezelfde periode ervoor teruggegeven.				
Verwijzing testboom	Incident-Operator-Scores-Timeframe				
Testactie(s)	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call “incident/ stats/operator-scores ” wordt gedaan met TimeFrame YEAR en userId 1000			

Id	SYN.T14-8				
Doel van de test	Verifieer dat er operator scores kunnen worden opgehaald met een valide TimeFrame.				
Preconditie	Er zijn operator scores aanwezig in het systeem.				
Postconditie	Er worden operator scores van de huidige dag en de dag ervoor teruggegeven.				
Verwijzing testboom	Incident-Operator-Scores-Timeframe				
Testactie(s)	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking

	1	De call “incident/stats/operator-scores” wordt gedaan met TimeFrame NULL en userId 1000			
--	---	--	--	--	--

Id	SYN.T14-9				
Doel van de test	Verifieer dat er operator scores kunnen worden opgehaald met een valide TimeFrame.				
Preconditie	Er zijn operator scores aanwezig in het systeem.				
Postconditie	Er worden operator scores van de huidige dag en de dag ervoor teruggegeven.				
Verwijzing testboom	Incident-Operator-Scores-Timeframe				
Testactie(s)	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call “incident/stats/operator-scores” wordt gedaan met TimeFrame 2016 en userId 1000			

Id	SYN.T14-10 KAN DEZE WEL??				
Doel van de test	Verifieer dat er operator scores kunnen worden opgehaald met een valide TimeFrame.				
Preconditie	Er zijn operator scores aanwezig in het systeem.				
Postconditie	Er worden operator scores van de huidige periode en dezelfde periode ervoor teruggegeven.				
Verwijzing testboom	Incident-Operator-Scores-Timeframe				
Testactie(s)	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call “incident/stats/operator-			

		scores” wordt gedaan met TimeFrame DAY en userId 1000			
--	--	---	--	--	--

Id	SYN.T14-11 KAN DEZE WEL??				
Doel van de test	Verifieer dat er operator scores kunnen worden opgehaald met een valide TimeFrame.				
Preconditie	Er zijn operator scores aanwezig in het systeem.				
Postconditie	Er worden operator scores van de huidige periode en dezelfde periode ervoor teruggegeven.				
Verwijzing testboom	Incident-Operator-Scores-Timeframe				
Testactie(s)	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call “incident/stats/operator-scores” wordt gedaan met TimeFrame DAY en userId 1000			

Id	SYN.T14-12 KAN DEZE WEL??				
Doel van de test	Verifieer dat er operator scores kunnen worden opgehaald met een valide TimeFrame.				
Preconditie	Er zijn operator scores aanwezig in het systeem.				
Postconditie	Er worden operator scores van de huidige periode en dezelfde periode ervoor teruggegeven.				
Verwijzing testboom	Incident-Operator-Scores-Timeframe				
Testactie(s)	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call “incident/stats/operator-scores” wordt gedaan met			

		TimeFrame DAY en userId 1000			
--	--	---------------------------------	--	--	--

Incident-Operator-Scores-Relative

Id	SYN.T15-1				
Doel van de test	Verifieer dat er operator scores kunnen worden opgehaald met een valide TimeFrame van de huidige periode en dezelfde periode het jaar ervoor.				
Preconditie	Er zijn operator scores aanwezig in het systeem.				
Postconditie	Er worden operator scores van de huidige periode en dezelfde periode het jaar ervoor teruggegeven.				
Verwijzing testboom	Incident-Operator-Scores-Relative				
Testactie(s)	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call “incident/stats/operator-scores/relative-to” wordt uitgevoerd met als TimeFrame MONTH en als userId 1000			

Id	SYN.T15-2				
Doel van de test	Verifieer dat er operator scores kunnen worden opgehaald met een valide TimeFrame van de huidige periode en dezelfde periode het jaar ervoor.				
Preconditie	Er zijn operator scores aanwezig in het systeem.				
Postconditie	Er worden operator scores van de huidige periode en dezelfde periode het jaar ervoor teruggegeven.				
Verwijzing testboom	Incident-Operator-Scores-Relative				
Testactie(s)	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call			

		“incident/stats/operator-scores/relative-to“ wordt uitgevoerd met als TimeFrame Q1 en als userId 1000			
--	--	---	--	--	--

Id	SYN.T15-3				
Doel van de test	Verifieer dat er operator scores kunnen worden opgehaald met een valide TimeFrame van de huidige periode en dezelfde periode het jaar ervoor.				
Preconditie	Er zijn operator scores aanwezig in het systeem.				
Postconditie	Er worden operator scores van de huidige periode en dezelfde periode het jaar ervoor teruggegeven.				
Verwijzing testboom	Incident-Operator-Scores-Relative				
Testactie(s)	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call “incident/stats/operator-scores/relative-to“ wordt uitgevoerd met als TimeFrame Q2 en als userId 1000			

Id	SYN.T15-4				
Doel van de test	Verifieer dat er operator scores kunnen worden opgehaald met een valide TimeFrame van de huidige periode en dezelfde periode het jaar ervoor.				
Preconditie	Er zijn operator scores aanwezig in het systeem.				
Postconditie	Er worden operator scores van de huidige periode en dezelfde periode het jaar ervoor teruggegeven.				
Verwijzing testboom	Incident-Operator-Scores-Relative				
Testactie(s)	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking

	1	De call “incident/stats/operator-scores/relative-to” wordt uitgevoerd met als TimeFrame Q3 en als userId 1000			
--	---	--	--	--	--

Id	SYN.T15-5				
Doel van de test	Verifieer dat er operator scores kunnen worden opgehaald met een valide TimeFrame van de huidige periode en dezelfde periode het jaar ervoor.				
Preconditie	Er zijn operator scores aanwezig in het systeem.				
Postconditie	Er worden operator scores van de huidige periode en dezelfde periode het jaar ervoor teruggegeven.				
Verwijzing testboom	Incident-Operator-Scores-Relative				
Testactie(s)	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call “incident/stats/operator-scores/relative-to” wordt uitgevoerd met als TimeFrame Q4 en als userId 1000			

Id	SYN.T15-6				
Doel van de test	Verifieer dat er operator scores kunnen worden opgehaald met een valide TimeFrame van de huidige periode en dezelfde periode het jaar ervoor.				
Preconditie	Er zijn operator scores aanwezig in het systeem.				
Postconditie	Er worden operator scores van de huidige periode en dezelfde periode het jaar ervoor teruggegeven.				
Verwijzing testboom	Incident-Operator-Scores-Relative				
Testactie(s)	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking

	1	De call “incident/stats/operator-scores/relative-to” wordt uitgevoerd met als TimeFrame YEAR en als userId 1000			
--	---	--	--	--	--

Id	SYN.T15-7				
Doel van de test	Verifieer dat er operator scores kunnen worden opgehaald met een valide TimeFrame van de huidige periode en dezelfde periode het jaar ervoor.				
Preconditie	Er zijn operator scores aanwezig in het systeem.				
Postconditie	Er worden operator scores van de huidige maand en dezelfde maand het jaar ervoor teruggegeven.				
Verwijzing testboom	Incident-Operator-Scores-Relative				
Testactie(s)	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call “incident/stats/operator-scores/relative-to” wordt uitgevoerd met als TimeFrame NULL en als userId 1000			

Id	SYN.T15-8				
Doel van de test	Verifieer dat er operator scores kunnen worden opgehaald met een valide TimeFrame van de huidige periode en dezelfde periode het jaar ervoor.				
Preconditie	Er zijn operator scores aanwezig in het systeem.				
Postconditie	Er worden operator scores van de huidige maand en dezelfde maand het jaar ervoor teruggegeven.				
Verwijzing testboom	Incident-Operator-Scores-Relative				
Testactie(s)	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking

	1	De call “incident/stats/operator-scores/relative-to” wordt uitgevoerd met als TimeFrame 2016 en als userId 1000			
--	---	--	--	--	--

Id	SYN.T15-9				
Doel van de test	Verifieer dat er operator scores kunnen worden opgehaald met een valide TimeFrame van de huidige periode en dezelfde periode het jaar ervoor.				
Preconditie	Er zijn operator scores aanwezig in het systeem.				
Postconditie	Er worden operator scores van de huidige maand en dezelfde maand het jaar ervoor teruggegeven.				
Verwijzing testboom	Incident-Operator-Scores-Relative				
Testactie(s)	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call “incident/stats/operator-scores/relative-to” wordt uitgevoerd met als TimeFrame DAY en als userId 1000			

Id	SYN.T15-10 KAN DEZE WEL??				
Doel van de test	Verifieer dat er operator scores kunnen worden opgehaald met een valide TimeFrame van de huidige periode en dezelfde periode het jaar ervoor.				
Preconditie	Er zijn operator scores aanwezig in het systeem.				
Postconditie	Er worden operator scores van de huidige periode en dezelfde periode het jaar ervoor teruggegeven.				
Verwijzing testboom	Incident-Operator-Scores-Relative				
Testactie(s)					
	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking

	1				
--	---	--	--	--	--

Id	SYN.T15-11 KAN DEZE WEL??				
Doel van de test	Verifieer dat er operator scores kunnen worden opgehaald met een valide TimeFrame van de huidige periode en dezelfde periode het jaar ervoor.				
Preconditie	Er zijn operator scores aanwezig in het systeem.				
Postconditie	Er worden operator scores van de huidige periode en dezelfde periode het jaar ervoor teruggegeven.				
Verwijzing testboom	Incident-Operator-Scores-Relative				
Testactie(s)					
	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1				

Id	SYN.T15-12 KAN DEZE WEL??				
Doel van de test	Verifieer dat er operator scores kunnen worden opgehaald met een valide TimeFrame van de huidige periode en dezelfde periode het jaar ervoor.				
Preconditie	Er zijn operator scores aanwezig in het systeem.				
Postconditie	Er worden operator scores van de huidige periode en dezelfde periode het jaar ervoor teruggegeven.				
Verwijzing testboom	Incident-Operator-Scores-Relative				
Testactie(s)					
	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1				

Incident-Opvragen

Id	SYN.T16-1				
Doel van de test	Verifieer dat er gegevens van een incident kunnen worden opgevraagd voor een valide feedbackEncId.				
Preconditie	Er is een feedback aanwezig voor het valide feedbackEncId.				

Postconditie	De informatie voor in de feedback mail wordt teruggegeven voor het gegeven feedbackEncId.				
Verwijzing testboom	Incident-Opvragen				
Testactie(s)					
	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call GET "incident/{feedbackEncId}" wordt aangeroepen met als feedbackEncId "XXXXX" en als userId 1000			

Id	SYN.T16-2				
Doel van de test	Verifieer dat er geen gegevens van een incident kunnen worden opgevraagd voor een niet valide feedbackEncId.				
Preconditie	Er is een feedback aanwezig voor het valide feedbackEncId.				
Postconditie	Er wordt een foutmelding gegeven.				
Verwijzing testboom	Incident-Opvragen				
Testactie(s)					
	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call GET "incident/{feedbackEncId}" wordt aangeroepen met als feedbackEncId ">35CHAR" en als userId 1000	Response bevat een message met de tekst: "No valid encId supplied!"		

Id	SYN.T16-3				
Doel van de test	Verifieer dat er geen gegevens van een incident kunnen worden opgevraagd voor een niet valide feedbackEncId.				

Preconditie	Er is een feedback aanwezig voor het valide feedbackEncId.				
Postconditie	Er wordt een foutmelding gegeven.				
Verwijzing testboom	Incident-Opvragen				
Testactie(s)	Testacti e Id	Testactie	Verwacht e reactie	Conclusi e	Opmerkin g
	1	De call GET "incident/{feedbackEncId} " wordt aangeroepen met als feedbackEncId "<35CHAR" en als userId 1000	Response bevat een message met de tekst: "No valid encId supplied!"		

Id	SYN.T16-4				
Doel van de test	Verifieer dat er geen gegevens van een incident kunnen worden opgevraagd voor een niet valide feedbackEncId.				
Preconditie	Er is een feedback aanwezig voor het valide feedbackEncId.				
Postconditie	Er wordt een foutmelding gegeven.				
Verwijzing testboom	Incident-Opvragen				
Testactie(s)	Testacti e Id	Testactie	Verwacht e reactie	Conclusi e	Opmerkin g
	1	De call GET "incident/{feedbackEncId} " wordt aangeroepen met als feedbackEncId "" en als userId 1000	Response bevat een message met de tekst: "No valid encId supplied!"		

Id	SYN.T16-5				
----	-----------	--	--	--	--

Doel van de test	Verifieer dat er geen gegevens van een incident kunnen worden opgevraagd voor een niet valide feedbackEncId.				
Preconditie	Er is een feedback aanwezig voor het valide feedbackEncId.				
Postconditie	Er wordt een foutmelding gegeven.				
Verwijzing testboom	Incident-Opvragen				
Testactie(s)					
	Testacti e Id	Testactie	Verwacht e reactie	Conclusi e	Opmerkin g
	1	De call GET "incident/{feedbackEncId} " wordt aangeroepen met als feedbackEncId "NULL" en als userId 1000	Response bevat een message met de tekst: "No valid encId supplied!"		

Incident-Versturen

Id	SYN.T17-1				
Doel van de test	Verifieer dat er een feedback mail van het valide feedbackEncId is verzonden aan de beoordelaar.				
Preconditie	Er is een feedback aanwezig voor het valide feedbackEncId.				
Postconditie	De mail is verzonden aan de beoordelaar.				
Verwijzing testboom	Incident-Versturen				
Testactie(s))					
	Testacti e Id	Testactie	Verwacht e reactie	Conclusi e	Opmerkin g
	1	De call "incident/{feedbackEncId}/sen d" wordt aangeroepen met feedbackEncId "XXXXXX" en als userId 1000	De response heeft een message met de tekst:		

			"Feedback mail send for encId: " + XXXXX X + "."		
--	--	--	--	--	--

Id	SYN.T17-2				
Doel van de test	Verifieer dat er geen feedback mail van het valide feedbackEncId is verzonden aan de beoordelaar.				
Preconditie	Er is een feedback aanwezig voor het valide feedbackEncId.				
Postconditie	De mail is niet verzonden aan de beoordelaar.				
Verwijzing testboom	Incident-Versturen				
Testactie(s)	Testacti e Id	Testactie	Verwacht e reactie	Conclusi e	Opmerkin g
	1	De call "incident/{feedbackEncId}/send" wordt aangeroepen met feedbackEncId ">35CHAR" en als userId 1000	De response heeft een message met de tekst: "No valid encId supplied!"		

Id	SYN.T17-3				
Doel van de test	Verifieer dat er geen feedback mail van het valide feedbackEncId is verzonden aan de beoordelaar.				
Preconditie	Er is een feedback aanwezig voor het valide feedbackEncId.				
Postconditie	De mail is niet verzonden aan de beoordelaar.				
Verwijzing	Incident-Versturen				

testboom					
Testactie(s))					
	Testacti e Id	Testactie	Verwacht e reactie	Conclusi e	Opmerkin g
	1	De call “incident/{feedbackEncId}/sen d“ wordt aangeroepen met feedbackEncId “<35CHAR” en als userId 1000	De response heeft een message met de tekst: "No valid encId supplied!"		

Id	SYN.T17-4				
Doel van de test	Verifieer dat er geen feedback mail van het valide feedbackEncId is verzonden aan de beoordelaar.				
Preconditie	Er is een feedback aanwezig voor het valide feedbackEncId.				
Postconditie	De mail is niet verzonden aan de beoordelaar.				
Verwijzing testboom	Incident-Versturen				
Testactie(s))					
	Testacti e Id	Testactie	Verwacht e reactie	Conclusi e	Opmerkin g
	1	De call “incident/{feedbackEncId}/sen d“ wordt aangeroepen met feedbackEncId “” en als userId 1000	De response heeft een message met de tekst: "No valid encId supplied!"		

