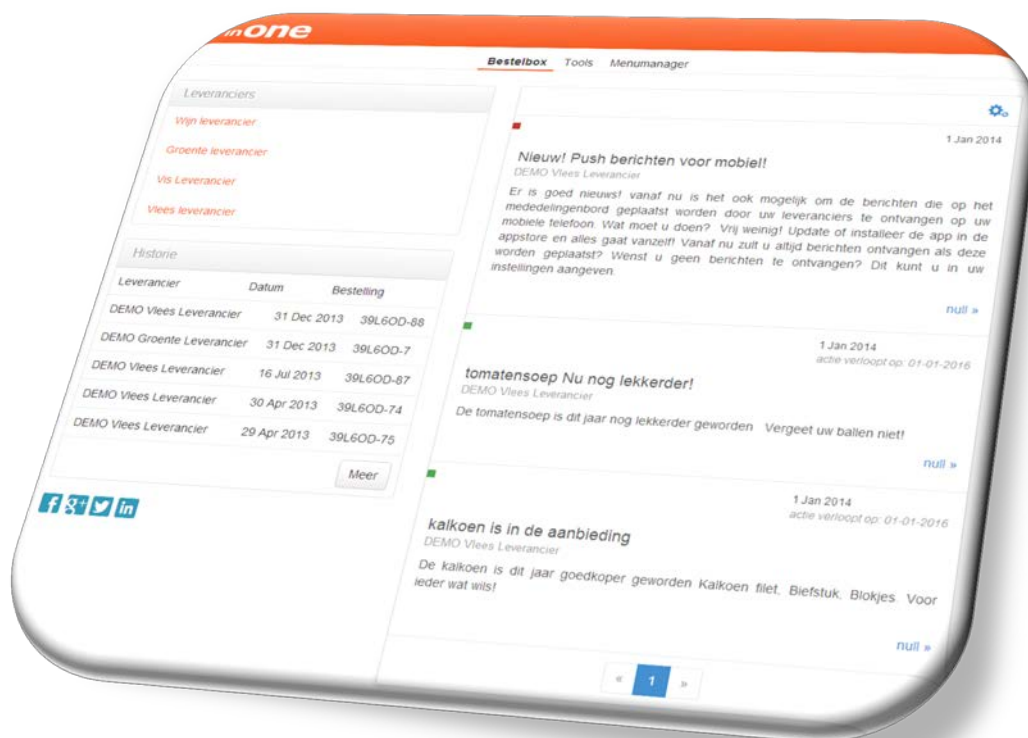


Scriptie Afstuderen inONE

Berichten service
Marcel Fasseur



Student: Marcel Fasseur
Studentnummer: 10069968
Onderwijsinstelling: De Haagse Hogeschool
Opleiding: HBO Informatica

Studieloopbaanbegeleider: Mevr. H.G.J. Bechet-Tjoonk
Begeleidend examiner: Dhr. G.A. Mijnaerends
Tweede examiner: Mevr. G.E. in 't Veld

Datum: 8 januari 2014
Versie: 1.03
Status: Document afgerond

Versiebeheer

Versie	Wijziging(en)	Datum
0.1	Opstellen document; Aantal hoofdstukken ingevuld.	20-08-2013
0.2	Voorwoord toegevoegd	30-08-2013
0.3	Beroepstaak 3.2 ingevuld	30-10-2013
0.4	Beroepstaak 3.3 ingevuld	16-11-2013
0.5	Beroepstaak 3.3 verder aangevuld	28-11-2013
0.6	Systeemontwikkelmethode verder aangevuld	02-12-2013
0.7	Document heringedeeld	07-12-2013
0.8	Aantal hoofdstukken aangevuld.	12-12-2013
0.9	Deel van feedback verwerkt	16-12-2013
0.10	Indeling herzien	18-12-2013
0.11	Sprints ingevuld	1-1-2014
1.0	Versie naar 1.0 gezet	5-1-2014
1.01	Probleem met hoofdstuknummering opgelost	6-1-2014
1.02	Feedback verwerkt	7-1-2014
1.03	Document afgemaakt	8-1-2014

Referaat

Afstudeeropdracht “Berichtenservice”

Fasseur M.A. 10069968

inONE , Leiden 12 September 2013 – 10 Januari 2014

Dit document geeft inzicht in de werkzaamheden van het n afstudeerproject van Marcel Fasseur bij inONE te Leiden. Het document gaat in op de werkzaamheden, op welke manier ze zijn uitgevoerd en waarom deze manier gekozen is.

Descriptoren:

- Mockups
- UML
- JAVA
- XPages
- Android
- IOS
- inONE
- Berichtenservice
- Bestelbox
- My SQL
- Lotus Domino

Voorwoord

Dit afstudeerverslag is geschreven in het kader van mijn opleiding informatica aan de Haagse Hogeschool, Academie voor ICT & Media. Dit verslag geeft inzicht in de afstudeerperiode van 12 sept 2013. Tot en met 10 jan. 2014 welk ik heb doorlopen bij inONE.

Ik wil inONE bedanken voor de afstudeer mogelijkheid welke voor mij gecreëerd is. Ik wil Rogier Bruin en Leon Belt bedanken voor de opdracht en de ondersteuning hiervan. Ook wil ik Renier van Dijk en Erik Bruin bedanken voor de ondersteuning bij het programmeren. En ten slotte bij inONE wil ik Aad van Tol en Rick Spijker bedanken voor het fine tunen van de design welke ik gemaakt heb.

Dhr. G.A. Mijnaerends en Mevr. G.E. in 't Veld wil ik bedanken voor de ondersteuning vanuit school. Dhr. G.A. Mijnaerends als begeleidende examinerator en Mevr. G.E. in 't Veld als tweede examinerator.

In het bijzonder wil mijn vriendin bedanken voor de steun, hulp en ruimte die ze mij heeft gegeven om mijn opdracht tot een succes te brengen.

Marcel Fasseur

Leiden, donderdag 8 januari 2014

Inhoudsopgave

Versiebeheer	I
Referaat	II
Voorwoord.....	ii
Inleiding	2
1 Organisatie	3
1.1 Missie	4
1.1 Processen	4
2 De opdracht.....	5
2.1 Probleemstelling	5
2.1.1 Oude situatie	5
2.1.2 Nieuwe situatie.....	5
2.2 Doelstelling	7
2.3 Resultaat	7
2.4 Producten.....	8
3 Software ontwikkelmethode.....	9
3.1 Afwijking van SCRUM	10
4 Oriënterende fase (Fase 1).....	11
4.1 Requirements.....	12
4.2 Mockups:.....	14
4.2.1 Mededelingenboard	14
4.2.2 Filteren.....	14
4.3 Nieuw bericht.....	15
4.4 PersonalMessage	16
4.4.1 Statusbalk	16
4.4.2 Berichten overzicht.....	16
4.5 Klassediagram en daarbij gemaakte keuzes	17
4.5.1 Specialisatie van berichten	17
4.5.2 Notificaties.....	18
4.5.3 Instellingen	19
4.5.4 Persoonlijk bericht.....	19
4.5.5 Taalondersteuning.....	20
4.6 MySQL + Keuzes	21
4.7 XPages	22
5 De sprints (Fase 2).....	22
5.1 Sprint 1	23
5.1.1 MVC	23
5.1.2 My SQL Werking in XPages	24
5.1.3 Het bouwen	25
5.1.4 Black box Interface test	29
5.2 Sprint 2	30
5.2.1 Testen	31

5.3	Sprint 3	33
5.3.1	Testen	35
5.4	Sprint 4	36
5.4.1	Testen	38
5.5	Sprint 5	39
5.5.1	Testen	42
6	Deployment diagram.....	43
7	Product evaluatie	44
8	Project evaluatie	44
9	Verantwoording beroepstaken	45
9.1	Beroepstaak 3.2 Ontwerpen systeemdeel.....	45
9.2	Beroepstaak 3.3 Bouwen applicatie	46
9.3	Beroepstaak 3.5 Uitvoeren en rapporteren over het testproces	47
10	Begrippenlijst	48
11	Bijlagen	50
	Bijlage 1. Afstudeerplan	
	Bijlage 2. Plan van Aanpak.....	
	Bijlage 3. Product backlog	
	Bijlage 4. Notulen	
	Bijlage 5. Klassediagram.....	
	Bijlage 6. SQL DDL.....	
	Bijlage 7. Mockups	
	Bijlage 8. Code voorbeeld filteren bericht	
	Bijlage 9. Test plan	
	Bijlage 10. Test rapportage	

Inleiding

Dit document is gemaakt om inzicht te tonen in het verloop van mijn afstuderen bij inONE te Leiden. Het doel van dit document is om stapsgewijs te laten zien hoe ik het proces heb doorlopen van mijn afstudeerstage.