Id	SYN.T17-5				
Doel van de test	Verifieer dat er geen feedback mail van het valide feedbackEncId is verzonden aan de beoordelaar.				
Preconditie	Er is een feedback aanwezig voor het valide feedbackEncId.				
Postconditie	De mail is niet verzonden aan de beoordelaar.				
Verwijzing testboom	Incident-Versturen				
Testactie(s)					
	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call "incident/{feedbackEncId}/send" wordt aangeroepen met feedbackEncId "NULL" en als userId 1000	De response heeft een message met de tekst: "No valid encId supplied!"		

Gebruiker-Toevoegen

Id	SYN.T18-1				
Doel van de test	Controleer dat er een nieuwe gebruiker is toegevoegd.				
Preconditie	De gebruiker is niet aanwezig in het systeem.				
Postconditie	De gebruiker is aanwezig in het systeem.				
Verwijzing testboom	Gebruiker-Toevoegen				
Testactie(s)					
	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call POST "accesstoken" wordt	De gebruiker is		

		uitgevoerd met de body: "XXXXXX" en userId 1000.	aangemaakt.		
--	--	--	-------------	--	--

Id	SYN.T18-2				
Doel van de test	Controleer dat er geen nieuwe gebruiker is toegevoegd.				
Preconditie	De gebruiker is niet aanwezig in het systeem.				
Postconditie	De gebruiker is niet aanwezig in het systeem.				
Verwijzing testboom	Gebruiker-Toevoegen				
Testactie(s)	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call POST "accesstoken" wordt uitgevoerd met de body: "" en userId 1000.	De response bevat een message met de tekst: "User not valid!"		

Id	SYN.T18-3				
Doel van de test	Controleer dat er geen nieuwe gebruiker is toegevoegd.				
Preconditie	De gebruiker is niet aanwezig in het systeem.				
Postconditie	De gebruiker is niet aanwezig in het systeem.				
Verwijzing testboom	Gebruiker-Toevoegen				
Testactie(s)	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call POST "accesstoken" wordt uitgevoerd met de body: "NULL" en userId 1000.	De response bevat een message met de tekst: "User not valid!"		

Id	SYN.T18-4 KAN DEZE WEL??				
Doel van de test	Controleer dat er geen nieuwe gebruiker is toegevoegd.				
Preconditie	De gebruiker is niet aanwezig in het systeem.				
Postconditie	De gebruiker is niet aanwezig in het systeem.				
Verwijzing testboom	Gebruiker-Toevoegen				
Testactie(s)	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call POST “accesstoken” wordt uitgevoerd met de body: “XXXXXX” en userId 1000.	De gebruiker is aangemaakt.		

Id	SYN.T18-5 KAN DEZE WEL??				
Doel van de test	Controleer dat er geen nieuwe gebruiker is toegevoegd.				
Preconditie	De gebruiker is niet aanwezig in het systeem.				
Postconditie	De gebruiker is niet aanwezig in het systeem.				
Verwijzing testboom	Gebruiker-Toevoegen				
Testactie(s)	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call POST “accesstoken” wordt uitgevoerd met de body: “XXXXXX” en userId 1000.	De gebruiker is aangemaakt.		

Id	SYN.T18-6 KAN DEZE WEL??				
Doel van de	Controleer dat er geen nieuwe gebruiker is toegevoegd.				

test					
Preconditie	De gebruiker is niet aanwezig in het systeem.				
Postconditie	De gebruiker is niet aanwezig in het systeem.				
Verwijzing testboom	Gebruiker-Toevoegen				
Testactie(s)					
	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call POST “accesstoken” wordt uitgevoerd met de body: “XXXXXX” en userId 1000.	De gebruiker is aangemaakt.		

Operator-Update

Id	SYN.T19-1				
Doel van de test	Verifieer dat een operator kan worden aangepast.				
Preconditie	Er is een valide operator aanwezig.				
Postconditie	De operator is aangepast.				
Verwijzing testboom	Operator-Update				
Testactie(s)					
	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call PUT “/operator” wordt uitgevoerd met als body: “dao1 + “nieuwe voornaam”” en als userId 1000.	De operator is aangepast met de nieuwe gegevens.		

Id	SYN.T19-2				
Doel van de	Verifieer dat een operator niet kan worden aangepast met een NULL body.				

test					
Preconditie	Er is een valide operator aanwezig.				
Postconditie	De operator is niet aangepast.				
Verwijzing testboom	Operator-Updates				
Testacties(s)	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call PUT "/operator" wordt uitgevoerd met als body: "NULL" en als userId 1000.	De operator is niet aangepast met de nieuwe gegevens. assertThat(operatorResponse.getMessage(), is("Operator not valid!"));	<pre> if(operator == null) { logger.error(operatorErrorText + "Body is null!"); return false; } </pre>	

Id	SYN.T19-3
Doel van de test	Verifieer dat een operator niet kan worden aangepast met niet niet valide operator.
Preconditie	Er is een valide operator aanwezig.
Postconditie	De operator is niet aangepast.
Verwijzing testboom	Operator-Updates
Testactie(

s)	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call PUT “/operator” wordt uitgevoerd met als body: “NIET VALIDE” en als userId 1000.	De operator is niet aangepast met de nieuwe gegevens. assertThat(operatorResponse.getMessage(), is("Operator not valid!"));		

Operator-Toevoegen

Id	SYN.T20-1				
Doel van de test	Verifieer dat er een nieuwe operator is toegevoegd.				
Preconditie	De operator is nog niet aanwezig in het systeem.				
Postconditie	De operator is aanwezig in het systeem.				
Verwijzing testboom	Operator-Toevoegen				
Testactie(s)	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call POST “/operator” wordt uitgevoerd met de body: operatorDao2.setExternalId (2);	De operator is toegevoegd. operatorDao1.setExternalId (1); operatorDao1.setFir		

	<pre> operatorDao2.setFirstName("Operator 2"); operatorDao2.setLastName("Lastname 2"); operatorDao2.setEncId("09889-abd7-7882-abd7-7882-abd7-7882"); operatorDao2.setEmail("oper@tor2.nl"); operatorDao2.setUserId(1200); operatorDao2.setActive(0); " en userId 1000. </pre>	<pre> sname("Operator 1"); operatorDao1.setLastName("Lastname"); operatorDao1.setEncId("345-0-09df-3209-09df-3209-09df-3209"); operatorDao1.setEmail("oper@tor.nl"); operatorDao1.setUserId(1100); operatorDao1.setActive(1); </pre>		
--	---	---	--	--

Id	SYN.T20-2				
Doel van de test	Verifieer dat er geen nieuwe operator is toegevoegd.				
Preconditie	De operator is nog niet aanwezig in het systeem.				
Postconditie	De operator is nog niet aanwezig in het systeem.				
Verwijzing testboom	Operator-Toevoegen				
Testactie(s)					
	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call POST "/operator" wordt uitgevoerd met de body: "NULL" en userId 1000.	De response bevat een message met de tekst: "Could not add new operator!"		

Id	SYN.T20-3
Doel van	Verifieer dat er geen nieuwe operator is toegevoegd.

de test					
Preconditie	De operator is nog niet aanwezig in het systeem.				
Postconditie	De operator is nog niet aanwezig in het systeem.				
Verwijzing testboom	Operator-Toevoegen				
Testactie(s)	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call POST “/operator” wordt uitgevoerd met de body: <pre> operatorDao1.setExternalId(1); operatorDao1.setLastname("Lastname"); operatorDao1.setEmail("operator.nl"); operatorDao1.setUserId(1100) ; operatorDao1.setActive(1); ” en userId 1000. </pre>	De respons bevat een message met de tekst: "Could not add new operator!"		

Incident-Toevoegen

Id	SYN.T21-1
Doel van de test	Verifieer dat er een nieuw incident is aangemaakt.
Preconditie	Het incident is nog niet aanwezig in het systeem.
Postconditie	Het incident is aanwezig in het systeem.
Verwijzing testboom	Incident-Toevoegen

Testactie(s)					
	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call POST “/incident” wordt uitgevoerd met de body: “XXXXXX” en userId 1000.	Het incident is toegevoegd.		

Id	SYN.T21-2				
Doel van de test	Verifieer dat er geen nieuw incident is aangemaakt.				
Preconditie	Het incident is nog niet aanwezig in het systeem.				
Postconditie	Het incident is niet toegevoegd aan het systeem.				
Verwijzing testboom	Incident-Toevoegen				
Testactie(s)					
	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call POST “/incident” wordt uitgevoerd met de body: “NULL” en userId 1000.	De response bevat een message met de tekst: "request is not valid, there is no feedback created."		

Id	SYN.T21-3				
Doel van de test	Verifieer dat er geen nieuw incident is aangemaakt.				
Preconditie	Het incident is nog niet aanwezig in het systeem.				
Postconditie	Het incident is niet toegevoegd aan het systeem.				
Verwijzing testboom	Incident-Toevoegen				
Testactie(s)					
	Testactie	Testactie	Verwachte reactie	Conclusie	Opmerking

	Id				
	1	De call POST “/incident” wordt uitgevoerd met de body: “NIET VALIDE” en userId 1000.	De response bevat een message met de tekst: "request is not valid, there is no feedback created."		

Incident-Updaten

Id	SYN.T22-1				
Doel van de test	Verifieer dat het incident met valide incidentEncId is aangepast.				
Preconditie	Er is een feedback aanwezig voor het incidentEncId.				
Postconditie	De feedback is aangepast.				
Verwijzing testboom	Incident-Updaten				
Testactie(s)	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call PUT “/incident” wordt uitgevoerd met de body: “XXXXXX” en userId 1000.	Het incident is aangepast en het incidentEncId wordt teruggegeven.		

Id	SYN.T22-1				
Doel van de test	Verifieer dat het incident met valide incidentEncId is aangepast.				
Preconditie	Er is een feedback aanwezig voor het incidentEncId.				
Postconditie	De feedback is aangepast.				
Verwijzing testboom	Incident-Updaten				

Testactie(s)					
	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call PUT “/incident” wordt uitgevoerd met de body: “NULL” en userId 1000.	Het incident is niet aangepast en het incidentEncId wordt teruggegeven.		

Id	SYN.T22-3				
Doel van de test	Verifieer dat het incident met valide incidentEncId niet is aangepast.				
Preconditie	Er is een feedback aanwezig voor het incidentEncId.				
Postconditie	De feedback is niet aangepast.				
Verwijzing testboom	Incident-Updates				
Testactie(s)					
	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call PUT “/incident” wordt uitgevoerd met de body: “NIET VALIDE” en userId 1000.	Het incident is niet aangepast en het incidentEncId wordt teruggegeven.		

Algoritme Test : Statement Covarage

Gebruiker-Regenerate-Keys

Id	AT.T01-1
Doel van de test	Succesvol genereren van nieuwe consumerKey en -Secret.

Preconditie	Er is een valide gebruiker aanwezig.				
Postconditie	De gebruiker heeft een nieuwe consumerKey en -Secret.				
Paden	1-3-4-5-6-8				
Verwijzing testboom	Gebruiker-Regenerate-Keys				
Testactie(s)					
	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	regenerateKeys wordt uitgevoerd met een valide user en als userId 1000.	Een response met de nieuwe waarden voor deze gebruiker.		

Id	AT.T01-2				
Doel van de test	De validatie van de consumerKey geeft een foutmelding, omdat deze niet valide is.				
Preconditie	Er is een valide gebruiker aanwezig.				
Postconditie	Het systeem geeft aan dat de consumerKey niet valide is.				
Paden	2-10				
Verwijzing testboom	Gebruiker-Regenerate-Keys				
Testactie(s)					
	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	regenerateKeys wordt uitgevoerd met een niet valide user en als userId 1000. De consumerKey is "<35CHAR"	De response bevat een message met de tekst: "No valid userkey supplied!"		

Id	AT.T01-3				
Doel van de test	Update geeft een foutmelding, omdat de update in de database verkeerd gaat.				

Preconditie	Er is een valide gebruiker aanwezig.				
Postconditie	De gebruiker heeft geen nieuwe consumerKey en -Secret.				
Paden	1-3-4-5-7-9				
Verwijzing testboom	Gebruiker-Regenerate-Keys				
Testactie(s)	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	regenerateKeys wordt uitgevoerd met een valide gebruiker en als userId 1000.	De response bevat een message met de tekst: "Could not create new key and secret!"		

Gebruiker-Opvragen

Id	AT.T02-1				
Doel van de test	Succesvol verkrijgen van gebruiker.				
Preconditie	Er is een valide gebruiker aanwezig in het systeem.				
Postconditie	De response bevat de gegevens van de opgevraagde gebruiker				
Paden	1-3-4				
Verwijzing testboom	Gebruiker-Opvragen				
Testactie(s)	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	getUser wordt uitgevoerd met een valide gebruiker en als userId 1000.	De response bevat de gegevens van de opgevraagde gebruiker.		

Id	AT.T02-2
----	----------

Doel van de test	Validate consumerKey geeft een foutmelding, omdat deze niet valide is.				
Preconditie	Er is een valide gebruiker aanwezig in het systeem.				
Postconditie	De response geeft een foutmelding.				
Paden	2-5				
Verwijzing testboom	Gebruiker-Opvragen				
Testactie(s)					
	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	getUser wordt uitgevoerd met een niet valide gebruiker en als userId 1000. Als consumerKey wordt "<35CHAR" meegegeven.	De response bevat een message met de tekst: "No valid userkey supplied!"		

Operator-Opvragen

Id	AT.T03-1				
Doel van de test	Succesvol verkrijgen van een Operator				
Preconditie	Er is een operator aanwezig in het systeem.				
Postconditie	De response bevat de gegevens van de opgevraagde operator.				
Paden	1-3-4				
Verwijzing testboom	Operator-Opvragen				
Testactie(s)					
	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	getOperat	De gegevens van de opgevraagde		

	<p>or wordt aangeroept en met als operatorEncId "09889- abd7- 7882- abd7- 7882- abd7-7882 " en userId 1000.</p>	<p>operator worden teruggegeven. operatorDao2.setExternalId(2);</p> <p>operatorDao2.setFirstname("Operator 2");</p> <p>operatorDao2.setLastname("Lastname 2");</p> <p>operatorDao2.setEncId("09889-abd7-7882-abd7-7882-abd7-7882");</p> <p>operatorDao2.setEmail("operator2.nl");</p> <p>operatorDao2.setUserId(1200);</p> <p>operatorDao2.setActive(0);</p>		
--	---	--	--	--

Id	AT.T03-2				
Doel van de test	De validatie van het operatorEncId geeft een foutmelding, omdat deze niet valide is.				
Preconditie	Er is een operator aanwezig in het systeem.				
Postconditie	De response bevat een foutmelding				
Paden	2-5				
Verwijzing testboom	Operator-Opvragen				
Testactie(s)	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	getOperator wordt aangeroepen met als operatorEncId "" en userId 1000.	De response bevat een message met de tekst: "OperatorEncId is not valid!"		

Incident-Customer-Satisfaction-Timeframe

Id	AT.T04-1				
Doel van de test	Succesvol verkrijgen van Customer Satisfaction per TimeFrame				
Preconditie	Er zijn customer satisfaction aanwezig in het systeem.				
Postconditie	De opgevraagde customer satisfactions worden teruggegeven.				
Paden	1-4-5-6				
Verwijzing testboom	Incident-Customer-Satisfaction-Timeframe				
Testactie(s)	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	getCustomerSatisfactionByTimeFrame wordt aangeroepen met het TimeFrame DAY en als userId 1000.	De response bevat de customer satisfaction voor de dag en de dag ervoor.		