Hoofdstuk 1 en 2 zijn de introducerende hoofdstukken. Hoe ziet de organisatie er uit en wat is de huidige situatie en wat wordt de nieuwe situatie. In hoofdstuk 3 staat welke software ontwikkelmethode ik gebruikt heb. Hoofdstuk 4 en 5 is wat ik gedaan heb in 2 fases verdeeld. Hoofdstuk 4 is de oriënterende fase en hoofdstuk 5 is de 2e fase met daarin de sprints en het testen. In hoofdstuk 6 staat het project evaluatie wat ging er niet zoals het zou moeten, en wat zou ik kunnen verbeteren. Hoofdstuk 7 staat de verantwoording van de beroepstaken. Hoofdstuk 8 is de begrippenlijst en hoofdstuk 9 staat de bronnenlijst. Aan het einde van dit document staan de bijlages in hoofdstuk 10.

1 Organisatie

inONE is een software bedrijf gevestigd in Leiden. Ze maken eenvoudige e-commerce software voor horecaleveranciers en handige tools voor de Horeca. inONE is een klein en jong bedrijf en bestaat sinds 2004. Ze opereert vanaf 1 locatie in Leiden. Er werken op dit moment 9 medewerkers en 1 stagiair. De afdeling waar dit project wordt uitgevoerd (de ontwikkelafdeling) bestaat uit 3 medewerkers exclusief de stagiair.

Er zijn vier producten ontwikkeld bij inONE te noemen: Leveranciers, Bestelbox, Tools en MenuManager. Bij het in gebruik nemen van het systeem wordt er bij alle klanten een speciale training gegeven zodat zij zonder problemen online kunnen bestellen. Ook bij eventuele latere vragen is er klantenondersteuning, als een specifiek verlengstuk van de service. Alle producten van inONE worden aangeboden d.m.v. het SAAS principe. Hieronder staat beschreven wat de producten inhouden.

Het product: Leveranciers

Met het product leveranciers is het mogelijk om de producten van leveranciers in te laden in de inONE webwinkel. Door middel van een koppeling met het de leveranciers zijn eigen systeem blijft het huidige systeem intact. Hierdoor blijft het eigen systeem van de leverancier intact en dient deze niet aan gepast te worden om een koppeling mee te kunnen maken. De producten die in de inONE webwinkel terecht komen, kunnen besteld worden door de eigen klanten (te noemen horeca bedrijven).

Het product: Bestellen

Met het product bestellen is het mogelijk dat de horecaonderneming via internet en de bestelbox app rechtstreeks bij de vaste leveranciers die zijn aangesloten bij inONE, een bestelling kunnen plaatsen. Bestellen wordt aangeboden door de leverancier, er zijn dus geen kosten aan verbonden voor horeca ondernemers.

Het product: Tools

Wanneer er met Bestelbox wordt gewerkt, kan er gebruik worden gemaakt van inONE Tools. Een pakket van programma's voor de inventarisatie, controle en analyse van de voorraad, inkoop, verkoop, goederenontvangst en facturen.

inONE Tools geeft toegang tot de volgende modules:

- Voorraad: inventarisatie en waarde van de voorraad.
- Goederenontvangst: komen het geleverde en het bestelde overeen?
- Facturen: alle facturen overzichtelijk bij elkaar en gemakkelijk te koppelen aan de grootboeknummers en boekhoudpakket van de horeca ondernemer
- Analyse: maak een analyse op basis van factuurgegevens, voorraad, goederenontvangst en inkoop/verkoop.
- Mail bestellen: handige tool als een van de leveranciers nog niet bij inONE is aangesloten.
- Import en export van bestellingen die in het eigen systeem zijn ingevoerd van de horeca ondernemer.

Het product: Menumanager

Een product voor online receptenbeheer, menu engineering en kostprijsberekeningen. Dit product is op het moment van schrijven nog in ontwikkeling.

1.1 Missie

De missie van inONE is als volgt: “Dat elke horecaonderneming/keuken in Nederland de mogelijkheid heeft om via één punt, Bestelbox, bij zijn eigen leveranciers kan bestellen.”

1.1 Processen

Het primaire proces van inONE is het aanbieden en onderhouden van een software pakket voor leveranciers en zijn klanten (horecaonderneming/keuken). De software moet ervoor zorgen dat de klanten via één punt, bestelbox, zijn bestellingen kan plaatsen.

2 De opdracht

2.1 Probleemstelling

Hoe kunnen de leveranciers die samenwerken met inONE op een afgesproken manier berichten versturen naar specifieke horecabedrijven via zowel het mobiele platform als online versie?

2.1.1 Oude situatie

In de huidige situatie kunnen blog berichten worden verstuurd via Bestelbox. Deze blog berichten kunnen worden gelezen en verstuurd. Er is weinig interactie behalve een link naar een pagina. De berichten kunnen naar specifieke horecabedrijven worden verzonden of naar allemaal. Met de huidige bestelbox app kunnen bestellingen geplaatst worden.

2.1.2 Nieuwe situatie

In de nieuwe situatie moet het mogelijk zijn om berichten te versturen naar gebruikers van bestelbox. Deze berichten moeten verschillende types kunnen zijn. (meer over de types later in dit hoofdstuk). De berichten moeten zowel via het mobiele platform (De inONE Bestelbox app) als via de online versie verstuurd en ontvangen kunnen worden. De berichten moeten naar 1 of meerdere klanten verstuurd kunnen worden. De berichtenservice moet daarom een verbinding hebben met de “groepen indeling” van Bestelbox. Er zal voor zowel de mobiele versie als de online versie 1 en dezelfde programma gemaakt worden. Er moet dus goed gelet worden op de usability dat deze voor beide geschikt is. De berichten die in het huidige mededelingenboard staan zullen niet worden meegenomen. Het huidige systeem zal voor een nog nader te bepalen tijdsduur blijven bestaan voor de oude berichten.

Er kunnen voor de berichtenservice een aantal typen rollen onderscheiden worden:

- Een leverancier kan:
 - Een bericht voor al zijn klanten maken.
 - Een bericht voor 1 of enkele van zijn klanten maken.
- Een horeca manager kan:
 - Een bericht voor alle locaties maken.
 - Een bericht voor 1 of enkele locaties maken.
 - Een privé bericht voor zijn leveranciers maken
- inONE kan:
 - Een bericht voor alle gebruikers maken.
 - Een bericht voor 1 of enkele leveranciers maken.
 - Een bericht voor 1 of enkele horeca locaties maken.

Ook zal er de mogelijkheid zijn om te reageren op berichten die zichtbaar zijn voor de gebruiker om meer communicatie over en weer te krijgen tussen de leverancier en horeca ondernemers.

Er zullen een viertal type berichten komen namelijk primair, sociaal, aanbiedingen en mededelingen. Deze types hebben allemaal een eigen eigenschap. Hieronder in het kort wat de verschillende types inhouden:

Primaire berichten

De primaire berichten zullen door inONE gebruikt worden en zijn de urgente berichten zoals storingen of updates. Het type primair zal altijd getoond worden op het mededelingenboard aan de bovenkant van alle berichten.

Sociale berichten

Berichten moeten via social media gepost kunnen worden. De berichten moeten worden opgepakt en getoond worden aan alleen de klanten van die leverancier. Dit is nog erg onduidelijk hoe het wordt geïmplementeerd omdat er mogelijk te veel berichten binnen komen als een klant meerdere leveranciers hebt en deze vrij veel posten.

Aanbiedingen

In de nieuwe situatie moet het mogelijk zijn om korte reclame berichten sturen ter illustratie: “100kg Zalm in de aanbieding voor €80, -“. De reclame berichten moeten gekoppeld worden aan Bestelbox zodat de korting die in het bericht staat verrekend wordt tijdens het bestellen. Dit zal een aanpassing worden aan het huidige systeem.

Mededelingen

De mededelingen zijn de algemene berichten zoals nieuwsbrieven van de leveranciers, blog berichten van inONE, of andere mededelingen.

Er zijn 2 nieuwe onderdelen gewenst in de berichtenservice. Met het ontwikkelen van het systeem moet hier rekeningen mee worden gehouden. Het betreft het bijhouden van statistieken over de berichten (bijvoorbeeld ...x gelezen) of het aanmaken van custom groepen voor leveranciers. Ook zal er rekening gehouden moeten worden met Windows Mobile voor in de toekomst

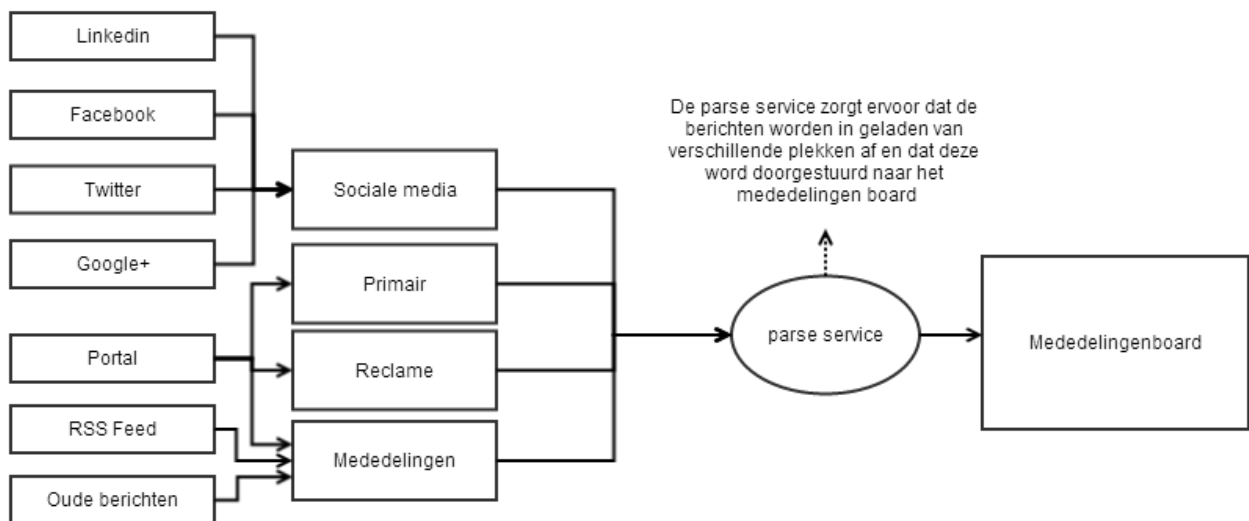
Er zijn een aantal kanalen waar de berichten binnen zullen komen. Dit kan zijn via de portal van inONE via sociale media of een RSS feed. De berichten zullen worden getoond in een soort container. In deze container moet er een mogelijkheid komen om berichten te filteren. Het filteren van berichten is een belangrijk onderdeel van de service. Doordat er veel verschillende kanalen worden gebruikt voor berichten moet er goed gekeken worden hoe voorkomen wordt dat het niet een overvolle service wordt met te veel spam waardoor er ergernis zou kunnen ontstaan. Om het via verschillende kanalen binnen te halen zal gekeken moeten worden of dit zelf geprogrammeerd moet worden of dat er mogelijk een tool bestaat waarbij dat mogelijk is.

De inONE Bestelbox app moet via push berichten kunnen ontvangen en in de statusbalk van het mobiele platform weergegeven worden. Zodra er op de notificatie wordt geklikt moet de app tevoorschijn komen. Als het gaat om een reclame bericht zal in de app zal de bestelling geregeld kunnen worden en zal de korting die in het bericht staat verrekend worden bij de bestelling. Dit onderdeel kan complex worden omdat ieder mobiel platform anders zal werken waardoor er veel uitgezocht zal moeten worden.

.

In Figuur 1 Voorbeeld werking ziet u een korte weergave van hoe de berichtenservice ongeveer in zijn werk zal gaan. De verschillende kanalen (zoals Facebook, Twitter, maar ook B.V. RSS Feeds) zullen als de verschillende soorten types worden opgeslagen waarna ze op het mededelingenboard getoond kunnen worden.

In Figuur 1 Voorbeeld werking staat dat de oude berichten ook mee worden genomen in de mededelingenboard. Het is besloten om de berichten niet mee te nemen omdat dit geen meerwaarde bieden voor het nieuwe systeem. Ook bestond de kans op fouten met het overzetten omdat deze berichten nog in Domino Databanken staan.



Figuur 1 Voorbeeld werking

2.2 Doelstelling

De doelstelling van het project is er om er voor te zorgen dat inONE een werkende berichtenservice krijgt die aan de wensen en eisen van alle stakeholders voldoet. Voor het einde van de afstudeerperiode moet de berichtenservice worden opgeleverd en volledig getest zijn.

2.3 Resultaat

Het resultaat van de afstudeeropdracht is een berichtenservice welke aan de wensen en eisen van alle stakeholders voldoet. De berichtenservice moet in productie kunnen worden genomen. Het moet dus volledig getest zijn. Door de nieuwe berichtenservice zullen de horecaleveranciers door middel van pushberichten reclameberichten kunnen sturen die bestemd zijn voor een specifieke groep horecabedrijven. De berichtenservice zal gaan werken om de producten van de leveranciers te promoten en de communicatie tussen horeca en leveranciers te verbeteren via de berichtenservice te laten lopen.

2.4 Producten

De volgende (tussen)producten zullen worden opgeleverd voor inONE tijdens het ontwikkelen van de berichtenservice:

- **Plan van aanpak**
In het plan van aanpak zal opgenomen worden hoe ik dit project zal uitvoeren. Dit zal in de eerste week van het project opgeleverd worden.
- **Het ontwerp van de berichtenservice**
Aan het begin van het project zal er een technisch ontwerp worden gemaakt. Het ontwerp zal worden gemaakt d.m.v. UML in het programma Gliffy.
- **Deployment diagram**
Het deployment diagram zal gebruikt worden om een beter inzicht te tonen in het complete project.
- **De applicatie**
De applicatie is een onderdeel van het project.
- **Testplan**
Het testplan zal worden opgesteld aan het begin van het project. Elke sprint zal getest worden, dit zal worden opgenomen in het testplan.
- **Testrapportage**
De uitkomsten van de testen zullen in een document worden bijgehouden zodat er voor het hele systeem een testrapportage is.

3 Software ontwikkelmethode

De software ontwikkelmethode die zal worden gebruikt is SCRUM. SCRUM is een framework voor agile management van softwareontwikkeling. Omdat de projecten die worden uitgevoerd bij inONE worden gemanaged met deze ontwikkelmethode is de keuze hierop gevallen. Voor het project zullen daarom de daarbij behorende documenten worden opgeleverd zoals het product back log voor alle functionele en niet functionele eisen, de sprint backlog voor de eisen van de sprint, de sprint burndown chart om de voortgang goed bij te houden.

Elke sprint zal een lengte van 3 weken hebben. Aan het eind van elke sprint zal een werkend deel opgeleverd worden als een prototype. Er zal niet worden gewerkt in teamverband maar door alles goed vast te leggen kan wel een goed overzicht gehouden worden over het werk door de bedrijfsmentor en de opdrachtgever. De ontwikkeling zal alleen door mij worden gedaan maar mochten er vragen zijn of uitleg nodig is hier altijd de mogelijkheid voor bij de bedrijfsmentor en overige programmeurs. Voor elke spint zal aan het begin vastgesteld worden welke onderdelen er behandeld gaan worden. De requirements worden vastgelegd in de product back log in de vorm van user stories. De onderdelen die behandeld zullen gaan worden, worden per sprint vastgelegd in de sprint backlog

Als ondersteuning van SCRUM wordt het SAAS pakket JIRA gebruikt. JIRA is een project management pakket. Alle documentatie en gemaakte beslissingen zullen hier in opgenomen worden. Ook bevat JIRA de module confluence in om modellen en diagrammen te maken. Omdat dit de standaard is bij inONE en alle systeem documentatie bijgehouden heb ik ook JIRA gebruikt om mijn modellen en diagrammen in confluence te maken.

3.1 Afwijking van SCRUM

In dit hoofdstuk staan de afwijkingen beschreven die ik heb gedaan tijdens mijn afstuderen ten opzichte van SCRUM beschreven.

Tijdens mijn afstuderen ben ik begonnen met de documentatie maken van mijn afstuderen. Dit omdat dit de wens was van de product owner om te zien welke functionaliteiten verwerkt moesten worden. SCRUM zegt dat de documentatie tijdens het project wordt bijgewerkt om meer flexibel te kunnen werken aan projecten zonder dat er bepaalde functionaliteiten vast staan. Dit heb ik bij de opdrachtgever aangegeven. Omdat de opdrachtgever niet goed wist wat hij precies wilde hebben, heb ik besloten om alle functionaliteiten aan het begin van het project op te leveren zodat de opdrachtgever weet wat de mogelijkheden zijn

Ik heb dus in 2 “fases” gewerkt namelijk de oriënterende (ontwerp) fase (Fase 1) en de fases van de sprints (Fase 2).Tijdens de eerste fase heb ik al het oriënterend werk verricht zoals het inlezen in XPages zodat ik wist hoe het werkte en heb ik mijn plan van aanpak opgezet. Ook heb ik in deze fase product back log ontwerpen gemaakt en de requirements duidelijker gemaakt. In fase 2 heb ik de sprints uitgevoerd. In de sprints ben ik het product backlog afgegaan en heb ik de backlog items ontwikkeld en getest. Het testen heb ik gedaan door middel van junit testen en een black box interface test. Meer over de 2 fases kunt u lezen in hoofdstuk 4 voor fase 1 en in hoofdstuk 5 staat fase 2.

Tussendoor zijn bepaalde functionele aspecten wel veranderd wanneer we beter inzicht hadden in de mogelijkheden of de functionaliteit op deze manier wel werkte. Over het algemeen zijn de functionele aspecten niet veel veranderd vaak ging dit meer over een klein onderdeel. Per sprint heb ik wel goed bekeken

De functionele aspecten die zijn veranderd zijn bijvoorbeeld het uploaden via AWS s3 i.p.v. in de database de bestanden opslaan of buttons die een checkbox zijn geworden en dit soort zaken.