Id	AT.T04-2				
Doel van de test	De waarde van het opgegeven TimeFrame wordt omgezet naar DAY en hiervan wordt de Customer Satisfaction opgehaald.				
Preconditie	Er zijn customer satisfaction aanwezig in het systeem.				
Postconditie	De customer satisfactions voor de dag worden teruggegeven.				
Paden	2-3-4-5-6				
Verwijzing testboom	Incident-Customer-Satisfaction-Timeframe				
Testactie(

s)	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	getCustomerSatisfactionByTimeFrame wordt aangeroepen met het TimeFrame dag en als userId 1000.	Het TimeFrame wordt omgezet naar DAY. De response bevat de customer satisfaction voor de dag en de dag ervoor.		

Incident-Customer-Satisfaction-Relative

Id	AT.T05-1				
Doel van de test	Succesvol ophalen van Customer Satisfaction. Per opgegeven kwartaal en hetzelfde kwartaal van het voorgaande jaar.				
Preconditie	Er zijn customer satisfaction aanwezig voor het opgevraagde TimeFrame.				
Postconditie	De customer satisfaction voor de huidige maand en dezelfde maand het jaar ervoor worden teruggegeven.				
Paden	1-3-6-8-9-11-14-17				
Verwijzing testboom	Incident-Customer-Satisfaction-Relative				
Testactie(s)	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	getCustomerSatisfactionRelativeTo wordt aangeroepen met Q1 als TimeFrame en userId 1000.	De response bevat de customer satisfaction van Q1 van het huidige		

			jaar en het jaar ervoor.		
--	--	--	--------------------------------	--	--

Id	AT.T05-2				
Doel van de test	De waarde van het opgegeven TimeFrame wordt omgezet naar MONTH. Vervolgens wordt succesvol de Customer Satisfaction opgehaald. Per huidige maand en dezelfde maand van het voorgaande jaar.				
Preconditie	Er zijn customer satisfaction aanwezig voor het opgevraagde TimeFrame.				
Postconditie	De customer satisfaction voor de huidige maand en dezelfde maand het jaar ervoor worden teruggegeven.				
Paden	2-4-3-5-7-8-9-10-13 -16				
Verwijzing testboom	Incident-Customer-Satisfaction-Relative				
Testactie(s)	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	getCustomerSatisfactionRelativeTo wordt aangeroepen met Q1-Q2 als TimeFrame en userId 1000.	Het TimeFrame wordt omgezet naar MONTH. De response bevat de customer satisfaction van de huidige maand van het huidige jaar en het jaar ervoor.		

Id	AT.T05-3				
----	----------	--	--	--	--

Doel van de test	Succesvol ophalen van Customer Satisfaction. Per huidige maand en dezelfde maand van het voorgaande jaar.				
Preconditie	Er zijn customer satisfaction aanwezig voor het opgevraagde TimeFrame.				
Postconditie	De customer satisfaction voor de huidige maand en dezelfde maand het jaar ervoor worden teruggegeven.				
Paden	1-3-5-7-8-9-10-13-16				
Verwijzing testboom	Incident-Customer-Satisfaction-Relative				
Testactie(s)	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	getCustomerSatisfactionRelativeTo wordt aangeroepen met DAY als TimeFrame en userId 1000.	Het TimeFrame wordt omgezet naar MONTH. De response bevat de customer satisfaction van de huidige maand van het huidige jaar en het jaar ervoor.		

Id	AT.T05-4
Doel van de test	Succesvol ophalen van Customer Satisfaction. Per huidig jaar en het voorgaande jaar.
Preconditie	Er zijn customer satisfaction aanwezig voor het opgevraagde TimeFrame.
Postconditie	De customer satisfaction voor het huidige jaar en het jaar ervoor worden teruggegeven.

Paden	1-3-6-8-9-12-15-18				
Verwijzing testboom	Incident-Customer-Satisfaction-Relative				
Testactie(s)	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	getCustomerSatisfactionRelativeTo wordt aangeroepen met YEAR als TimeFrame en userId 1000.	De response bevat de customer satisfaction van het huidige jaar en het jaar ervoor.		

Incident-Operator-Scores-Timeframe

Id	AT.T06-1				
Doel van de test	Succesvol ophalen van Operator Scores voor het opgegeven TimeFrame.				
Preconditie	Er zijn operator scores aanwezig in het systeem.				
Postconditie	De operator scores van het opgevraagde TimeFrame worden teruggegeven.				
Paden	1-4-5-6-7				
Verwijzing testboom	Incident-Operator-Scores-Timeframe				
Testactie(s)	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	getOperatorScoresByTimeFrame wordt uitgevoerd met DAY als TimeFrame en 1000 als userId.	De operator scores voor de huidige dag en de dag ervoor worden teruggegeven		

			n.		
--	--	--	----	--	--

Id	AT.T06-2				
Doel van de test	Omzetten van een niet valide TimeFrame en daarna succesvol ophalen van Operator Scores voor het TimeFrame DAY.				
Preconditie	Er zijn operator scores aanwezig in het systeem.				
Postconditie	De operator scores van TimeFrame DAY worden teruggegeven.				
Paden	2-3-4-5-6-7				
Verwijzing testboom	Incident-Operator-Scores-Timeframe				
Testactie(s)	Testacti e Id	Testactie	Verwachte reactie	Conclusi e	Opmerkin g
	1	getOperatorScoresByTimeFrame wordt uitgevoerd met maand als TimeFrame en 1000 als userId.	Het TimeFrame wordt omgezet naar DAY. De operator scores voor de huidige dag en de dag ervoor worden teruggegeven.		

Incident-Operator-Scores-Relative

Id	AT.T07-1
Doel van de test	Succesvol ophalen van Operator Scores. Per huidig kwartaal en hetzelfde kwartaal van het voorgaande jaar.
Preconditie	Er zijn operator scores aanwezig in het systeem.

Postconditie	De operator scores voor de opgevraagde periode en dezelfde periode het jaar ervoor worden teruggegeven.				
Paden	1-4-6-8-9-11-14-16				
Verwijzing testboom	Incident-Operator-Scores-Relative				
Testactie(s)					
	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call "/incident/stats/operator-scores/relative-to " wordt uitgevoerd met als TimeFrame Q2 en als userId 1000.	De operator scores voor Q2 van het huidige jaar en het jaar ervoor worden opgehaald.		

Id	AT.T07-2				
Doel van de test	De waarde van het opgegeven TimeFrame wordt omgezet naar MONTH. Vervolgens wordt succesvol de Operator Scores opgehaald. Per huidige maand en dezelfde maand van het voorgaande jaar.				
Preconditie	Er zijn operator scores aanwezig in het systeem.				
Postconditie	De operator scores voor de maand en dezelfde maand het jaar ervoor worden teruggegeven.				
Paden	2-3-4-5-7-8-9-10-13-17				
Verwijzing testboom	Incident-Operator-Scores-Relative				
Testactie(s)					
	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call "/incident/stats/operator-scores/relative-to " wordt uitgevoerd met als TimeFrame Q5 en als userId 1000.	Het TimeFrame wordt omgezet naar MONTH. De operator scores voor		

			de huidige maand van het huidige jaar en het jaar ervoor worden opgehaald.		
--	--	--	--	--	--

Id	AT.T07-3				
Doel van de test	De waarde van het opgegeven TimeFrame wordt omgezet naar MONTH. Vervolgens wordt succesvol de Operator Scores opgehaald. Per huidige maand en dezelfde maand van het voorgaande jaar.				
Preconditie	Er zijn operator scores aanwezig in het systeem.				
Postconditie	De operator scores voor de maand en dezelfde maand het jaar ervoor worden teruggegeven.				
Paden	1-4-5-7-8-9-10-13-17				
Verwijzing testboom	Incident-Operator-Scores-Relative				
Testactie(s)					
	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call "/incident/stats/operator-scores/relative-to " wordt uitgevoerd met als TimeFrame DAY en als userId 1000.	Het TimeFrame wordt omgezet naar MONTH. De operator scores voor de huidige maand van het huidige jaar en het jaar ervoor worden opgehaald.		

Id	AT.T07-4				
Doel van de	Succesvol ophalen van Operator Scores. Per huidig jaar en het voorgaande jaar.				

test					
Preconditie	Er zijn operator scores aanwezig in het systeem.				
Postconditie	De operator scores voor de opgevraagde periode en dezelfde periode het jaar ervoor worden teruggegeven.				
Paden	1-4-6-8-9-12-15-18				
Verwijzing testboom	Incident-Operator-Scores-Relative				
Testactie(s)	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call “/incident/stats/operator-scores/relative-to” wordt uitgevoerd met als TimeFrame YEAR en als userId 1000.	De operator scores van het huidige jaar en het jaar ervoor worden opgehaald.		

Algoritme Test : Branche Coverage

Incident-Opvragen

Id	AT.T08-1				
Doel van de test	Succesvol ophalen van Feedback voor het opgegeven valide feedbackEncId				
Preconditie	Er is een feedback aanwezig voor het valide feedbackEncId.				
Postconditie	De gegevens van de feedback worden teruggegeven.				
Paden	1-3-5				
Verwijzing testboom	Incident-Opvragen				
Testactie(s)	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call “incident/{feedbackEncId}”	De gegevens van de opgevraagde		

		“ wordt aangeroepen met als feedbackEncId “XXXXXX” en als userId 1000.	feedback worden teruggegeven		
--	--	--	------------------------------	--	--

Id	AT.T08-2				
Doel van de test	De validatie van het FeedbackEncId geeft een error, omdat het opgegeven feedbackEncId niet valide is.				
Preconditie	Er is een feedback aanwezig voor het valide feedbackEncId.				
Postconditie	Er wordt een foutmelding teruggegeven.				
Paden	2-4				
Verwijzing testboom	Incident-Opvragen				
Testactie(s)	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call “incident/{feedbackEncId}” wordt aangeroepen met als feedbackEncId “<CHAR35” en als userId 1000.	De response bevat een message met de tekst: "No valid encId supplied!"		

Incident-Versturen

Id	AT.T09-1				
Doel van de test	het succesvol verzenden van een feedback mail voor het opgegeven feedbackEncId.				
Preconditie	Er is een feedback aanwezig voor het valide feedbackEncId.				
Postconditie	De feedback mail is verzonden aan de beoordelaar en een succes message wordt teruggegeven.				
Paden	1-3-6				

Verwijzing testboom	Incident-Versturen				
Testactie(s)					
	Testacti e Id	Testactie	Verwachte reactie	Conclusi e	Opmerkin g
	1	De call “incident/{feedbackEncId}/send” wordt aangeroepen met als feedbackEncId “XXXXXX” en als userId 1000.	De response bevat een message met de tekst: "Feedback mail send for encId: " + XXXXXX + "." En de feedback mail is verzonden aan de beoordelaar.		

Id	AT.T09-2				
Doel van de test	De validatie van het FeedbackEncId geeft een error, omdat het opgegeven feedbackEncId niet valide is.				
Preconditie	Er is een feedback aanwezig voor het valide feedbackEncId.				
Postconditie	De feedback mail is niet verzonden aan de beoordelaar en een fout message wordt teruggegeven.				
Paden	2-4				
Verwijzing testboom	Incident-Versturen				
Testactie(s)					
	Testacti e Id	Testactie	Verwachte reactie	Conclusi e	Opmerkin g
	1	De call “incident/{feedbackEncId}/send	De response		

		nd “ wordt aangeroepen met als feedbackEncId “<35CHAR” en als userId 1000.	bevat een message met de tekst: "No valid encId supplied!" En de feedback mail is niet verzonden aan de beoordelaar.		
--	--	---	---	--	--

Id	AT.T09-3				
Doel van de test	SendFeedbackMail geeft FALSE, omdat de feedback mail niet verzonden kon worden.				
Preconditie	Er is een feedback aanwezig voor het valide feedbackEncId.				
Postconditie	De feedback mail is niet verzonden aan de beoordelaar en een fout message wordt teruggegeven.				
Paden	1-5-7				
Verwijzing testboom	Incident-Versturen				
Testactie(s)					
	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call “incident/{feedbackEncId}/send “ wordt aangeroepen met als feedbackEncId “XXXXXX” en als userId 1000.	De response bevat een message met de tekst: "Feedback mail not send for encId: " + feedbackEncId + "!" En de feedback mail is verzonden aan de		

			beoordelaar.		
--	--	--	--------------	--	--

Gebruiker-Toevoegen

Id	AT.T10-1				
Doel van de test	Het succesvol toevoegen van een User.				
Preconditie	De gebruiker is nog niet aanwezig in het systeem.				
Postconditie	De gebruiker is toegevoegd aan het systeem.				
Paden	1-3-5-7-10				
Verwijzing testboom	Gebruiker-Toevoegen				
Testactie(s)	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call POST “/accesstoken” wordt uitgevoerd met de body: “XXXXXX”	De gebruiker is toegevoegd. De response bevat de gegevens van de nieuwe gebruiker.		

Id	AT.T10-2				
Doel van de test	De validatie van de User geeft een foutmelding, omdat de User niet valide is.				
Preconditie	De gebruiker is nog niet aanwezig in het systeem.				
Postconditie	De gebruiker is niet toegevoegd aan het systeem.				
Paden	2-4				

Verwijzing testboom	Gebruiker-Toevoegen				
Testactie(s)					
	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call POST “/accesstoken” wordt uitgevoerd met de body: “XXXXXX”	De gebruiker is niet toegevoegd. De response bevat een message met de tekst: "User not valid!"		

Id	AT.T10-3				
Doel van de test	Er wordt een foutmelding gegeven, omdat de gebruiker reeds bestaat.				
Preconditie	De gebruiker is reeds aanwezig in het systeem.				
Postconditie	De gebruiker is niet toegevoegd aan het systeem. De response geeft aan dat de gebruiker al bestaat.				
Paden	1-3-6-8				
Verwijzing testboom	Gebruiker-Toevoegen				
Testactie(s)					
	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call POST “/accesstoken” wordt uitgevoerd met de body: “XXXXXX”	De gebruiker is niet toegevoegd. De response bevat een message met de tekst: "User already exists!"		

Id	AT.T10-4				
Doel van de test	CreateUser geeft een exception, omdat het toevoegen niet lukt.				

Preconditie	De gebruiker is nog niet aanwezig in het systeem.				
Postconditie	De gebruiker is niet toegevoegd aan het systeem. Hiervan wordt melding gegeven in de response.				
Paden	1-3-5-9-11				
Verwijzing testboom	Gebruiker-Toevoegen				
Testactie(s)					
	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call POST “/accesstoken” wordt uitgevoerd met de body: “XXXXXX”	De gebruiker is niet toegevoegd. De response bevat een message met de tekst: "Could not create new user with accountname: " + user.getAccountName() + "!"		

Operator-Updaten

Id	AT.T11-1				
Doel van de test	Succesvol updaten van een Operator.				
Preconditie	De operator is reeds aanwezig in het systeem.				
Postconditie	De operator is aangepast en is niet meer gelijk aan de operator voor de test.				
Paden	1-2-4-5-9-10-11-12				
Verwijzing testboom	Operator-Updaten				
Testactie(s)					
	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call PUT	De operator is		

		“/operator” wordt uitgevoerd met als body: “dao1 + “nieuwe voornaam 2” en als userId 1000.	aangepast. De response bevat de gegevens van de operator.		
--	--	--	---	--	--

Id	AT.T11-2				
Doel van de test	De validatie van de Operator geeft een foutmelding, omdat de Operator niet valide is.				
Preconditie	De operator is reeds aanwezig in het systeem.				
Postconditie	De operator is niet aangepast. Hiervan wordt in de response een melding gemaakt.				
Paden	1-2-3-7				
Verwijzing testboom	Operator-Update				
Testactie(s)	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call PUT “/operator” wordt uitgevoerd met als body: “dao1 + lege achternaam” en als userId 1000.	De operator is niet aangepast. De response bevat een message met de tekst: "Operator not valid!"		