4 Oriënterende fase (Fase 1)

In de eerste fase 1 de berichtenservice heb ik vooral oriënterend werk gedaan zoals uitzoeken hoe xPages werken of hoe bepaalde functies werken. In deze fase heb ik het plan van aanpak (PvA) gemaakt. Het PvA is te vinden in Bijlage 2. In het PvA staat de aanpak die ik voor ogen had om te gebruiken voor dit project zodat het project kan slagen.

In het PvA heb ik bij hoofdstuk 2.1.2 Nieuwe situatie uitleg gegeven over hoe de berichtenservice ongeveer eruit zal moeten gaan zien. Aan de hand van deze omschrijving ben ik begonnen met het maken van de product backlog (zie Bijlage 3 voor de eerste versie van de product backlog) hierin stonden de globale functies. En was nog niet goed duidelijk hoe het vorm moest krijgen.

Na het afronden van de backlog ben ik de Mockups gaan maken aan de hand van de backlog. Ik kwam er al gauw achter dat er veel details maar ook globalere functies misten. In het gesprek van 21-8-2013 (De notulen zijn te vinden in Bijlage 4) werd duidelijk wat de berichtenservice moet gaan worden en wat de berichten service is. In een korte zin is het volgende: “Puur voor interne communicatie tussen leveranciers en zijn klanten, oftewel een Interne berichten service. “

4.1 Requirements

De requirements heb ik opgezet aan de kennis welke opgedaan is tijdens mijn inwerk periode en aan de hand van gesprekken. Ook tijdens het maken van de Mockups ben ik achter requirements gekomen die handig zoude kunnen zijn voor het project. Deze functionaliteiten heb ik in Jira gezet en regelmatig met de opdrachtgever overleg gehad. In Bijlage 3 staat de versie van de product backlog die in de eerste 3 weken is gemaakt. Gaandeweg het project zijn er meerdere punten bijgekomen. Hieronder staat de volledige lijst met story points. In de laatste kolom staat of de story point is afgerond.

Key	Samenvatting	Priority	Story Points	Af?
BER-5	Als inONE wil ik het type Primair gebruiken om belangrijke berichten te versturen zoals storingen of updates.	Critical	32	Ja
BER-4	Als een Stakeholder wil ik een viertal types in kunnen stellen: Primair, Sociaal, Aanbiedingen en Mededelingen	Critical	64	Ja
BER-3	Als inONE wil ik een bericht kunnen sturen naar 1 of meerdere leveranciers/horecamanagers en horeca	Critical	32	Ja
BER-1	Als een leverancier wil ik een bericht kunnen sturen naar 1 of meer klanten.	Critical	32	Ja
BER-26	wysiwyg editor uitzoeken en implementeren	Major	32	Ja
BER-25	Als een Stakeholder wil ik mijn eigen berichten kunnen inzien	Critical	64	Ja
BER-10	Als een Stakeholder wil ik berichten via een online versie of een mobiele versie kunnen versturen en ontvangen.	Critical	32	Ja
BER-7	Als een Stakeholder wil ik een filter mogelijkheid hebben voor de berichten zodat ik niet wordt overladen met berichten.	Critical	64	Ja
BER-12	Als een Stakeholder wil ik alle berichten in een "container" kunnen zien.	Critical	32	Ja
BER-34	Als een stakeholder wil ik alleen mijn eigen berichten laten zien.	Major	64	Ja
BER-29	Als een stakeholder wil ik validatie hebben over de ingevulde gegevens	Critical	64	Ja
BER-32	Als een stakeholder wil ik via een type ahead een product kunnen zoeken in een nieuw bericht	Critical	128	Ja
BER-2	Als een horecamanager wil ik een bericht kunnen sturen naar 1 of meer locaties.	Critical	32	Ja
BER-9	Als een Stakeholder wil ik instellingen voor filters eenmalig in hoeven stellen, maar ook later kunnen wijzigen.	Critical	64	Ja
BER-37	Als een stakeholder wil ik pagination hebben op de portal.	Critical	32	Ja
BER-36	De afbeeldingen bij de producten moeten bij een aanbieding opgehaald worden. Dit ophalen uit notes database.	Blocker	32	Ja
BER-38	BER-8 → Android notificaties maken	Critical	128	Ja

BER-39	BER-8 → IOS notificatie maken.	Major	128	Nee
BER-8	Als een bestelbox app gebruiker wil ik notificaties krijgen van belangrijke berichten op mijn mobiel apparaat en in de statusbalk van het mobiele platform weergegeven worden.	Critical	128	Alleen android
BER-35	Als een stakeholder wil ik mijn eigen berichten kunnen aanpassen.	Critical	128	Nee
BER-33	Java code aanpassen zodat deze verwijst naar de inone Dtb	Trivial	64	Nee
BER-31	Radio button types bij nieuw bericht gebruik laten maken van clientside script om divs te hiden ipv via serverside script.	Minor	32	Nee
BER-30	Als een stakeholder wil ik groepen aan kunnen maken en hiermee naar een groep een bericht kunnen sturen	Trivial	64	Nee
BER-28	De ccPicklist wordt met grote aantallen onoverzichtelijk. Dit uitbreiden met een zoekveld/srollbar.	Minor	32	Nee
BER-24	Als een Stakeholder wil ik persoonlijke berichten hebben	Minor	132	Nee
BER-23	Als een Stakeholder wil ik persoonlijke berichten kunnen bekijken.	Minor	132	Nee
BER-21	Als een Stakeholder wil ik dat er de mogelijkheid is voor meertaligheid.	Trivial	132	Nee
BER-20	Als een Stakeholder wil ik een status van een persoonlijk bericht in kunnen stellen.	Minor	32	Nee
BER-19	Als een Stakeholder wil ik kunnen reageren op berichten	Major	64	Nee
BER-18	Als een Stakeholder wil ik statistieken inzien van mijn berichten zoals bijvoorbeeld hoe vaak een bericht gelezen is..	Minor	64	Nee
BER-17	Als een Stakeholder wil ik een email notificatie krijgen wanneer ik een persoonlijk bericht ontvangen heb.	Major	32	Nee
BER-16	Als een Stakeholder wil ik persoonlijke berichten kunnen sturen. Dit zijn simpele tekstuele berichten.	Major	132	Nee
BER-14	Als een Stakeholder wil ik wanneer het bestelscherm opent gelijk de aanbieding naar voren komt als ik op een aanbieding geklikt heb.	Minor	64	Nee
BER-13	Als een Stakeholder wil ik kunnen klikken op een notificatie en op die manier de app openen met het bericht. Bij een reclamebericht wil ik dat het bestelscherm opent.	Major	132	Nee
BER-11	Als een Stakeholder wil ik berichten ontvangen via verschillende kanalen zoals social media, een RSS feed of via de inONE portal.	Major	128	Nee
BER-6	Als een leverancier wil ik mijn sociale media berichten automatisch laten doorsturen naar de berichtenservice.	Major	128	Nee

4.2 Mockups:

In dit hoofdstuk staan mijn ontwerpen beschreven met de daarbij horende uitleg. Het project ben ik begonnen met het ontwerpen van het systeem documentatie. Ik ben als eerste begonnen met het maken van Mockups aan de hand van de Mockups heb ik een klassediagram gemaakt. Nadat de Mockups overeen kwamen met het diagram en volledig waren heb ik de SQL DDL gemaakt zodat de database gelijk volledig ingericht kon worden.

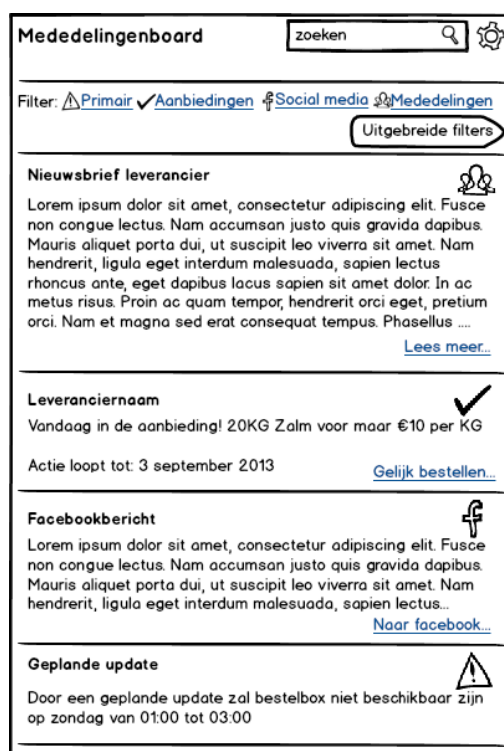
4.2.1 Mededelingenboard

Het mededelingenboard zal onderdeel worden van de portal. en is daarom een smalle balk zodat deze in de portal geïmplementeerd kan worden. In deze balk komt een overzicht van de meest recente berichten.

De berichten zullen op de volgende wijze worden weergegeven:

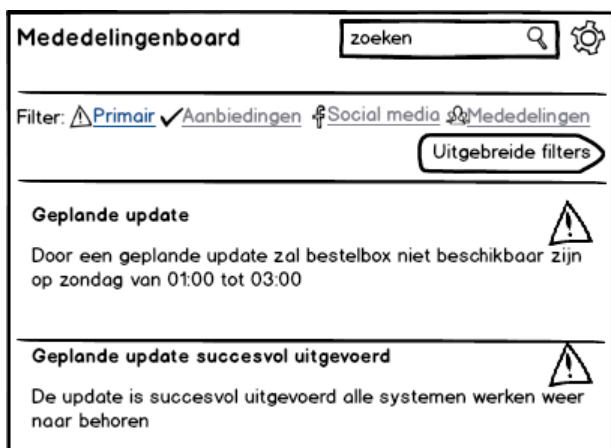
- Primair: Bovenaan. Deze worden altijd weergegeven omdat dit belangrijke berichten zijn zoals geplande updates, storingen etc.
- Aanbiedingen. Hier zullen de meest recente aanbiedingen worden weergegeven welk zijn ingevoerd in de berichtenservice.
- Social Media berichten - De social media berichten zullen als een scroller worden weergegeven zodat er meerdere berichten op 1 plek kunnen worden weergegeven
- Mededelingen - Ook de mededelingen worden als een scroller weergegeven.

Hoeveel berichten worden weergegeven moet nog bekeken worden hoeveel ruimte de berichten service in gaat nemen met een aantal berichten.



Figuur 2 Mockup - Mededelingenboard

4.2.2 Filteren



Figuur 3 Mockup - Filteren

De berichten kunnen op een simpele wijze worden gefilterd door op primair, Aanbiedingen, Social media, of mededelingen geklikt wordt. Door op een van deze knoppen te drukken zullen de berichten worden verborgen/weergegeven. Of een bericht wordt verborgen of weergegeven hangt af van de status van de knop. Standaard staan de filters uit zodat alle berichten worden weergegeven.

Voor elke bericht is het ook mogelijk om een titel en een omschrijving in te voeren. De beschrijving is alleen niet altijd mogelijk om te gebruiken bij mededeling, meer hierover bij het hoofdstuk van de mededelingen

Primair

Reclame

Mededeling

Titel

B I U ↵ style ▾ : i ≡ | ↶ ↷ | 🖼️ 😊

Versturen

Figuur 4 Mockup - Bericht invoeren

4.4 PersonalMessage

4.4.1 Statusbalk

Bovenin in de header van de applicatie zal een menu knopje komen om de persoonlijke berichten te kunnen bekijken of openen in een kort overzicht van de laatste 4 berichten. Zodra je op een bericht klikt, opent zich een uitgebreider scherm waarmee je toegang krijgt tot het versturen van berichten.



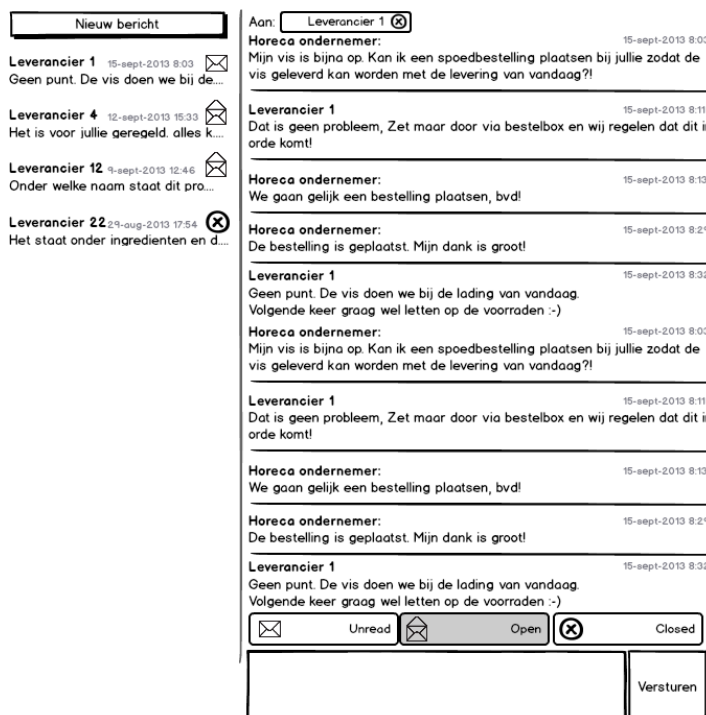
Figuur 5 Mockup - Persoonlijk bericht

4.4.2 Berichten overzicht

Door onderin een bericht op te stellen en te drukken op versturen wordt het mogelijk om met een van de eigen contacten (Lees leveranciers/horeca locaties) te chatten via de applicatie. Ook kan er een status worden bijgehouden worden van het bericht. Wanneer een bericht geopend wordt verandert de status automatisch naar "open" er kan ook gekozen worden om een bericht weer op "ongelezen" te zetten of op "gesloten". Zodra het bericht op "gesloten" staat zal het niet meer mogelijk zijn om berichten via de chatsessie te versturen. Om berichten naar de leverancier te versturen zal een nieuw bericht aangemaakt moeten worden om op deze manier een nieuw bericht te versturen. Aan de linkerkant van het

scherm staan alle chatsessies welke zijn gestart met daarbij een icoon van de status.

Aan de rechterkant staat de chatsessie. Boven het invulveld is de mogelijkheid om de status te veranderen van de chatsessie.



Figuur 6 Mockup - Overzicht persoonlijk bericht

4.5 Klassediagram en daarbij gemaakte keuzes

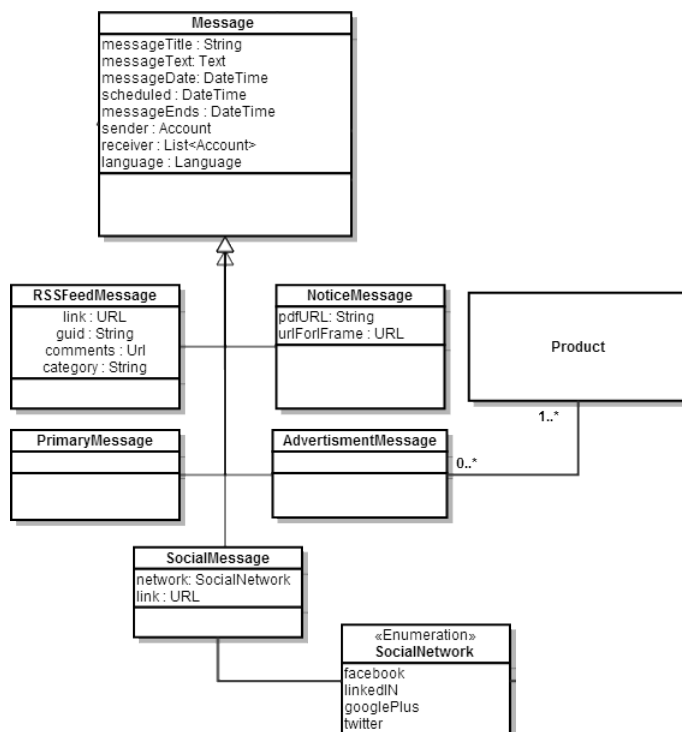
In dit hoofdstuk staat de uitleg en de gemaakte keuzes over de klassediagram welke ik heb gemaakt voor de berichtenservice. Omdat de wens van de opdrachtgever was om alle functionaliteiten uit te werken is het klassediagram compleet en zijn alle attributen uitgewerkt. Ik heb om een duidelijker beeld te krijgen onderdelen los van elkaar gehaald. Het volledige klassediagram is te vinden in Bijlage 5. In dit hoofdstuk worden de namen gebruikt van de klassen welke zijn terug te vinden in de diagram.

In de eerste versie van het klassediagram stonden de hoofdlijnen van het diagram voor 60% a 70% goed. In versie 1 staat bijvoorbeeld wel al de specialisatie van berichten. Alleen missen deze specialisaties nog een aantal attributen omdat dit later in het project naar voren is gekomen. In versie 2 is het abstract factory pattern toegevoegd. Bij de eerste 2 versie dacht ik dat een Supplier en een Account 2 Klassen die nodig waren voor de koppeling met een gebruiker. Later is het naar voren gekomen dat een Account of een Supplier is of een Buyer (Supplier is een leverancier en een Buyer is een horeca onderneming). A versie 4 zijn er geen grote veranderingen meer geweest behalve dat er een aantal attributen zijn gewijzigd zoals bijvoorbeeld bij SocialConnection heeft een extra attribuut defaultPublish gekregen. Deze instelling word gebruikt of deze connectie standaard gebruikt moet worden om een bericht te versturen naar een sociaal platform

4.5.1 Specialisatie van berichten

Elke type heeft zijn eigen klassen als specialisatie omdat ze elk van elkaar verschillen in attributen en in werking (Zie Figuur 7 Specialisatie). De berichten kunnen verschillende types bevatten. Namelijk:

- PrimaryMessage (inONE bericht)
 - Staat altijd bovenaan op het mededelingenboard en is alleen bedoeld voor berichten van inONE
- NoticeMessage (Mededeling)
 - Een mededeling heeft de mogelijkheid om een PDF of een URL in te voeren voor in een IFrame. Een requirement was om deze functionaliteiten ook in de nieuwe situatie te gebruiken.
- AdvertisementMessage (Advertentie)
 - Een AdvertisementMessage heeft een relatie met Product zodat een advertentie specifiek voor 1 of meerdere artikelen is.
- SocialMessage (Sociale berichten)
 - Deze berichten komen van de sociale media af zoals Facebook, Twitter en LinkedIn Deze heeft een Enumeratie van SocialNetwork.