Id	AT.T11-3				
Doel van de test	Er wordt een foutmelding gegeven omdat de Operator niet bestaat.				
Preconditie	De operator is niet aanwezig in het systeem.				
Postconditie	De operator is niet aangepast. Hiervan wordt een melding gemaakt in de response.				
Paden	1-2-4-5-6-8				
Verwijzing	Operator-Update				

testboom					
Testactie(s)					
	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call PUT “/operator” wordt uitgevoerd met als body: “dao1 + 09889-abd7-7882-abd7-3209-09df-3209” en als userId 1000.	De operator is niet aangepast. De response bevat een message met de tekst: "Operator not found!"		

Operator-Toevoegen

Id	AT.T12-1				
Doel van de test	Het toevoegen van een bestaande Operator welke niet actief is, wordt weer actief gemaakt.				
Preconditie	De operator is aanwezig, maar heeft de status active=False.				
Postconditie	De operator heeft de status active=True				
Paden	1-2-4-6-7-9-13-16-17				
Verwijzing testboom	Operator-Toevoegen				
Testactie(s)					
	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call POST “/operator” wordt uitgevoerd met de body: DAO1 en active = false ” en als userId 1000.	De status van de operator is active=true. De response bevat een message met de tekst: "Operator already created!" The operator status is set to active, it was inactive."		

Id	AT.T12-2
----	----------

Doel van de test	De validatie van de Operator geeft een foutmelding, omdat de Operator niet valide is.				
Preconditie	De operator is niet aanwezig.				
Postconditie	De operator is niet toegevoegd. In de response wordt hier melding van gegeven.				
Paden	1-2-3-5				
Verwijzing testboom	Operator-Toevoegen				
Testactie (s)					
	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call POST “/operator” wordt uitgevoerd met de body: <pre> operatorDao1.setExternalId(1); operatorDao1.setLastname("Lastname"); operatorDao1.setEncId("345-0-09df-3209-09df-3209-09df-3209"); operatorDao1.setEmail("operator@tor.nl"); operatorDao1.setUserId(1100); operatorDao1.setActive(1); </pre> ” en als userId 1000.	De operator is niet toegevoegd. De response bevat een message met de tekst: "Could not add new operator!"		

Id	AT.T12-3
Doel van de test	Het toevoegen van een bestaande Operator welke ook actief is.
Preconditie	De operator is aanwezig en heeft de status active=True.
Postconditie	Er wordt een melding teruggegeven dat de operator al bestaat.
Paden	1-2-4-6-7-10-17

Verwijzing testboom	Operator-Toevoegen				
Testactie(s)					
	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call POST “/operator” wordt uitgevoerd met de body: “XXXXXX” en als userId 1000.	De response bevat een message met de tekst: "Operator already created!"		

Id	AT.T12-4				
Doel van de test	Het succesvol toevoegen van een nieuwe Operator.				
Preconditie	De operator is niet aanwezig.				
Postconditie	De operator is toegevoegd.				
Paden	1-2-4-6-8-12-15				
Verwijzing testboom	Operator-Toevoegen				
Testactie(s)					
	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call POST “/operator” wordt uitgevoerd met de body: “XXXXXX” en als userId 1000.	De nieuwe operator is toegevoegd aan het systeem. De gegevens van de operator worden teruggegeven in de response.		

Id	AT.T12-5				
Doel van de test	CreateOperator geeft een exception, omdat het toevoegen niet is gelukt.				
Preconditie	De operator is niet aanwezig.				
Postconditi	De operator is niet toegevoegd. Er wordt hiervan een melding gegeven.				

e					
Paden	1-2-4-6-8-11-14				
Verwijzing testboom	Operator-Toevoegen				
Testactie(s)					
	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call POST "/operator" wordt uitgevoerd met de body: "XXXXXX" en als userId 1000.	De operator is niet toegevoegd. De response bevat een message met de tekst: "Could not create operator for userId: " + operator.getUserId().toString() + "!"		

Incident-Toevoegen = Incident-Update

Id	AT.T13-1				
Doel van de test	Het succesvol toevoegen van een nieuw incident.				
Preconditie	De feedback is niet aanwezig in het systeem.				
Postconditie	De feedback is toegevoegd aan het systeem. Het feedbackEncId wordt terug gegeven.				
Paden	1-2-4-7-9-10				
Verwijzing testboom	Incident-Toevoegen = Incident-Update				
Testactie(s)					
	Testactie	Testactie	Verwachte reactie	Conclusie	Opmerking

	Id				
	1	De call POST “/incident” wordt aangeroepen met de body: “XXXXXX” en userId 1000.	De feedback is toegevoegd aan het systeem. Het feedbackEncId wordt teruggegeven in de response.		

Id	AT.T13-2				
Doel van de test	De validatie van het Incident geeft een foutmelding, omdat het incident niet valide is.				
Preconditie	De feedback is niet aanwezig in het systeem.				
Postconditie	De feedback is niet toegevoegd. Hiervan wordt een melding gegeven.				
Paden	1-2-3-5				
Verwijzing testboom	Incident-Toevoegen = Incident-Update				
Testactie(s)	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call POST “/incident” wordt aangeroepen met de body: “XXXXXX” en userId 1000.	De feedback is niet toegevoegd aan het systeem. De response bevat een message met de tekst: "Request is not valid!"		

Id	AT.T13-3				
Doel van de test	ProcesIncident geeft een exception, omdat het verwerken van het incident niet is gelukt.				
Preconditie	De feedback is niet aanwezig in het systeem.				
Postconditie	De feedback is niet toegevoegd aan het systeem. Hiervan wordt een melding gegeven.				
Paden	1-2-4-6-8				

Verwijzing testboom	Incident-Toevoegen = Incident-Update				
Testactie(s)					
	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call POST “/incident” wordt aangeroepen met de body: “XXXXXX” en userId 1000.	De feedback is niet toegevoegd aan het systeem. De response bevat een message met de tekst: "Could not process incident!"		

Beslissingstabellen Test

Incident-Toevoegen = Incident-Update

Id	CE.T01-1				
Doel van de test	Het request is niet valide, omdat het incident niet valide is.				
Preconditie	Er is geen feedback aanwezig in het systeem.				
Postconditie	er is geen feedback toegevoegd, omdat het incident niet valide is.				
Verwijzing testboom	Incident-Toevoegen = Incident-Update				
Testactie(s)					
	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call POST “/incident” wordt aangeroepen met de body: “XXXXXX” en userId 1000.	De body is niet valide, er wordt een message teruggegeven met de melding: "Request is not valid!"		

Id	CE.T01-2				
Doel van de	Het incident wordt verwerkt tot een feedback in de database.				

test					
Preconditie	Er is geen feedback aanwezig in het systeem.				
Postconditie	Er is een feedback toegevoegd. Het feedbackEncId wordt teruggegeven in de response.				
Verwijzing testboom	Incident-Toevoegen = Incident-Update				
Testactie(s)					
	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call POST "/incident" wordt aangeroepen met de body: "XXXXXX" en userId 1000.	De reponse bevat een message met het feedbackEncId.		

Id	CE.T01-4				
Doel van de test	Er treed een exception op doordat het incident niet verwerkt kan worden.				
Preconditie	Er is geen feedback aanwezig in het systeem.				
Postconditie	er is geen feedback toegevoegd, omdat het incident niet verwerkt kon worden.				
Verwijzing testboom	Incident-Toevoegen = Incident-Update				
Testactie(s)					
	Testactie Id	Testactie	Verwachte reactie	Conclusie	Opmerking
	1	De call POST "/incident" wordt aangeroepen met de body: "XXXXXX" en userId 1000.	Tijdens het verwerken van het incident gaat iets mis. Er wordt een exception gegeven. De response heeft een message met de melding: "Could not process incident!"		

Bijlage X: Rest-API documentatie

POST {{url}}accesstoken/

body:

```
{
  "accountName": "Test Bedrijf A.P.I"
}
```

response:

```
{
  "consumerKey": "VBXpDZPSakFQswTWG8xdwtjWwCdZ9ZUr",
  "consumerSecret": "jvT25xxfGWlr4by3NRmb75kWvypaZtPw",
  "accountName": "Test Bedrijf A.P.I",
  "role": "DASHBOARD"
}
```

GET {{url}}accesstoken/

header x-user-key

response:

```
{
  "consumerKey": "thHNsVStEyQDqj5Zh6feRYutfhwgNcgZ",
  "consumerSecret": "5x2debhYGV6WMX6bDuRz6sRmWtbyXYLW",
  "accountName": "Mailplus Dashboard",
  "role": "SUPERUSER"
}
```

POST {{url}}accesstoken/regenerate-keys

body:

```
{
  "consumerKey": "6c92b35a7f13c383ef7c32a147439927"
}
```

response:

```
{
  "consumerKey": "ckfNSymwisfHUSwV6mkfk4ipXmCh2DLZ",
  "consumerSecret": "cp9tkqcLjDhngisrI8htpt7hKUKHryPD",
  "accountName": "Test Bedrijf A.P.I",
  "role": "DASHBOARD"
}
```

POST {{url}}operator

body:

```
{
  "externalId": 3,
```

```
    "firstname": "Jos",
    "lastname": "Verhaar",
    "email": "josverhaar@gmail.com"
}
```

response:

```
{
  "externalId": "5",
  "encId": "263c7508-35e5-4a38-b72a-b3fc6936158b",
  "firstname": "Jos",
  "lastname": "Verhaar",
  "email": "josverhaar@gmail.com",
  "active": true
}
```

PUT {{url}}operator

body:

```
{
  "externalId": 3,
  "firstname": "Jos",
  "lastname": "Verhaar",
  "email": "jos@mailplus.nl"
}
```

response:

```
{
  "externalId": "5",
  "encId": "263c7508-35e5-4a38-b72a-b3fc6936158b",
  "firstname": "Jos",
  "lastname": "Verhaar",
  "email": "jos@mailplus.nl",
  "active": true
}
```

GET {{url}}operator/{{operatorId}}

response:

```
{
  "externalId": "2",
  "encId": "61987163-ca2f-4897-8cb3-e786701b41f",
  "firstname": "Jos",
  "lastname": "Verhaar",
  "email": "jos@mailplus.nl",
  "active": true
}
```

DELETE {{url}}operator/{{operatorId}}

response:

```
{
  "externalId": "2",
  "encId": "61987163-ca2f-4897-8cb3-e786701b41f",
  "firstname": "Jos",
  "lastname": "Verhaar",
  "email": "jos@mailplus.nl",
  "active": false
}
```

GET {{url}}operator/list

response:

```
[
  {
    "externalId": "2",
    "encId": "61987163-ca2f-4897-8cb3-e786701b41f",
    "firstname": "Jos",
    "lastname": "Verhaar",
    "email": "jos@mailplus.nl",
    "active": false
  },
  {
    "externalId": "5",
    "encId": "263c7508-35e5-4a38-b72a-b3fc6936158b",
    "firstname": "Jos",
    "lastname": "Verhaar",
    "email": "jos@mailplus.nl",
    "active": true
  },
  {
    "externalId": "1",
    "encId": "e54beacc-09be-4bc8-bd41-a7b255bbc22",
    "firstname": "Paul",
    "lastname": "Nederlof",
    "email": "paulne@mailplus.nl",
    "active": true
  },
  {
    "externalId": "3",
    "encId": "78222c7a-0c7d-47f8-8f1c-d3e3cfb8dd1",
    "firstname": "Ralph",
    "lastname": "van Delft",
    "email": "ralph@mailplus.nl",
    "active": true
  }
]
```

GET {{url}}operator/stats/operator-score/{{operatorId}}?timeframe=day

GET {{url}}operator/stats/operator-score/{{operatorId}}?timeframe=month

GET {{url}}operator/stats/operator-score/{{operatorId}}?timeframe=q1

```
GET {{url}}operator/stats/operator-score/{{operatorId}}?timeframe=q2
GET {{url}}operator/stats/operator-score/{{operatorId}}?timeframe=q3
GET {{url}}operator/stats/operator-score/{{operatorId}}?timeframe=q4
GET {{url}}operator/stats/operator-score/{{operatorId}}?timeframe=year
```

response:

```
{
  "operatorScore": [
    {
      "good": {
        "count": 1,
        "percentage": 100
      },
      "ok": {
        "count": 0,
        "percentage": 0
      },
      "bad": {
        "count": 0,
        "percentage": 0
      },
      "operatorFirstname": "Jos"
    }
  ]
}
```

POST {{url}}incident

body:

```
{
  "id": "I2016 007",
  "shortDescription": "Feedback verzoek via Rest API call.",
  "requestor": {
    "firstname": "Rest API",
    "lastname": "feedback/create",
    "account": "Postman",
    "email": "josverhaar@gmail.com"
  },
  "operator": {
    "firstname": "Jos",
    "lastname": "Verhaar",
    "externalId": 2
  }
}
```

Response:

```
{
```

```
"encId": "30408d8c-d48b-40d7-be5c-6a3af9b3736b"
}
```

PUT {{url}}incident

body:

```
{
  "id": "I2016 1107",
  "shortDescription": "Feedback verzoek via Rest API call.",
  "requestor": {
    "firstname": "Rest API",
    "lastname": "feedback/create",
    "account": "Postman",
    "email": "josverhaar@gmail.com"
  },
  "operator": {
    "firstname": "Jos",
    "lastname": "Verhaar",
    "externalId": 2
  }
}
```

response:

```
{
  "encId": "30408d8c-d48b-40d7-be5c-6a3af9b3736b"
}
```

GET {{url}}incident/{{feedbackEncId}}

response:

```
{
  "encId": "qwGF2nNWLS8qku6KZCJDJb59KLngAjM",
  "ticketNumber": "I2016 011",
  "shortDescription": "Short description 11",
  "operator": {
    "lastName": "Nederlof",
    "encId": "e54beacc-09be-4bc8-bd41-a7b255bbc22",
    "firstName": "Paul"
  },
  "requestor": {
    "lastName": "API 11",
    "email": "paulne+11@mailplus.nl",
    "account": "Postman 11",
    "firstName": "Rest"
  },
  "feedbackLinks": {
    "OK": {
```

```

    "url":
"http://localhost:8080/supportfeedback/feedback/qwGF2nNWLS8qku6KZCJDJjb59KLNgAj
M/2"
  },
  "GOOD": {
    "url":
"http://localhost:8080/supportfeedback/feedback/qwGF2nNWLS8qku6KZCJDJjb59KLNgAj
M/1"
  },
  "BAD": {
    "url":
"http://localhost:8080/supportfeedback/feedback/qwGF2nNWLS8qku6KZCJDJjb59KLNgAj
M/3"
  }
}
}
}

```

POST {{url}}incident/{{feedbackEncId}}/send

```

response:
{
  "message": "Feedback mail send for encId: qwGF2nNWLS8qku6KZCJDJjb59KLNgAjM."
}

```

```

GET {{url}}incident/stats/customer-satisfaction?timeframe=day
GET {{url}}incident/stats/customer-satisfaction?timeframe=month
GET {{url}}incident/stats/customer-satisfaction?timeframe=q1
GET {{url}}incident/stats/customer-satisfaction?timeframe=q2
GET {{url}}incident/stats/customer-satisfaction?timeframe=q3
GET {{url}}incident/stats/customer-satisfaction?timeframe=q4
GET {{url}}incident/stats/customer-satisfaction?timeframe=year

```

```

response:
{
  "current": {
    "customerSatisfaction": {
      "requestCount": 0,
      "responseCount": 0
    }
  },
  "previous": {
    "customerSatisfaction": {
      "good": {
        "count": 1,
        "percentage": 25
      },
      "ok": {
        "count": 2,
        "percentage": 50
      }
    }
  }
}

```

```

    },
    "bad": {
      "count": 1,
      "percentage": 25
    },
    "requestCount": 4,
    "responseCount": 4
  }
}
}

```

GET {{url}}incident/stats/customer-satisfaction/count?count=100

response:

```

{
  "current": {
    "customerSatisfaction": {
      "good": {
        "count": 2,
        "percentage": 66.667
      },
      "ok": {
        "count": 1,
        "percentage": 33.333
      },
      "bad": {
        "count": 0,
        "percentage": 0
      }
    }
  },
  "previous": {
    "customerSatisfaction": {
      "good": {
        "count": 2,
        "percentage": 66.667
      },
      "ok": {
        "count": 1,
        "percentage": 33.333
      },
      "bad": {
        "count": 0,
        "percentage": 0
      }
    }
  }
}

```

```

GET {{url}}incident/stats/operator-scores?timeframe=day
GET {{url}}incident/stats/operator-scores?timeframe=month
GET {{url}}incident/stats/operator-scores?timeframe=q1
GET {{url}}incident/stats/operator-scores?timeframe=q2
GET {{url}}incident/stats/operator-scores?timeframe=q3
GET {{url}}incident/stats/operator-scores?timeframe=q4
GET {{url}}incident/stats/operator-scores?timeframe=year

```

response:

```

{
  "current": {},
  "previous": {
    "operatorScores": [
      {
        "good": {
          "count": 1,
          "percentage": 50
        },
        "ok": {
          "count": 1,
          "percentage": 50
        },
        "bad": {
          "count": 0,
          "percentage": 0
        },
        "operatorFirstname": "Paul"
      },
      {
        "good": {
          "count": 0,
          "percentage": 0
        },
        "ok": {
          "count": 1,
          "percentage": 100
        },
        "bad": {
          "count": 0,
          "percentage": 0
        },
        "operatorFirstname": "Ralph"
      }
    ]
  }
}

```

```

GET {{url}}incident/stats/live-scores?timeframe=day
GET {{url}}incident/stats/live-scores?timeframe=month

```



```

GET {{url}}incident/stats/live-scores?timeframe=q1
GET {{url}}incident/stats/live-scores?timeframe=q2
GET {{url}}incident/stats/live-scores?timeframe=q3
GET {{url}}incident/stats/live-scores?timeframe=q4
GET {{url}}incident/stats/live-scores?timeframe=year

```

```

{
  "current": {},
  "previous": {
    "liveScores": [
      {
        "rating": "OK",
        "ticketNumber": "I2016 010",
        "requestor": {
          "firstname": "Rest",
          "lastname": "API 10",
          "account": "Postman 10"
        }
      },
      {
        "rating": "GOOD",
        "ticketNumber": "I2016 009",
        "requestor": {
          "firstname": "Rest",
          "lastname": "API 9",
          "account": "Postman 9"
        }
      },
      {
        "rating": "GOOD",
        "ticketNumber": "I2016 017",
        "requestor": {
          "firstname": "Rest",
          "lastname": "API 17",
          "account": "Postman 17"
        }
      },
      {
        "rating": "OK",
        "ticketNumber": "I2016 016",
        "requestor": {
          "firstname": "Rest",
          "lastname": "API 16",
          "account": "Postman 16"
        }
      }
    ]
  }
}

```

```

GET {{url}}incident/stats/remarks?timeframe=day
GET {{url}}incident/stats/remarks?timeframe=month
GET {{url}}incident/stats/remarks?timeframe=q1
GET {{url}}incident/stats/remarks?timeframe=q2
GET {{url}}incident/stats/remarks?timeframe=q3
GET {{url}}incident/stats/remarks?timeframe=q4
GET {{url}}incident/stats/remarks?timeframe=year

```

```

private String remark;
    private String requestorFirstname;
    private String requestorLastname;
    private String accountName;
    private String operatorFirstname;
    private String rating;

```

```

GET {{url}}incident/stats/customer-satisfaction/relative-to?timeframe=month
{
  "current": {
    "January": {
      "requestCount": 0,
      "responseCount": 0
    }
  },
  "previous": {
    "January": {
      "requestCount": 0,
      "responseCount": 0
    }
  }
}

```

```

GET {{url}}incident/stats/customer-satisfaction/relative-to?timeframe=q1
{
  "Q1 2016": {
    "January": {
      "requestCount": 0,
      "responseCount": 0
    },
    "February": {
      "requestCount": 0,
      "responseCount": 0
    },
    "March": {
      "requestCount": 0,
      "responseCount": 0
    }
  },
  "Q1 2017": {
    "January": {

```

```

    "requestCount": 0,
    "responseCount": 0
  },
  "February": {
    "requestCount": 0,
    "responseCount": 0
  },
  "March": {
    "requestCount": 0,
    "responseCount": 0
  }
}
}
}

```

```

GET {{url}}incident/stats/customer-satisfaction/relative-to?timeframe=q2
GET {{url}}incident/stats/customer-satisfaction/relative-to?timeframe=q3
GET {{url}}incident/stats/customer-satisfaction/relative-to?timeframe=q4
GET {{url}}incident/stats/customer-satisfaction/relative-to?timeframe=year
response:

```

```

{
  "2016": {
    "January": {
      "requestCount": 0,
      "responseCount": 0
    },
    "February": {
      "requestCount": 0,
      "responseCount": 0
    },
    "March": {
      "requestCount": 0,
      "responseCount": 0
    },
    "April": {
      "requestCount": 0,
      "responseCount": 0
    },
    "May": {
      "requestCount": 0,
      "responseCount": 0
    },
    "June": {
      "requestCount": 0,
      "responseCount": 0
    },
    "July": {
      "requestCount": 0,
      "responseCount": 0
    },
    "August": {

```

```

    "requestCount": 0,
    "responseCount": 0
  },
  "September": {
    "good": {
      "count": 1,
      "percentage": 50
    },
    "ok": {
      "count": 1,
      "percentage": 50
    },
    "bad": {
      "count": 0,
      "percentage": 0
    },
    "requestCount": 2,
    "responseCount": 2
  },
  "October": {
    "good": {
      "count": 0,
      "percentage": 0
    },
    "ok": {
      "count": 0,
      "percentage": 0
    },
    "bad": {
      "count": 1,
      "percentage": 100
    },
    "requestCount": 1,
    "responseCount": 1
  },
  "November": {
    "good": {
      "count": 0,
      "percentage": 0
    },
    "ok": {
      "count": 1,
      "percentage": 100
    },
    "bad": {
      "count": 0,
      "percentage": 0
    },
    "requestCount": 1,
    "responseCount": 1
  }

```

```

    },
    "December": {
        "requestCount": 0,
        "responseCount": 0
    }
},
"2017": {
    "January": {
        "requestCount": 0,
        "responseCount": 0
    },
    "February": {
        "requestCount": 0,
        "responseCount": 0
    },
    "March": {
        "requestCount": 0,
        "responseCount": 0
    },
    "April": {
        "requestCount": 0,
        "responseCount": 0
    },
    "May": {
        "requestCount": 0,
        "responseCount": 0
    },
    "June": {
        "requestCount": 0,
        "responseCount": 0
    },
    "July": {
        "requestCount": 0,
        "responseCount": 0
    },
    "August": {
        "requestCount": 0,
        "responseCount": 0
    },
    "September": {
        "requestCount": 0,
        "responseCount": 0
    },
    "October": {
        "requestCount": 0,
        "responseCount": 0
    },
    "November": {
        "requestCount": 0,
        "responseCount": 0
    }
}

```

```

    },
    "December": {
      "requestCount": 0,
      "responseCount": 0
    }
  }
}

```

```

GET {{url}}incident/stats/customer-satisfaction/{{operatorId}}/relative-to?timeframe=month
GET {{url}}incident/stats/customer-satisfaction/{{operatorId}}/relative-to?timeframe=q1
GET {{url}}incident/stats/customer-satisfaction/{{operatorId}}/relative-to?timeframe=q2
GET {{url}}incident/stats/customer-satisfaction/{{operatorId}}/relative-to?timeframe=q3
GET {{url}}incident/stats/customer-satisfaction/{{operatorId}}/relative-to?timeframe=q4
GET {{url}}incident/stats/customer-satisfaction/{{operatorId}}/relative-to?timeframe=year

```

```

GET {{url}}incident/stats/operator-scores/relative-to?timeframe=month
GET {{url}}incident/stats/operator-scores/relative-to?timeframe=q1
GET {{url}}incident/stats/operator-scores/relative-to?timeframe=q2
GET {{url}}incident/stats/operator-scores/relative-to?timeframe=q3
GET {{url}}incident/stats/operator-scores/relative-to?timeframe=q4
GET {{url}}incident/stats/operator-scores/relative-to?timeframe=year

```

```

response:
{
  "2016": {
    "January": [],
    "February": [],
    "March": [],
    "April": [],
    "May": [],
    "June": [],
    "July": [],
    "August": [],
    "September": [
      {
        "good": {
          "count": 1,
          "percentage": 50
        },
        "ok": {
          "count": 1,
          "percentage": 50
        },
        "bad": {
          "count": 0,
          "percentage": 0
        }
      },
      "operatorFirstname": "Paul"
    ]
  }
}

```

```

],
"October": [],
"November": [
  {
    "good": {
      "count": 0,
      "percentage": 0
    },
    "ok": {
      "count": 1,
      "percentage": 100
    },
    "bad": {
      "count": 0,
      "percentage": 0
    },
    "operatorFirstname": "Ralph"
  }
],
"December": []
},
"2017": {
  "January": [],
  "February": [],
  "March": [],
  "April": [],
  "May": [],
  "June": [],
  "July": [],
  "August": [],
  "September": [],
  "October": [],
  "November": [],
  "December": []
}
}

```

```

GET {{url}}incident/stats/operator-scores/{{operatorId}}/relative-to?timeframe=month
GET {{url}}incident/stats/operator-scores/{{operatorId}}/relative-to?timeframe=q1
GET {{url}}incident/stats/operator-scores/{{operatorId}}/relative-to?timeframe=q2
GET {{url}}incident/stats/operator-scores/{{operatorId}}/relative-to?timeframe=q3
GET {{url}}incident/stats/operator-scores/{{operatorId}}/relative-to?timeframe=q4
GET {{url}}incident/stats/operator-scores/{{operatorId}}/relative-to?timeframe=year
response:

```

```

{
  "2016": {
    "January": [],
    "February": [],
    "March": [],

```

```

    "April": [],
    "May": [],
    "June": [],
    "July": [],
    "August": [],
    "September": [],
    "October": [
      {
        "good": {
          "count": 1,
          "percentage": 100
        },
        "ok": {
          "count": 0,
          "percentage": 0
        },
        "bad": {
          "count": 0,
          "percentage": 0
        }
      },
      "operatorFirstname": "Jos"
    ]
  ],
  "November": [],
  "December": []
},
"2017": {
  "January": [],
  "February": [],
  "March": [],
  "April": [],
  "May": [],
  "June": [],
  "July": [],
  "August": [],
  "September": [],
  "October": [],
  "November": [],
  "December": []
}
}

```


Bijlage XI: Afstudeerplan

Informatie afstudeerder en gastbedrijf *(structuur niet wijzigen)*

Afstudeerblok: 2016-1.2 (start uiterlijk 9 mei 2016)

Startdatum uitvoering afstudeeropdracht: 9 mei 2016

Inleverdatum afstudeerdossier volgens jaarrooster: 10 oktober 2016

Studentnummer: 20052930

Achternaam: dhr. Verhaar

Voorletters: JJM

Roepnaam: Jos

Adres: Meidoornstraat 11

Postcode: 2313 XD

Woonplaats: Leiden

Telefoonnummer: n.v.t.

Mobiel nummer: 06 22 18 18 17

Privé emailadres: josverhaar@gmail.com

Opleiding: Informatica

Locatie: Den Haag

Variant: deeltijd c.q. avond

Naam studieloopbaanbegeleider: Merie Heijne

Naam begeleidend examiner: Arie Toet

Naam tweede examiner: Gerard Mijnaerends

Naam bedrijf: Mailplus B.V.

Afdeling bedrijf: Development

Bezoekadres bedrijf: Boris Pasternaklaan 18

Postcode bezoekadres: 2719 DA

Postbusnummer: n.v.t.

Postcode postbusnummer: n.v.t.

Plaats: Zoetermeer

Telefoon bedrijf: 079 363 12 10

Telefax bedrijf: n.v.t.

Internetsite bedrijf: www.mailplus.nl

Achternaam opdrachtgever: dhr. Paul Nederlof

Voorletters opdrachtgever: P.V.

Titulatuur opdrachtgever: n.v.t.

Functie opdrachtgever: Teammanager Support

Doorkiesnummer opdrachtgever: 70

Email opdrachtgever: paulne@mailplus.nl

Achternaam bedrijfsmentor: dhr. Niels Noorlander

Voorletters bedrijfsmentor: N.E.

Titulatuur bedrijfsmentor: ir.

Functie bedrijfsmentor: Team Manager Product Management

Doorkiesnummer bedrijfsmentor: 079 363 7065

Email bedrijfsmentor: niels@mailplus.nl

Doorkiesnummer afstudeerder: n.v.t.

Functie afstudeerder (deeltijd/duaal): Medewerker Support

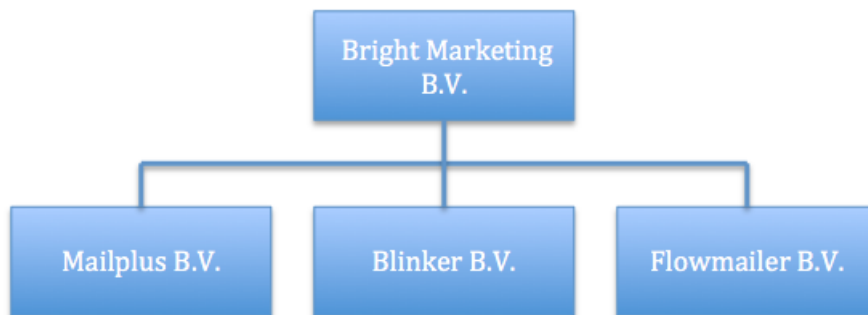
Titel afstudeeropdracht: "Ontwikkelen van een nieuw Support Feedback Systeem voor Mailplus B.V."

Opdrachtomschrijving

1. Bedrijf

Mailplus B.V. is een E-mail Service Provider (hierna: ESP). Het bedrijf levert een e-mail marketingtool met de naam MailPlus als SaaS oplossing. Het primaire doel van dit pakket is het versturen van marketing uitingen via e-mail. Hierbij is te denken aan nieuwsbrieven en serviceberichten. De software biedt verder de mogelijkheid tot het inrichten van uitgebreide campagnes. Bij deze campagnes kun je denken aan verlatenwinkelwagen campagne, verjaardag campagne, profiel verrijking campagne, enzovoort. Verder is het mogelijk om pagina's aan te maken waarop je formulieren, survey's, polls, en dergelijke kunt plaatsen. Mailplus B.V. verzorgt hiervan de verkoop, het onderhoud en uitbreiden van de software, het operationeel houden van de software en het geven van support aan de gebruikers. In de hoedanigheid van ESP zijn alle servers in eigenbeheer, maar extern gehuisvest.

Mailplus bestaat sinds 2004 en is een onderdeel van Bright Marketing Holding. Hieronder vallen eveneens Blinker B.V. en Flowmailer B.V. Binnen de Bright Marketing Holding zijn ongeveer 80 mensen werkzaam, waarvan 36 bij Mailplus B.V. In onderstaand organogram is de onderlinge verhouding inzichtelijk gemaakt van de verschillende B.V.'s.