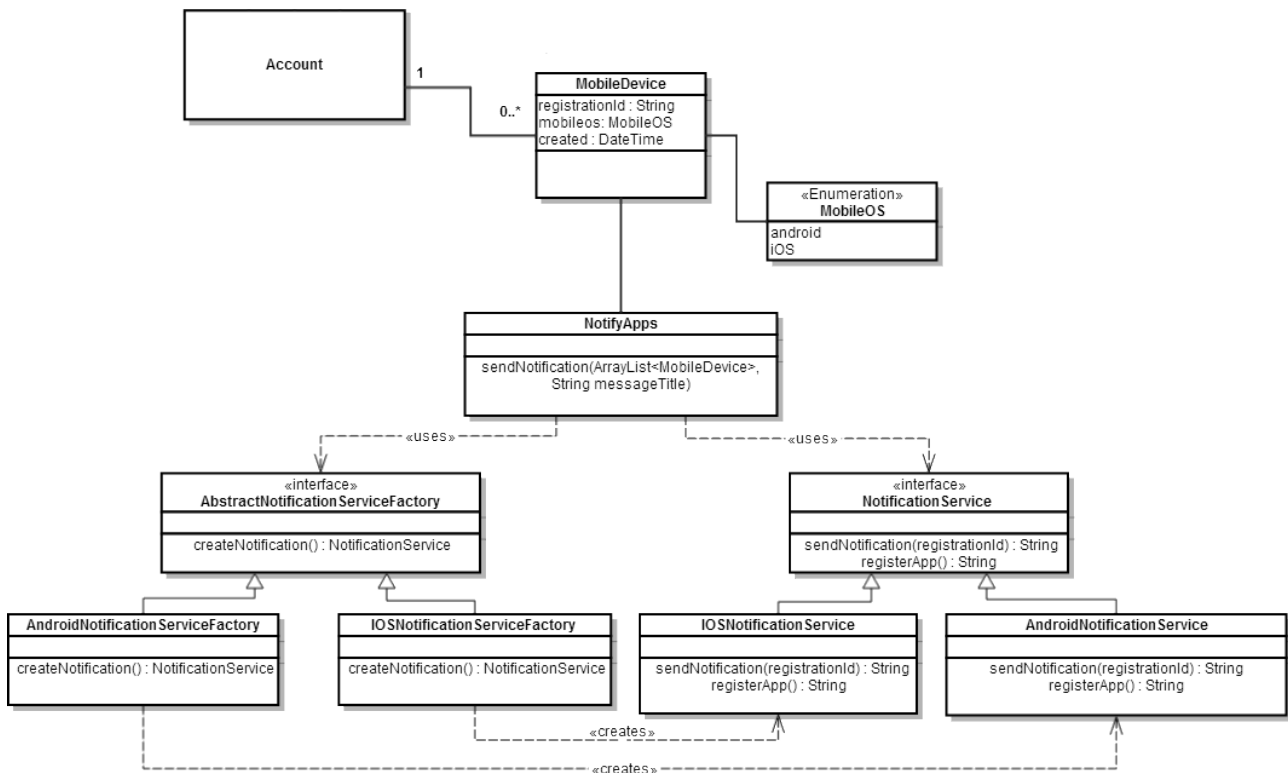
Figuur 7 Specialisatie

- RSSFeedMessage (Voor RSSFeeds)
 - Deze worden met een polling systeem opgehaald uit de verschillende RSS feeds (hoe dit in zijn werk gaat is in het project niet gerealiseerd)

PrimaryMessage en AdvertisementMessage hebben geen attributen. Wel zijn dit specialisaties van Message. Het is niet gebruikelijk om een specialisatie te hebben zonder attributen. Ik heb er toch voor gekozen om de specialisatie toe te voegen zodat er in de toekomst aan deze klassen attributen toegevoegd kunnen worden als dit gewenst is.

4.5.2 Notificaties

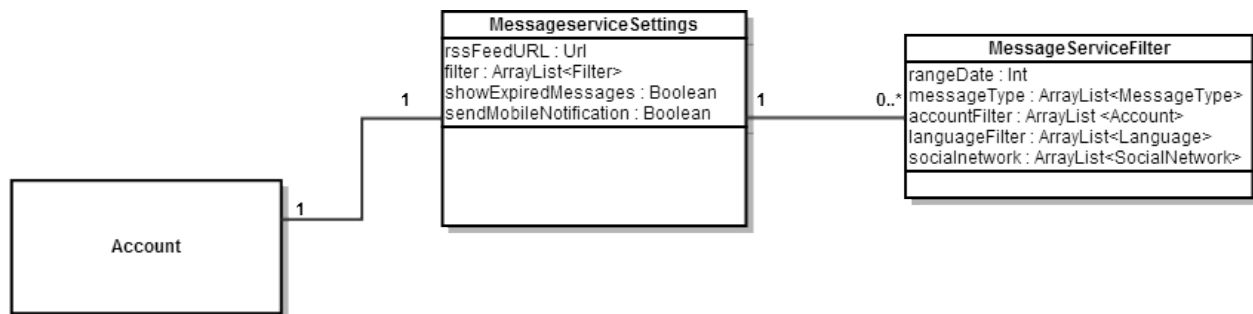
Om de notificaties voor de verschillende mobiele devices te laten werken heb ik gebruik gemaakt van het abstract factory pattern (zie Figuur 8 Abstract factory). Dit pattern zorgt ervoor dat in de code geen rekening dient te houden met de implementatie van de notificaties. Beide hebben dezelfde methodes alleen deze methodes werken verschillend van elkaar omdat elk mobiele platform zijn eigen manier heeft van het versturen van notificaties. Tijdens runtime moet er worden aangegeven om welk mobiel platform het gaat zodat er een notificatie verstuurd kan worden naar de mobiele devices. Als er meerdere mobiele platformen toegevoegd moeten worden kan dit eenvoudig door de classes aan de abstract factory pattern toe te voegen zonder dat de huidige code aangepast moet worden. Hierdoor zijn de notificaties modulair en werken deze onafhankelijk van elkaar.



Figuur 8 Abstract factory

4.5.3 Instellingen

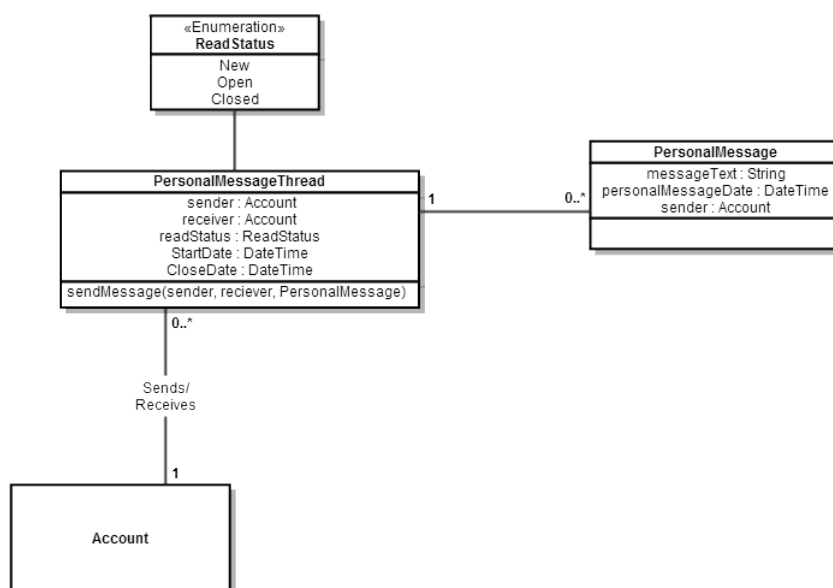
MessageServiceSettings heeft een 1 op 1 relatie met Account. Ik heb er voor gekozen om dit niet in account op te slaan omdat ik de berichtenservice als een losse module wil hebben zodat geen bestaande onderdelen dienen te worden aangepast. Dit vergroot de onderhoudbaarheid van het huidige systeem en de berichtenservice omdat de onderdelen volledig los van elkaar werken en dus los van elkaar kunnen worden aangepast en/of uitgebreid (Zie Figuur 9 Instellingen).