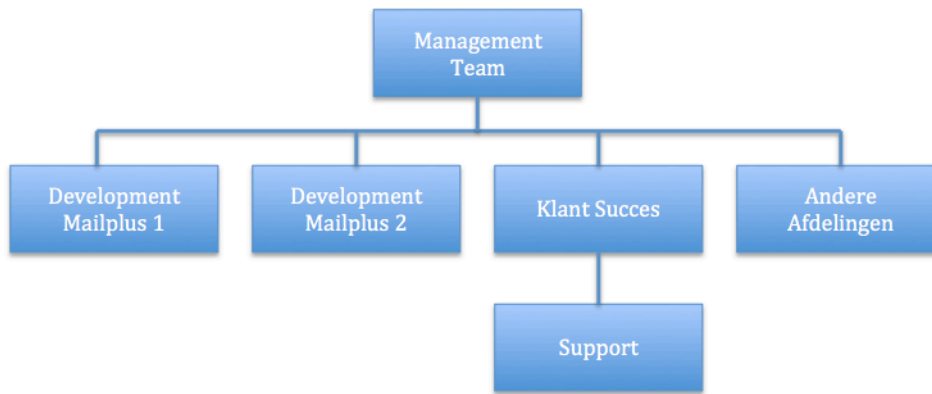


Figuur 1: Organogram Bright Marketing Holding

De afdeling support behandelt jaarlijks 20.000 support aanvragen van gebruikers. Ook collega's vallen onder de gebruikers. De afdeling support bestaat uit 8 personen. Een deel hiervan werkt parttime. De support aanvragen zijn verschillend van aard. Het kunnen eenvoudige vragen zijn over bijvoorbeeld e-mail templates, of meer ingewikkelde vragen over onze koppelingen. Hierbij kun je denken aan vragen over de werking en integratie van onze REST-API of SOAP-API.

De opdracht wordt uitgevoerd bij één van de twee product development teams. Deze teams bevinden zich direct onder het management team. Elk team is opgebouwd uit software developers en product managers. De product managers brengen klantwensen in kaart en beschrijven nieuwe features die vanuit de business units worden aangedragen. In samenwerking met de developers worden die wensen en features uitgewerkt en geïmplementeerd. Op dit moment werkt het team waar de opdracht uitgevoerd zal worden met name aan het implementeren van klantwensen. Het andere team richt zich op integraties tussen MailPlus en CRM/Webshop systemen. Beide teams werken volgens de Agile ontwikkelmethode SCRUM.

In onderstaand organogram is zichtbaar gemaakt hoe de support afdeling en beide development afdelingen zich verhouden binnen Mailplus B.V.



Afbeelding 1: Organogram Support ten opzichte van de Development teams

2. Probleemstelling

--Wat is het Support Feedback Systeem?

Het Support Feedback Systeem (hierna: SFS) is een geautomatiseerd systeem dat intern is ontwikkeld. Het doel van het systeem is het doorlopend meten van de klanttevredenheid. Het SFS wordt alleen intern gebruikt door de supportafdeling van Mailplus B.V. en haar gebruikers. Klanten geven feedback over de hulp die geleverd is door een support medewerker. De medewerkers van support worden beoordeeld op hun functioneren naar aanleiding van de statistieken die uit dit systeem worden gehaald. Een support medewerker verstuurt, na het sluiten van een support aanvraag, een feedback e-mail. De klant kan via deze mail de geleverde support beoordelen. Feedback kan gegeven worden door op één van de drie smiley-iconen te klikken. Deze smiley-iconen staan ieder voor een andere gradatie in tevredenheid. 'Ontevreden' (rood) staat voor de meest negatieve beoordeling, 'gewoon ok' (geel) staat voor voldoende support en 'zeer tevreden' (groen) staat voor zeer goede support. Desgewenst is er de mogelijkheid tot het toevoegen van een schriftelijke reactie. De gegeven feedback wordt via een scherm getoond op de afdeling.

De huidige functionaliteiten van het SFS omvatten het genereren van unieke links voor het kunnen geven van feedback. Daarnaast het genereren van een e-mail aan de klant met een vaste indeling en opmaak, het opslaan van ontvangen feedback en het opvragen van beperkte statistieken uit het systeem.

--Wat wil MailPlus B.V. bereiken?

Meerdere klanten hebben interesse getoond in het SFS. Naar aanleiding hiervan is besloten om van het SFS een product te maken, zodat klanten in staat zijn om op eenvoudige wijze de tevredenheid van hun eigen klanten doorlopend te kunnen meten. Doordat het SFS niet wordt gezien als een marketing tool wordt er met dit product een andere klantgroep aangesproken. Hierdoor is het niet wenselijk om MailPlus data te vermengen met de data van het SFS. Om deze reden is ervoor gekozen om het SFS apart op te zetten. Klanten kunnen door middel van authenticatie tokens gebruikmaken van het SFS. Zelf dienen ze een dashboard te ontwikkelen om de data te presenteren.

--Waarom voldoet de huidige implementatie niet?

Op dit moment kan het SFS niet één-op-één worden gebruikt door meerdere gebruikers. De huidige implementatie is ontwikkeld voor intern gebruik door de support afdeling van Mailplus B.V. Ook de functionaliteiten van het SFS zijn hierop gericht. Tijdens de ontwikkeling van het SFS is er geen rekening gehouden met meerdere gebruikers. Dit betekent bijvoorbeeld dat het database ontwerp enkel bedacht is voor de huidige vorm van feedback vragen (de drie smileys), dat de gegevens die je uit het systeem kunt opvragen beperkt zijn en dat de configuratie gedaan moet worden in de database. Het bericht dat verzonden wordt aan de klant is niet aan te passen en alleen beschikbaar in het Nederlands. Momenteel wordt er dan ook geen feedback gevraagd aan de Engels sprekende klanten van Mailplus B.V. Een groot probleem van het koppelen van meerdere gebruikers aan het huidige SFS is dat de gegeven feedback door elkaar gaat. Dit betekent dat iedere gebruiker, iedere feedback kan zien. Deze vermenging van gegevens is niet gewenst. Uit deze voorbeelden blijkt dat de huidige opzet niet voldoet om deze één-op-één in te kunnen zetten door de klanten van Mailplus B.V. Om dit te realiseren dient er een nieuwe opzet ontwikkeld te worden van het gehele SFS systeem.

Hierbij wordt rekening gehouden met wensen vanuit nieuwe- en bestaande gebruikers. Ook zal er worden gekeken naar wat vergelijkbare pakketten momenteel voor functionaliteit aanbieden.

3. Doelstelling van de afstudeeropdracht

Het doel van de afstudeeropdracht is het ontwikkelen van een nieuw SFS waarmee klanten hun klanttevredenheid doorlopend kunnen meten. Via een REST-API koppeling is het mogelijk om statistieken uit het systeem op te vragen en deze inzichtelijk te maken met een zelf te ontwikkelen dashboard.

4. Resultaat

Na het uitvoeren van de afstudeeropdracht heeft Mailplus B.V. een nieuw SFS. Het nieuwe SFS biedt klanten van Mailplus de mogelijkheid hun klanttevredenheid te registreren. Ook is het mogelijk om het systeem intern op meerdere afdelingen in te zetten. Het nieuwe SFS bestaat uit een backend met dataopslag en een REST-API waarmee gecommuniceerd kan worden. De front-end bestaat uit een webpagina waar gebruikers de feedback kunnen versturen en een reactie op kunnen plaatsen. Er zal een eenvoudige applicatie voor presentatie doeleinden worden ontwikkeld. De nieuwe applicatie wordt bij Mailplus gehost. Hierdoor hoeven gebruikers deze applicatie niet zelf te hosten. Door middel van een koppeling met hun eigen systeem kunnen ze gebruik maken van het nieuwe SFS.

5. Uit te voeren werkzaamheden, inclusief een globale fasering, mijlpalen en bijbehorende activiteiten

-) Opbouw afstudeerdossier
duur: 15 dagen
-) Schrijven plan van aanpak
duur: 1 dag
-) Inlezen huidige situatie (software en documentatie)
duur: 1 dag
-) Huidige situatie inzichtelijk maken (maken klassendiagram aan de hand van de aangeleverde code)
duur: 2 dagen
-) Concurrenten onderzoek
duur: 5 dagen
methoden: kwalitatief onderzoek
doel: onderzoeken wat gelijksoortige systemen doen/ hebben op het gebied van feedback systemen
-) Vaststellen functionele- en niet functionele requirements
duur: 7 dagen
methoden: interview technieken, requirements analyse
-) Ontwerp database
duur: 5 dagen
methoden:
-) Ontwerp software
duur: 14 dagen
methoden: Design Patterns, Software ontwikkelmethoden
-) Ontwerp tests
duur: 5 dagen
methoden: Test Goal, opstellen testgevallen
-) Coderen
duur: 21 dagen
-) Testen
duur: 4 dagen
methoden: test ontwerpen, TestGoal
-) Schrijven documentatie
duur: 4 dagen
-) Overdracht
duur: 1 dag

6. Op te leveren (tussen)producten

Tijdens het uitvoeren van dit afstudeerplan worden de volgende (tussen)producten opgeleverd:

- Plan van Aanpak:
Uitleg wat de opdracht inhoudt, inclusief een gedetailleerde planning
- Klassen- en Database diagram huidige implementatie
Om de huidige implementatie beter te begrijpen wordt hiervan een klassen- en database diagram opgesteld
- Uitwerking onderzoek naar gelijksoortige systemen
Het onderzoeksrapport van het gedane onderzoek naar gelijksoortige systemen op de markt
- Geïnterviewde (niet)functionele eisen
- Database ontwerp ter ondersteuning van de nieuwe applicatie
- Detail software ontwerp van de nieuwe applicatie (onder andere klassendiagram en use case diagrammen)
- Test ontwerp, -plan en -rapportage voor de nieuwe applicatie
- Geschreven code van de nieuwe applicatie
- Documentatie van de REST-API

7. Te demonstreren competenties en wijze waarop

-- 2.2 Ontwerpen, bouwen en bevragen van een database

niveau: 4 (zelfstandig, complex)

Het ontwerpen en implementeren van het gegevensmodel voor de relationele database. Het omzetten en implementeren van het fysieke ontwerp met behulp van SQL. Het uitdenken en het ontwerpen van de benodigde query's en constraints met behulp van ODL. De database draait op een Tomcat applicatie cluster met een MySQL database. Met het ontwerpen van de database structuur wordt rekening gehouden met concurrent gebruikers, performance- en beveiligingsaspecten.

-- 3.2 Ontwerpen systeemdeel

niveau: 4 (zelfstandig, complex)

Het in detail ontwerpen van de applicatie met behulp van design patterns en andere best-practices. Er zal rekening worden gehouden met de frameworks waarin de applicatie uiteindelijk wordt gebouwd.

-- 3.3 Bouwen applicatie

niveau: 3 (zelfstandig, lastig)

Het in delen implementeren en documenteren van het ontwerp tot één werkend systeem. De te ontwerpen applicatie zal geschreven worden in de programmeertaal Java. Hiervoor is gekozen omdat dit de standaard programmeertaal is binnen MailPlus. Er zal gebruik worden gemaakt van Spring MVC als web applicatie framework met Jersey (JAX-RS) voor de implementatie van de API. Voor persistence (JPA) zal gebruik worden gemaakt van EclipseLink. Als template engine wordt gebruik gemaakt van Velocity.

De user interface voor demonstratie doeleinde zal gemaakt worden in AngularJS met het Twitter Bootstrap framework.

-- 3.5 Uitvoeren van en rapporteren over het testproces

niveau: 3 (zelfstandig, lastig)

Het opstellen per testobject van de logische- en fysieke testgevallen, het bepalen van de testbasis en verwachte uitkomst. Vervolgens het uitvoeren van de tests en het noteren van de bevindingen. Aan de hand van de uitkomsten van de uitgevoerde tests zal een testrapport worden opgesteld. Als testmethode zal er gebruik worden gemaakt van TestGoal. Voor de uitvoering van geautomatiseerde testgevallen zal gebruik worden gemaakt van het pakket dat reeds in gebruik is binnen MailPlus en anders wordt JUnit gebruikt. Voor de demo interface zal voor unit tests Karma worden gebruikt en voor end-2-end tests Protractor.

Bijlage XII: Beoordeling

Faculteit IT & Design

Delft
Den Haag
Zoetermeer

Evaluatieformulier afstuderen

In te vullen door opdrachtgever c.q. bedrijfsmentor(en)

Student: Jos Verhaar

Periode:

Bedrijf c.q. instelling: MailPlus B.V.

Bedrijfsmentor: Niels Noorlander

Plaats: Zoetermeer

Datum: 6 januari 2017

1. Heeft de student zich zelf snel en goed ingewerkt in het bedrijf en de uit te voeren afstudeeropdracht?

Ja, waarbij Jos uiteraard het voordeel heeft gehad dat al bij ons bedrijf werkzaam is en dus vooral de uitdaging had om in een bestaand software project te duiken, dat matig gedocumenteerd is en buiten het dagelijks aandachtgebied van onze product development teams valt.

2. Hoe beoordeelt u de communicatieve vaardigheden van de student (in de samenwerking met collega's, in contacten met de opdrachtgever, bij mondelinge presentaties, schriftelijke rapportages)?

Op het communicatieve vlak zou ik de vaardigheden als voldoende beoordelen. In de schriftelijke communicatie komen nogal eens wat slordigheden voor die ik vooral wijt aan tijdsdruk en in veel mindere mate aan vaardigheid. In de samenwerking met collega's en opdrachtgever vind ik dat Jos heel goed zijn verhaal kan overbrengen, duidelijk formuleert en goed luistert.

3. Hoe heeft de student tijdens het uitvoeren van de opdracht gefunctioneerd?

- Qua verantwoordelijkheid ~~goed~~ / voldoende / ~~matig~~ / onvoldoende
- Qua zelfstandigheid goed / ~~voldoende~~ / ~~matig~~ / onvoldoende
- Qua planmatig werken ~~goed~~ / voldoende / ~~matig~~ / onvoldoende
- Qua creativiteit goed / voldoende / ~~matig~~ / onvoldoende
- Qua productiviteit ~~goed~~ / voldoende / ~~matig~~ / onvoldoende
- Qua samenwerken met collega's ~~goed~~ / voldoende / ~~matig~~ / onvoldoende
- Qua draagvlakontwikkeling ~~goed~~ / voldoende / ~~matig~~ / onvoldoende
- Qua inspelen op bedrijfscultuur goed / ~~voldoende~~ / ~~matig~~ / onvoldoende
- Qua rekening houden met de specifieke context van het bedrijf goed / ~~voldoende~~ / ~~matig~~ / onvoldoende
- Qua het op gang brengen van de nodige veranderingen ~~goed~~ / voldoende / ~~matig~~ / onvoldoende

4. Hoe beoordeelt u de kennis en kunde van de student in verhouding tot wat u verwacht van een bijna afgestudeerde?

Jos heeft wat mij betreft aangetoond bij het uitvoeren van zijn opdracht dat hij zowel de theorie als de praktijk kennis en kunde heeft die ik verwacht van een student in de laatste fase van zijn studie. Ook het uitzoeken van en inlezen in kennis die nieuw voor hem was tijdens het uitvoeren van de afstudeeropdracht is hem goed af gegaan.

Meeste moeite had Jos bij het realistisch plannen van zijn werkzaamheden en het bewaken van de planning.

5. Hoe beoordeelt u de kwaliteit van de opgeleverde (tussen)producten?

Goed: door de gestructureerde aanpak van ontwerp, implementatie en testen in een iteratief proces denk ik dat Jos een product heeft opgeleverd dat van goede kwaliteit is en een goede basis biedt om in de toekomst als een volwaardig product te gaan verkopen.

6. Bent u tevreden over het opgeleverde (eind)product?

- **In hoeverre heeft u gekregen wat is afgesproken?**

Uitgangspunt was vooral de ombouw van een maatwerk oplossing tot een SaaS product. Dat is gelukt in het door Jos opgeleverde product. Daarnaast heeft hij een aantal toevoegingen gedaan die hoog op de wensenlijst van de opdrachtgever stonden, waardoor deze nieuwe versie van het product ook al gewenst is voor intern gebruik, dus nog ongeacht of we het als product aan derden willen leveren. Dat is dus meer dan we in eerste instantie hadden afgesproken.

- **In hoeverre voldoet het (eind)product aan uw verwachtingen?**

Het voldoet volledig aan mijn verwachtingen.