Figuur 9 Instellingen

4.5.4 Persoonlijk bericht

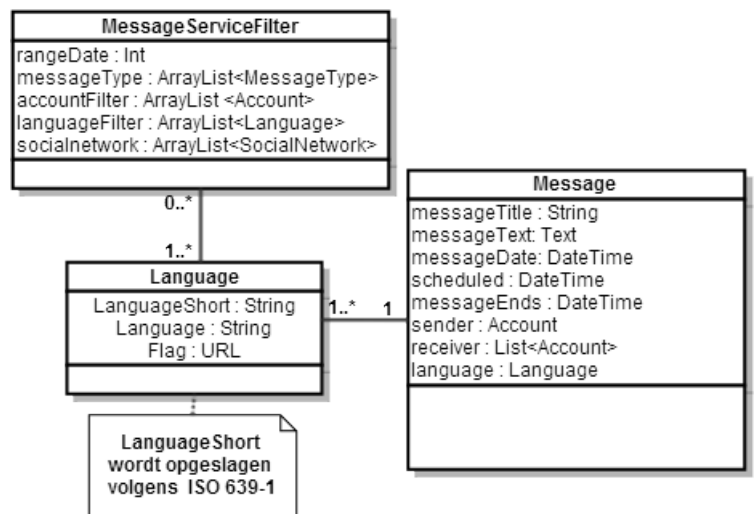
Een requirement van de stakeholders was de mogelijkheid om een persoonlijk bericht te kunnen sturen. Dit heb ik opgelost door 2 klassen toe te voegen namelijk **PersonalMessageThread** en **PersonalMessage**. (zie Figuur 10 Persoonlijk bericht). 1 thread kan meerdere berichten bevatten. Een thread kan worden gezien als een forumthread met meerdere berichten welke de status “nieuw”, “open” of “gesloten” kan hebben. Het enige verschil is dat er in plaats van meerdere gebruikers het een 1 op 1 gesprek is. Ik heb gekozen voor threads zodat er per gesprek een status bijgehouden kan worden. Zodra een vraag of probleem opgelost is, kan de thread worden gesloten. Mocht er bijvoorbeeld een nieuwe vraag zijn dan kan er een nieuwe thread geopend worden. Zodra een thread is afgesloten kan deze niet meer aangepast of aangevuld worden.



Figuur 10 Persoonlijk bericht

4.5.5 Taalondersteuning

Om de verschillende talen te ondersteunen heb ik een Language tabel aangemaakt (Zie Figuur 11 Talen). De korte benamingen van de talen wordt onder de ISO 639-1 opgeslagen. Deze ISO norm slaat talen op als variabelen met 2 karakters. Ik heb gekeken naar de mogelijkheden van 639-2, deze norm slaat talen op in variabelen met 3 karakters. En is meer gericht op het bijvoorbeeld de verschillende soorten Duitse taal zoals Duits Oud hoog of Duits Midden hoog en Zwitsers Duits. In overleg met een ontwikkelaar van inONE heb ik gekozen voor de 639-1 omdat dit genoeg detail bevat voor de berichtenservice en de verschillende “dialecten” nooit voor zullen komen.



Figuur 11 Talen

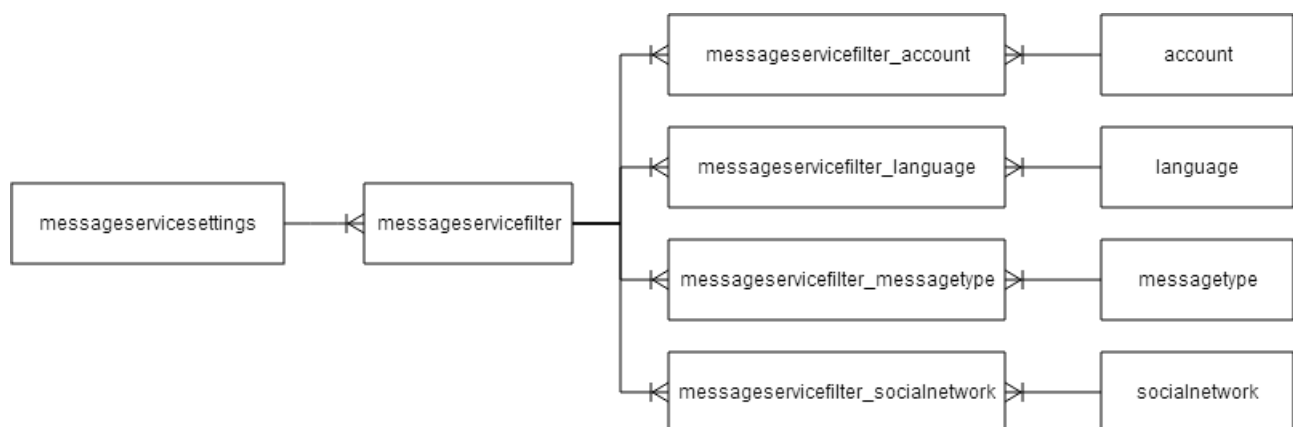
4.6 MySQL + Keuzes

Een requirement van de stakeholder was om de opslag van data te doen in MySQL omdat alle data bij inONE in MySQL word opgeslagen. Voor de MySQL heb ik een aantal keuzes moeten maken om het klassediagram goed te laten overeenkomen met de MySQL.

Voor de specialisatie heb ik gekozen om het type bij te houden in Message. Dit is een foreign key (FK) met de tabel messageType. Op deze manier is de FK altijd de goede waarde. Als er een type bij moet komen moet er dus in messageType een extra regel worden aangemaakt in de database. De variabele waarden van de verschillende types worden bijgehouden in aparte tabellen. Elk type krijgt dus zijn eigen tabel. In het kort houdt dit dus het volgende in: Het type wordt opgeslagen in Message onder MessageTypeeld hieraan kan je herleiden welke tabel er benodigd is om de extra variabelen op te halen voor het type.

Om de status bij te houden van een bericht heb ik ervoor gekozen om bij de relatie van een ontvanger in de tussentabel de column readStatus toe te voegen welke dezelfde enumeratie gebruikt als deze van PersonalMessage zoals staat beschreven in hoofdstuk 4.5.4. Bij een persoonlijk bericht houd ik de status bij in de PersonalMessageThread. Een Thread heeft een status. Zodra deze gesloten is, kan deze niet meer aangepast worden en zal er een nieuwe thread aangemaakt moeten worden. Onder de thread hangen meerdere PersonalMessages om er zo voor te zorgen dat er een status bijgehouden kan worden van een bepaald gesprek en niet de hele conversatie tussen de verschillende partijen.

Het filteren van berichten hangt aan MessageserviceSettings een tabel MessageServiceFilter. In deze tabel worden verschillende types bijgehouden zoals van/tot datum, MessageType, account filter, taal filter en SocialNetwork filter. De laatste vier worden bijgehouden in een tussentabel omdat deze vaker per filter voor kunnen komen. Het diagram in Figuur 12 MessageServiceSettings - MessageServiceFilter is hoe de structuur in MySQL is.



Figuur 12 MessageServiceSettings - MessageServiceFilter

De berichtenservice SQL DDL is te vinden in Bijlage 6

4.7 XPages

In de eerste week van mijn stage heb ik XPages geleerd. Dit heb ik gedaan door middel van het ontwikkelen van een RSS feed lezer. Dit was ook in het oogpunt van mijn opdracht zodat ik dit deel zou kunnen hergebruiken in mijn project. Ik heb het uitlezen van gegevens en het ophalen ervan gebruikt in mijn project. Ook hoe ik tekst vanuit Java kon weergeven in de view.

Een XPage is in principe een JavaServer Faces (JSF) pagina welke verder is uitgebreid welke een HTML pagina genereert. Zo is het altijd mogelijk om HTML Tags te gebruiken in een XPage, hiermee heb je alle HTML tags tot je beschikking. Er zijn voor XPages ook eigen controls, tags en elementen zodat er op een verbeterde/ gemakkelijke manier serverside javascript gecombineerd kan worden met cliënt side javascript.

5 De sprints (Fase 2)

In fase 2 van het project ben ik begonnen met het bouwen van de berichtenservice. Omdat alle modellen en Mockups klaar waren was er minder denkwerk nodig voor het programmeren. Uiteraard is niet alles precies zo geworden als dat ik als eerste instantie bedacht had omdat dit niet goed zou werken.

De volgende onderdelen zijn veranderd:

- De mogelijkheid voor het opzoeken van oude berichten is toegevoegd.
- In plaats van knoppen voor het filteren worden er checkboxes gebruikt.
- Er worden plaatjes weergegeven bij berichten.

Dit zijn geen grote veranderingen maar vooral details. Meet over de onderdelen die zijn veranderd staan beschreven in de sprints zelf. Ook staat beschreven welke problemen ik ben tegengekomen, hoe ik eraan ben gekomen en hoe ik deze problemen heb opgelost. Dit kunnen zowel functionele als technische problemen zijn. De keuzes die ik heb moeten maken tijdens de sprints staan ook beschreven bij de desbetreffende sprint.

Aan het einde van elke sprint heb ik testen uitgevoerd. Zo heb ik voor elke sprint junit testen gemaakt voor de white box testen. Ook heb ik een black box interface test gedaan. In de black box test heb ik alle stappen doorlopen van het eind gebruiker. Deze stappen zijn uitgewerkt in Bijlage 9

Alle code voorbeelden die in dit hoofdstuk zijn gebruikt zijn onderdeel van de berichtenservice. Sommige stukken code zijn verkleind/verkort voor de leesbaarheid van de voorbeelden.