- **Wat is de bruikbaarheid en onderhoudbaarheid hiervan?**

Zoals hier boven al aangegeven is het door Jos opgeleverde resultaat sowieso al inzetbaar als vervanging van het huidige maatwerk. Om het SFS ook als product te gaan verkopen zullen er nog extra stappen gezet moeten worden, die we met opzet buiten de scope van deze opdracht hebben gehouden, met name op het gebied van tooling voor de levering en support van het product, instellen van het product naar wens van de klant, etc.

- **Wat gebeurt er met het opgeleverde (eind)product?**

We zullen het product opnemen in het portfolio van onze product development en in eerste instantie in gebruik gaan nemen als vervanging van het huidige maatwerk. Daarnaast zullen we met commercieel product management een plan moeten maken welke stappen er gezet moeten worden om het product verder te ontwikkelen (ook wat betreft documentatie, pricing, support, etc.) om het tot een volwaardig SaaS product te laten groeien.

- **Kunt u direct met het opgeleverde product aan de slag?**

Ja, dat kan en gaan we ook doen.

7. Zijn er nog aspecten voor u van belang die nog niet aan de orde zijn geweest?

In eerste instantie leek Jos het uitvoeren van een afstudeeropdracht naast zijn werk als MailPlus support medewerker wat te onderschatten, maar de afgelopen maanden heeft hij er hard aan

getrokken met ondersteuning van het thuisfront, wat niet altijd even makkelijk is met een heel jong kind. Veel respect en waardering daar voor van mijn kant.

8. Bent u bereid een volgende keer weer uw medewerking te verlenen aan het beschikbaar stellen van een afstudeerplaats (graag met toelichting)?

Ja, dat lijkt me zeker, hoewel ik dan wel zou willen gaan voor de situatie waarbij de afstudeerder het gros van zijn of haar tijd bij ons in de development teams aan de slag gaat, omdat het uitvoeren van de opdracht volledig in eigen tijd zowel zwaar is voor de student maar ook suboptimaal voor de begeleiding.

Bijlage XIII: Plan van aanpak

Plan van aanpak

Auteur: Jos Verhaar (20052930)
Bedrijf: Mailplus B.V.
Opdrachtgever: Paul Nederlof
Bedrijfsmentor: Niels Noorlander
Examinatoren: Arie Toet & Gerard Mijnares
Datum: 9 januari 2017
Instelling: Haagse Hogeschool
Stageperiode: juni 2016-februari 2017

1. Inleiding

1.1 Doel van het document

In het verslag wordt de uitvoering van de afstudeeropdracht beschreven. Het verslag is geschreven met als doel om geïnteresseerden te informeren over het proces van de afstudeeropdracht.

In het verslag wordt het proces om het Support Feedback Systeem (hierna: SFS) te ontwikkelen, beschreven. Het gehele proces en de bijbehorende keuzes die gedurende de ontwikkeling zijn gemaakt worden hierin beschreven.

1.2 Leeswijzer

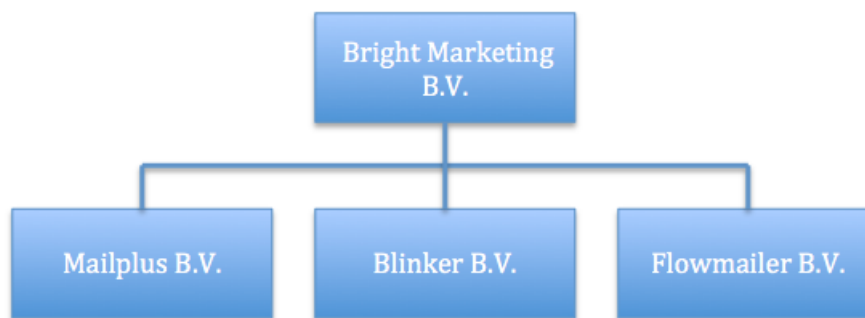
Vanwege de leesbaarheid van het document zal een “veel gebruikte termen” lijst worden opgesteld. Met name om hierin de bedrijfsterminologie en de technische termen in te verduidelijken. Voor de rest wordt aangenomen dat de lezer basiskennis bezit van software ontwikkeling.

De afstudeeropdracht is uitgevoerd bij Mailplus B.V. In hoofdstuk twee wordt een organisatieverschrijving gegeven. Alsmede een omschrijving van de werkomgeving en tot slot een overzicht van alle betrokkenen bij deze afstudeeropdracht. Vervolgens wordt in hoofdstuk drie een uitgebreide opdrachtdefinitie gegeven. In hoofdstuk vier worden de methoden en technieken, die tijdens het onderzoek zijn gebruikt, beschreven. In hoofdstuk 5 en 6 wordt zowel de aanpak als de uitvoering beschreven. Na deze hoofdstukken volgen de sprint hoofdstukken. In elk hoofdstuk wordt per sprint beschreven wat er gedaan is, hoe het gedaan is en welke keuzes zijn gemaakt. Aan het einde van elke sprint wordt er gereflecteerd op het proces. Tot slot zijn in de laatste hoofdstukken de procesevaluatie en productevaluatie opgenomen. Aan het einde van het document zijn de bijlagen opgenomen.

2. Het bedrijf

2.1 Organisatiebeschrijving

Mailplus B.V. is een E-mail Service Provider (hierna: ESP). Het bedrijf levert een e-mail marketingtool met de naam MailPlus als SaaS oplossing. Mailplus bestaat sinds 2004 en is een onderdeel van Bright Marketing Holding. Hieronder vallen eveneens Blinker B.V. en Flowmailer B.V. Binnen de Bright Marketing Holding zijn ongeveer 80 mensen werkzaam, waarvan 36 bij Mailplus B.V. In onderstaand organogram is de onderlinge verhouding inzichtelijk gemaakt van de verschillende B.V.'s.

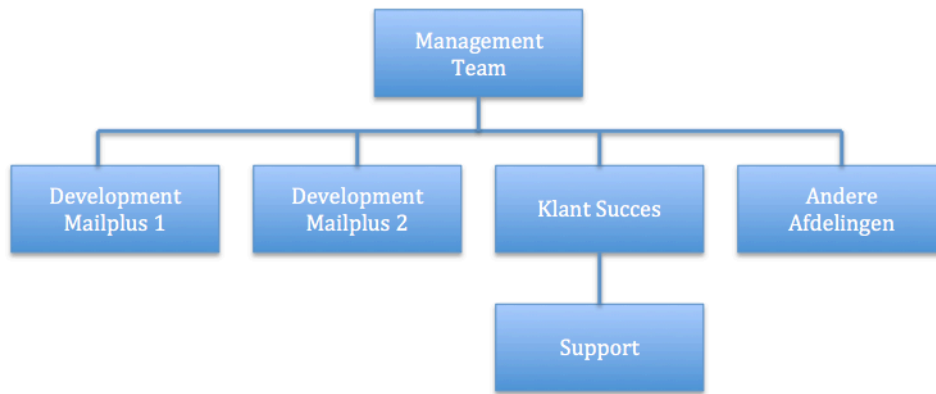


Figuur 3: Organogram Bright Marketing Holding

De afdeling support behandelt jaarlijks 20.000 support aanvragen van gebruikers. Ook collega's vallen onder de gebruikers. De afdeling support bestaat uit 8 personen. Een deel hiervan werkt parttime. De support aanvragen zijn verschillend van aard. Het kunnen eenvoudige vragen zijn over bijvoorbeeld e-mail templates, of meer ingewikkelde vragen over onze koppelingen. Hierbij kun je denken aan vragen over de werking en integratie van onze REST-API of SOAP-API.

De opdracht wordt uitgevoerd bij één van de twee product development teams. Deze teams bevinden zich direct onder het management team. Elk team is opgebouwd uit software developers en product managers. De product managers brengen klantwensen in kaart en beschrijven nieuwe features die vanuit de business units worden aangedragen. In samenwerking met de developers worden die wensen en features uitgewerkt en geïmplementeerd. Op dit moment werkt het team waar de opdracht uitgevoerd zal worden met name aan het implementeren van klantwensen. Het andere team richt zich op integraties tussen MailPlus en CRM/Webshop systemen. Beide teams werken volgens de Agile ontwikkelmethode Scrum.

In onderstaand organogram (figuur 2) is zichtbaar gemaakt hoe de support afdeling en beide development afdelingen zich verhouden binnen Mailplus B.V.



Figuur 4: Organogram Support ten opzichte van de Development teams.

2.2 Jouw werkomgeving

Het primaire doel van het MailPlus pakket is het versturen van marketing uitingen via e-mail. Hierbij is te denken aan nieuwsbrieven en serviceberichten. De software biedt verder de mogelijkheid tot het inrichten van uitgebreide campagnes. Bij deze campagnes kun je denken aan verlatenwinkelwagen campagne, verjaardag campagne, profiel verrijking campagne, enzovoort. Verder is het mogelijk om pagina's aan te maken waarop je formulieren, survey's, polls, en dergelijke kunt plaatsen. Mailplus B.V. verzorgt hiervan de verkoop, het onderhoud en uitbreiden van de software, het operationeel houden van de software en het geven van support aan de gebruikers. In de hoedanigheid van ESP zijn alle servers in eigenbeheer, maar extern gehuisvest.

2.3 Betrokkenen afstudeeropdracht

Bij de afstudeeropdracht zijn een aantal medewerkers van Mailplus B.V. betrokken. In dit document zal worden gerefereerd naar de rollen.

Naam	Email	Rol
Jos Verhaar	josverhaar@gmail.com jos@mailplus.nl	Afstudeerder
Paul Nederlof	paulne@mailplus.nl	Opdrachtgever
Niels Noorlander	niels@mailplus.nl	Bedrijfsmentor

3. Opdrachtdefinitie

3.1 Opdrachtoomschrijving

3.1.1 Wat is het Support Feedback Systeem?

Het Support Feedback Systeem (hierna: SFS) is een geautomatiseerd systeem dat intern is ontwikkeld. Het doel van het systeem is het doorlopend meten van de klanttevredenheid. Het SFS wordt alleen intern gebruikt door de supportafdeling van Mailplus B.V. en haar gebruikers. Klanten geven feedback over de hulp die geleverd is door een support medewerker. De medewerkers van support worden beoordeeld op hun functioneren naar aanleiding van de statistieken die uit dit systeem worden gehaald.

3.1.2 Huidige implementatie

Wanneer een support medewerker de vraag van een gebruiker heeft afgehandeld, wordt er door TopDesk een e-mail gegenereerd met een JSON-object. Deze e-mail is de trigger voor het SFS.



E-mail versturen voor I1508 964
Feedback support

Algemeen Extra Bijlagen

Adres

Afzender:

Naar ->

☐ Kopie naar afzender

Bericht

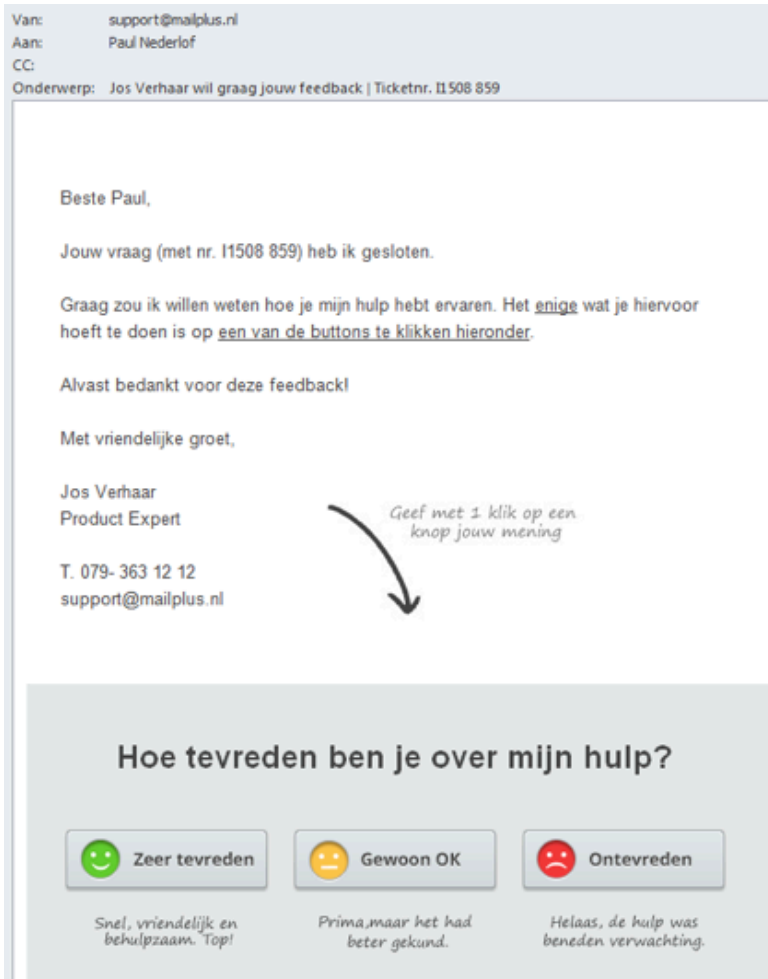
Onderwerp:

☐ HTML e-mail

```
{
  "id": "I1508 964",
  "requestor": {
    "firstname": "Paul",
    "lastname": "Nederlof",
    "email": "paulne@mailplus.nl"
  },
  "operator": {
    "firstname": "Ralph",
    "lastname": "van Delft",
    "email": "ralph@mailplus.nl"
  },
  "account": "Medewerkers",
  "number": "8"
}
```

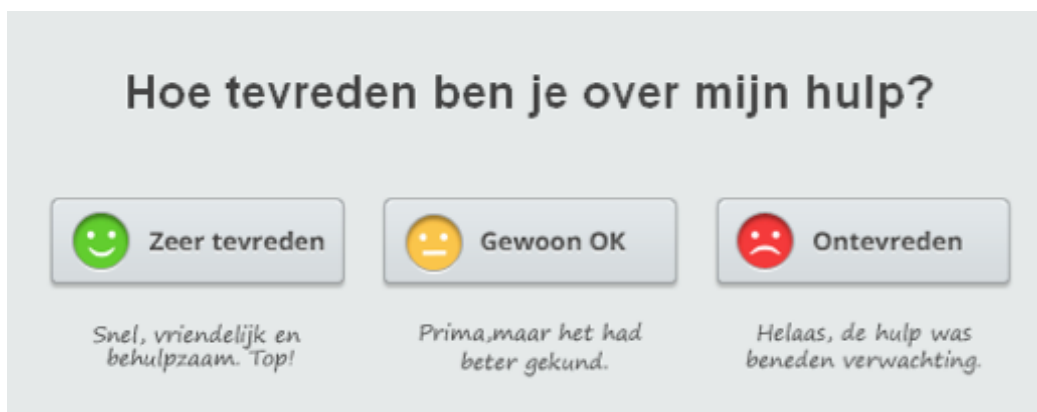
Afbeelding 54: Mail met JSON-object dat vanuit TopDesk wordt gestuurd

Het SFS leest de e-mail in en verwerkt deze tot een andere e-mail, het feedbackverzoek, welke naar de gebruiker wordt verstuurd.



Afbeelding 55: Voorbeeld van de feedbackmail

Vanuit het feedbackverzoek kan de gebruiker zijn waardering geven. Het enige dat de gebruiker hiervoor hoeft te doen is te klikken op een van de smiley-iconen onder in de mail.



Afbeelding 56: Close-up smiley's in de feedbackmail

Na het klikken heeft de gebruiker zijn waardering gegeven en wordt de gebruiker naar een dankpagina geleid. Op deze pagina is er de mogelijkheid tot het geven van een schriftelijke reactie.



Afbeelding 57: Landingspagina met mogelijkheid tot geven toelichting

Een support medewerker verstuurt, na het sluiten van een support aanvraag, een feedback e-mail. De klant kan via deze mail de geleverde support beoordelen. Feedback kan gegeven worden door op één van de drie smiley-iconen te klikken. Deze smiley-iconen staan ieder voor een andere gradatie in tevredenheid. 'Ontevreden' (rood) staat voor de meest negatieve beoordeling, 'gewoon ok' (geel) staat voor voldoende support en 'zeer tevreden' (groen) staat voor zeer goede support. Desgewenst is er de mogelijkheid tot het toevoegen van een schriftelijke reactie. De gegeven feedback wordt via een scherm getoond op de afdeling. De huidige functionaliteiten van het SFS omvatten het genereren van unieke links voor het kunnen geven van feedback. Daarnaast het genereren van een e-mail aan de klant met een vaste indeling en opmaak, het opslaan van ontvangen feedback en het opvragen van beperkte statistieken uit het systeem.

3.1.3 Wat wil MailPlus B.V. bereiken?

Meerdere klanten hebben interesse getoond in het SFS. Naar aanleiding hiervan is besloten om van het SFS een product te maken, zodat klanten in staat zijn om op eenvoudige wijze de tevredenheid van hun eigen klanten doorlopend te kunnen meten. Doordat het SFS niet wordt gezien als een marketing tool wordt er met dit product een andere klantgroep aangesproken. Hierdoor is het niet wenselijk om MailPlus data te vermengen met de data van het SFS. Om deze reden is ervoor gekozen om het SFS apart op te zetten. Klanten kunnen door middel van authenticatie tokens gebruikmaken van het SFS. Zelf dienen ze een dashboard te ontwikkelen om de data te presenteren.

3.1.4 Waarom voldoet de huidige implementatie niet?

Op dit moment kan het SFS niet één-op-één worden gebruikt door meerdere gebruikers. De huidige implementatie is ontwikkeld voor intern gebruik door de support afdeling van Mailplus B.V. Ook de functionaliteiten van het SFS zijn hierop gericht. Tijdens de ontwikkeling van het SFS is er geen rekening gehouden met meerdere gebruikers. Dit betekent bijvoorbeeld dat het database ontwerp enkel bedacht is voor de huidige vorm van feedback vragen (de drie smileys), dat de gegevens die je uit het systeem kunt opvragen beperkt zijn en dat de configuratie gedaan moet worden in de database. Het bericht dat verzonden wordt aan de klant is niet aan te passen en alleen beschikbaar in het Nederlands. Momenteel wordt er dan ook geen feedback gevraagd aan de Engels sprekende klanten van Mailplus B.V. Een groot probleem van het koppelen van meerdere gebruikers aan het huidige SFS is dat de gegeven feedback door elkaar gaat. Dit betekent dat iedere gebruiker, iedere feedback kan zien. Deze vermenging van gegevens is niet gewenst. Uit deze voorbeelden blijkt dat de huidige opzet niet voldoet om deze één-op-één in te kunnen zetten door de klanten van Mailplus B.V. Om dit te realiseren dient er een nieuwe opzet ontwikkeld te worden van het gehele SFS systeem. Hierbij wordt rekening gehouden met wensen vanuit nieuwe- en bestaande gebruikers. Ook zal er worden gekeken naar wat vergelijkbare pakketten momenteel voor functionaliteit aanbieden.

3.2 Doelstelling

Het doel van de afstudeeropdracht is het ontwikkelen van een nieuw SFS waarmee klanten hun klanttevredenheid doorlopend kunnen meten. Via een REST-API koppeling is het mogelijk om statistieken uit het systeem op te vragen en deze inzichtelijk te maken met een zelf te ontwikkelen dashboard.

3.3 Gewenste resultaat

Na het uitvoeren van de afstudeeropdracht heeft Mailplus B.V. een nieuw SFS. Het nieuwe SFS biedt klanten van Mailplus B.V. de mogelijkheid hun klanttevredenheid te registreren. Ook is het mogelijk om het systeem intern op meerdere afdelingen in te zetten. Het nieuwe SFS bestaat uit een backend met dataopslag en een REST-API waarmee gecommuniceerd kan worden. De front-end bestaat uit een webpagina waar gebruikers de feedback kunnen versturen en een reactie op kunnen plaatsen. Er zal een eenvoudige applicatie voor presentatie doeleinden worden ontwikkeld. De nieuwe applicatie wordt bij Mailplus B.V. gehost. Hierdoor hoeven gebruikers deze applicatie niet zelf te hosten. Door middel van een koppeling met hun eigen systeem kunnen ze gebruik maken van het nieuwe SFS.

3.4 Op te leveren producten

Tijdens het uitvoeren van deze afstudeeropdracht worden de volgende (tussen)producten opgeleverd:

- Plan van Aanpak: Uitleg wat de opdracht inhoudt, inclusief een gedetailleerde planning.
- Klassen- en Database diagram huidige implementatie: Om de huidige implementatie beter te begrijpen wordt hiervan een klassen- en database diagram opgesteld.
- Uitwerking onderzoek naar gelijksoortige systemen (concurrentie onderzoek). Het onderzoeksrapport van het gedane onderzoek naar gelijksoortige systemen op de markt.

- Geïntariseerde (niet)functionele eisen.
- Database ontwerp ter ondersteuning van de nieuwe applicatie.
- Detail software ontwerp van de nieuwe applicatie (onder andere klassendiagram en use case diagrammen).
- Test ontwerp, -plan en -rapportage voor de nieuwe applicatie.
- Documentatie van de REST-API.
- Het afstudeerverslag.

4. Randvoorwaarden

In dit hoofdstuk worden de afspraken rondom dit afstudeeropdracht beschreven. Tevens zal de scope en hierin de afbakening van de opdracht worden aangegeven.

4.1 Afspraken.

4.1.1 Sprint planning

Een sprint duurt twee weken. Elke week wordt de voortgang van de sprint besproken met de bedrijfsmentor. Aan het einde van de sprint wordt er, samen met de bedrijfsmentor, teruggekeken hoe de sprint is gegaan, de sprint retrospective, en welke producten er zijn opgeleverd. Tevens wordt er een planning en indeling gemaakt voor de volgende sprint.

4.1.2 Afstudeerverslag

Elke maandag zal er gewerkt worden aan het verslag. Tevens wordt hier aan gewerkt tijdens de avonden en in de weekenden.

4.1.3 Afspraken met de Haagse Hogeschool

Na ongeveer 25% van de looptijd zal de eerste stagecoördinator bij het bedrijf op bezoek komen. Na ongeveer 45% van de doorlooptijd zal een voortgangsverslag naar de eerste stagecoördinator gestuurd worden. Met het bijhorende afstudeerdossier dat alle gemaakte producten bevat. Bij ongeveer 60% van de doorlooptijd van de afstudeeropdracht zal er tijdens een tussentijds assessment het afstudeerdossier worden besproken met zowel de eerste stagecoördinator als de tweede, begeleidend stagecoördinator.

4.1.4 Afspraken met bedrijfsmentor

Vanwege de aanwezigheid van mogelijke risicofactoren die het onderzoek kunnen belemmeren (zie voor verdere informatie hoofdstuk 5 uit het plan van aanpak), zijn er vaste afspraken met de bedrijfsmentor gemaakt. Zo is er elke maandag een overleg gepland om de voortgang van de afstudeeropdracht te bespreken, maar zo ook de wensen en de eventuele wijzigingen hierin met de bedrijfsmentor te bespreken. Om ervoor te zorgen dat de bedrijfsmentor up to date is wat betreft de voortgang van het onderzoek, heeft de bedrijfsmentor toegang tot deze documenten middels een Google Drive account. De bedrijfsmentor levert hierop zijn input. Dit wordt tevens besproken tijdens de vaste overlegmomenten.

4.1.5 Scope van de afstudeeropdracht

Tijdens de besprekingen met de bedrijfsmentor zijn er afspraken gemaakt van de complexiteit van de opdracht. Er is gekozen om het nieuwe SFS als losstaand product te ontwikkelen. Hier is voor gekozen omdat het SFS niet wordt gezien als marketingtool. Door deze visie is het ook niet wenselijk dat het SFS wordt geïntegreerd in MailPlus of dat het SFS gebruik kan maken van data vanuit MailPlus.

De communicatie met het SFS gebeurt voornamelijk via de REST-API. Via deze API is het mogelijk om feedback verzoeken te versturen naar klanten, om gegeven feedback op te slaan en om statistieken uit het systeem op te vragen. Middels deze functionaliteiten is het

voor gebruikers van het systeem mogelijk om een eigen dashboard te ontwikkelen. De webpagina's voor het geven en bewerken van de feedback en schriftelijke reactie vallen binnen de opdracht. Om de complexiteit en omvang van de afstudeeropdracht beperkt te houden en om gebruikers meer vrijheid te geven is er, in overleg met de bedrijfsmentor, voor gekozen om het ontwikkelen van een dashboard buiten de scope van de opdracht te houden. Om meer dan alleen regels code te kunnen laten zien tijdens mijn verdediging zal er een eenvoudig dashboard worden ontwikkeld.

5. Risico's

Er zijn een aantal risicofactoren te benoemen die gedurende de uitvoering van de afstudeeropdracht zijn te verwachten. Allereerst het niet beschikken over voldoende kennis om de afstudeeropdracht tot een goed einde te brengen. Dit risico is te ondervangen door dit tijdig te herkennen en mijzelf in te lezen op het onderwerp en om de bedrijfsmentor in te schakelen. Het tijdig herkennen van het niet hebben van de juiste kennis kan zichtbaar worden doordat een onderdeel langer duurt dan gepland. Echter is ook mogelijk dat dit onderdeel complexer blijkt dan van tevoren is ingeschat. Een gevolg kan zijn dat het resultaat hieronder lijdt en/of de planning aangepast dient te worden zodat deze weer realistisch is.

Ten tweede tijdens de uitvoering van de afstudeeropdracht zou ik in tijdsnood kunnen komen. Dit risico kan worden beperkt door het maken van een degelijke planning en deze planning naargelang de voortgang aan te passen. Oorzaken kunnen zijn dat een onderdeel complexer is dan vooraf verwacht of de planning te optimistisch is. Door de voortgang te bewaken en hierop de planning aan te passen, verwacht ik dit risico te kunnen beperken.

Ten derde kan de afhankelijkheid van input van mijn bedrijfsmentor een knelpunt vormen tijdens de uitvoering van deze opdracht. Het is hierdoor van belang dat er vaste afspraken worden gemaakt waarop er gezamenlijk vergaderd kan worden over de voortgang en de input van de afstudeeropdracht. Om dit te ondervangen zijn er over geruime tijd vaste afspraken ingepland. Tevens heeft de bedrijfsmentor toegang tot de documenten die mijn voortgang vastleggen. Hierdoor is de bedrijfsmentor up to date over de voortgang van de opdracht.

6. Contact & rollen

Alle betrokken bij dit afstudeeronderzoek worden in onderstaande tabel genoemd.

Naam	Instantie	Email	Rol
Jos Verhaar		josverhaar@gmail.com jos@mailplus.nl	Afstudeerder
Arie Toet	De Haagse Hogeschool	a.j.toet@hhs.nl	Begeleidend examinator
Gerard Mijharends	De Haagse Hogeschool	g.a.mijnarends@hhs.nl	Tweede examiner
Paul Nederlof	Mailplus B.V.	paulne@mailplus.nl	Opdrachtgever
Niels Noorlander	Mailplus B.V.	niels@mailplus.nl	Bedrijfsmentor en product owner

7. Methodologie en planning

7.1 Methode

Als methode voor de verwerking van de afstudeeropdracht wordt gebruik gemaakt van Scrum. Op de development afdeling van Mailplus B.V. wordt gewerkt met Scrum. Om deze reden is ook tijdens deze opdracht gewerkt met de methodologie Scrum. Scrum is de manier van werken die de effectiviteit van teams verhoogt. Het voordeel van Scrum is dat projecten worden opgedeeld in korte sprints, waarin tussenproducten worden opgeleverd. Hierdoor is het zichtbaar waaraan gewerkt wordt en wat de voortgang is. Regelmatig overleg en feedback zorgt er tevens voor dat er een effectieve en intensieve samenwerking is wat resulteert in een effectievere aanpak van de opdracht (de Boer, Bruggink, Bruns et al., 2015).

Vanwege bovenstaande voordelen is daarom ook gekozen om Scrum te gebruiken tijdens deze afstudeeropdracht. Om het overzicht en de structuur te bewaren wordt door van Loon, Barendsen & Hoppenbrouwers (2015) aangeraden om gebruik te maken van een Scrum board. Dit wordt op zowel de development afdeling als tijdens deze opdracht digitaal bijgehouden middels Jira. Op dit Scrumboard worden de volgende drie kolommen bijgehouden; to do, in progress en done.

Een sprint duurt twee weken. Na deze twee weken wordt teruggekeken hoe de sprint is gegaan, de sprint retrospective, en welke producten er zijn opgeleverd. Tevens wordt er een planning en indeling gemaakt voor de volgende sprint.

7.2 Planning

	ma		wo		do		za		ma		wo		do		za		ma		wo		do		za		ma		wo		do		za	
Sprint #	0	0	1	1	1	1	2	2	VAK	VAK	2	2	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
Taak	38		39		40		41		42		43		44		45																	
Inlezen huidige situatie																																
Schrijven Plan van aanpak																																
Huidige situatie inzichtelijk maken																																
Concurrenten onderzoek																																
Vaststellen (niet-) functionelen eisen																																
Ontwerp Database																																
Bedrijfsbezoek																																
Ontwerp Software																																
Ontwerp Tests																																
Coderen																																
Voortgangsverslag																																
Testen																																
Schrijven documentatie																																
Retrospective																																
Tussentijds assessment																																
Opbouw Afstudeerdossier																																
Overdracht																																
Inleveren Afstudeerdossier																																

Afbeelding 58: Planning sprint 0 t/m 3

[illegible]

8. Stakeholders

Tijdens de afstudeeropdracht zijn een aantal groepen stakeholders betrokken. Zij worden per groep gedefinieerd. In dit hoofdstuk zal worden aangegeven wie als vertegenwoordiger optreedt van deze groepen.

8.1 Groepen en hun vertegenwoordiging

8.1.1 Geïnteresseerde klanten

Deze groep bestaat uit klanten van Mailplus die interesse hebben getoond in het SFS. Deze personen zijn de toekomstige gebruikers van het SFS en zullen feedbackverzoeken gaan verzenden en statistieken gaan opvragen met het SFS. Deze groep zal vertegenwoordigd worden door Paul Nederlof. Hij heeft vanaf het in gebruik nemen van het SFS tot nu gesprekken gevoerd met geïnteresseerde klanten. Tijdens deze gesprekken heeft hij hun wensen en ideeën in kaart gebracht.

8.1.2 Huidige gebruikers

Met de huidige gebruikers worden de medewerkers van de afdeling support bij Mailplus B.V. bedoeld. Zij zijn de personen die dagelijks werken met het huidige SFS. Zij verzenden feedbackverzoeken en bekijken de statistieken die het dashboard toont. De huidige gebruikers worden vertegenwoordigd door de opdrachtgever. Hij is de direct leidinggevende van de afdeling support. Ook kunnen, wanneer nodig, andere medewerkers van de afdeling worden geraadpleegd.

8.1.3 Product Management

Product Management is de afdeling binnen Mailplus B.V. die de requirements van het huidige SFS hebben bedacht. Zij zijn nauw betrokken geweest met het uitdenken en implementeren van de huidige versie van het SFS. Deze groep zal vertegenwoordigd worden door de bedrijfsmentor. Hij is de direct leidinggevende van deze afdeling en was zelf betrokken bij de ontwikkeling van het huidige SFS.

Auteur: Jos Verhaar (20052930)
Bedrijf: Mailplus B.V.
Opdrachtgever: Paul Nederlof
Bedrijfsmentor: Niels Noorlander
Examinatoren: Arie Toet & Gerard Mijnares
Datum: 9 januari 2017
Instelling: Haagse Hogeschool
Stageperiode: mei 2016-januari 2017