5.1 Sprint 1

In sprint 1 heb ik er voor gezorgd dat berichten in de database komen te staan. Omdat dit mijn eerste sprint was en ik nog niet veel ervaring had met XPages was dit voor mij vooral ook het echte leren van XPages. Ik had al in de voorbereiding een RSS feed gemaakt. Alleen hiermee werd geen MySQL database benaderd. Na sprint 1 wist ik goed hoe XPages werkte en hoe ik gegevens moest wegschrijven in de database. Omdat het uitlezen ongeveer hetzelfde ging heb ik dit gelijk in mijn hele project kunnen gebruiken.

In sprint 1 heb ik ook veel ontdekt van de werking van XPages. Daarom staat in deze sprint uitgelegd hoe dingen zijn opgelost en hoe XPages samen werken met Java.

Hieronder een overzicht van de tickets welke ik heb opgelost tijdens deze sprint:

Key	Summary	Priority	Story Points
BER-5	Als inONE wil ik het MessageType Primair gebruiken om belangrijke berichten te versturen zoals storingen of updates.	Critical	32
BER-4	Als een Stakeholder wil ik een drietal types in kunnen stellen: Primair, Aanbiedingen en Mededelingen	Critical	64
BER-3	Als inONE wil ik een bericht kunnen sturen naar 1 of meerdere leveranciers/horecamanagers en horeca	Critical	32
BER-1	Als een leverancier wil ik een bericht kunnen sturen naar 1 of meer klanten.	Critical	32

Voordat ik daadwerkelijk ben begonnen met het bouwen van het project heb ik nog een aantal dingen onderzocht. Zo heb ik besloten om via het MVC principe mijn project op te zetten. En heb ik korte uitleg gehad over hoe ik de database kon benaderen.

5.1.1 MVC

Ik heb het project opgezet door middel van het Model View Controller (MVC) principe. Door het MVC principe te gebruiken is de onderhoudbaarheid vergroot. Ook wordt hierdoor de testbaarheid verhoogd. Dit komt omdat de code zo veel mogelijk gescheiden is van elkaar. Zo zorgt het model voor de representatie van het datamodel. De view is de interface en hier worden geen berekeningen in gedaan. En de controller doet de afhandeling van alle gegevens en schrijft de gegevens naar de database. Hierdoor kunnen views toegevoegd worden zonder het model aan te hoeven passen.

Ik heb het MVC op de volgende manier toegepast:

- **Model** – Deze klassen zijn altijd enkelvoudig zoals Message of Setting.
- **View** – Dit zijn alle interfaces (XSP pagina's)
- **Controller** – Bijna alle modellen hebben een eigen controller dit is de meervoudige naam van het model klasse zoals de controller van Message is Messages, en van Setting is de controller klasse Settings.

Uitzonderingen zijn de specialisaties van Message zoals PrimaryMessage NoticeMessage etc. Omdat de Messages controller alle modellen van Message en zijn specialisaties afhandelt.

Het MVC principe heb ik door het hele project toegepast.

5.1.2 My SQL Werking in XPages

Omdat het voor mij nog niet duidelijk was hoe ik met XPages i.c.m. Java een database moest benaderen heb ik een aantal voorbeelden gekregen om mee te werken die al eerder bij inONE zijn ontwikkeld. In deze klassen kwam ik er achter dat het niet moeilijk is om een database te benaderen omdat alle klassen reeds aanwezig zijn. Hieronder een overzicht met de klassen welke ik nodig heb gehad om de database te benaderen:

- MySQLDatasource.java
 - o Deze klasse wordt gebruikt om een connection pool aan te maken door middel van het object pool pattern Elke verbinding heeft zijn eigen pool. Zo wordt ervoor gezorgd dat er meerdere verbindingen mogelijk zijn in dezelfde applicatie. In deze Klasse staan tevens ook de login gegevens van de My SQL database om een verbinding te kunnen leggen.
- SQLQuery.java
 - o SQL Query handelt alle My SQL query's af zo is het mogelijk om met de methode executeSelect een select statement uit te voeren. Of met de methode executeInsert een insert te doen. Deze methode geeft de ID terug van het item wat in de database is gezet

Hieronder een voorbeeld hoe deze klassen met elkaar samenwerken:

```
String query = "SELECT * FROM message WHERE messageid = ?";
try {
    PreparedStatement pst = SQLQuery.getDbConn().prepareStatement(query);
    pst.setInt(1, messageId);
    ResultSet rs = pst.executeSelect();

    while (rs.next()) {
        new Message(rs);
    }
} catch (SQLException e) {
    Debug.consoleWriteln(e.getLocalizedMessage());
    Debug.consoleWriteln(e);
} finally {
    SQLQuery.close(rs);
    SQLQuery.close(pst);
}
return new Message(rs);
}
```

SQLQuery.getDBCon zorgt ervoor dat er een connectie met de database is. Ook wordt in de connectie gelijk het statement neergezet zodat deze uitgevoerd kan worden. Vervolgens worden de waarden van de parameters gevuld doormiddel van pst.SetInt, hier kunnen alle datatypes welke beschikbaar zijn in Java gevuld worden. Door pst.ExecuteSelect uit te voeren wordt de database bevraagd en er wordt een resultset

terug gegeven met alle gevonden data. Deze methode heb ik in alle klassen toegepast wanneer ik de database moest benaderen.

5.1.3 Het bouwen

Zodra ik wist hoe XPages werkte en ik wist hoe ik de database kon benaderen ben ik daadwerkelijk begonnen met het bouwen van de berichtenservice.

5.1.3.1 Model

Als eerste stap heb ik alle benodigde Model Klassen aangemaakt in Domino Designer de ontwikkelclient waarmee de applicatie is gemaakt. Zodat ik de controllers voor de modellen kon aanmaken. Het betreft de volgende klassen

- Message
 - En zijn specialisaties:
 - PrimaryMessage
 - NoticeMessage
 - AdvertisementMessage
 - SocialMessage
 - RSSFeedmessage
- MessageType (enum)

MessageType is een enumeratie met alle mogelijk bericht types. Omdat in JAVA normaal een enumeratie alleen Namen zonder getallen kan bevatten en ik getallen wilde gebruiken met de namen van de specialisatie zodat het duidelijker is om welke specialisatie het gaat, heb ik een aantal methodes toe moeten voegen om ervoor te zorgen dat ik de namen van de specialisatie aan getallen kan toekennen. Ik heb een constructor toegevoegd welke een Value mee gegeven kan worden en een methode getValue om deze waarde weer op te kunnen halen. Voor elke type heb ik de constructor aangeroepen met daarin de waarde welke ik eraan wilde toekennen. Hieronder staat de uiteindelijke enumeratie klasse MessageType:

```
public enum MessageType {
    PRIMARYMESSAGE(1), ADVERTISEMENTMESSAGE(2), NOTICEMESSAGE(3), SOCIALMESSAGE(4),
    RSSFEEDMESSAGE(5), GHOSTMESSAGE(6);
    private final int value;

    private MessageType (int value) {
        this.value = value;
    }

    public int getValue() {
        return value;
    }
}
```

5.1.3.2 Controllers

Nadat de model Klassen volgens het MVC principe waren aangemaakt ben ik begonnen met de controller klasse dit was voor dit onderdeel alleen de controller voor Message de controller wordt dus Messages. Ik heb besloten om 1 controller te maken voor alle messageTypes omdat de specialisaties bij een bericht horen en dus op hetzelfde moment opgehaald moeten worden uit de database.

Met de Messages controller ben ik begonnen met het maken van het eerste createState welke ervoor zorgde dat een bericht in de database komt zonder specialisatie (createMessage(Message message, int messageType). De methode kon in eerste instantie alleen een insert doen voor een Message zonder ontvanger. Nadat de methode werkte heb ik er ook voor gezorgd dat de ontvangers van een bericht in de database komen. Dit heb ik gedaan om door een lijst van ontvangers te lopen doormiddel van een FOR loop zodat de berichten allemaal in de database terecht komen. Nadat een bericht en de ontvangers gerealiseerd waren, ben ik begonnen met het maken van de specialisaties. Ik heb een methode aangemaakt welke het type van de specialisatie terug geeft. Als parameter moet het bericht als de specialisatie type mee worden gegeven. Dit houdt dus in dat ik 5 methodes heb aangemaakt namelijk:

```
- public static PrimaryMessage createMessage(PrimaryMessage message){}
- public static NoticeMessage createMessage(NoticeMessage message){}
- public static RssFeedMessage createMessage(RssFeedMessage message){}
- public static SocialMessage createMessage(SocialMessage message){}
- public static AdvertisementMessage createMessage(AdvertisementMessage message){}
```

In alle methodes heb ik de methode (createMessage) aangeroepen met een cast van het type bericht die ik eerder gemaakt heb om alle overeenkomstige variabelen in de database te zetten inclusief ontvangers.

```
message = (PrimaryMessage) createMessage(message, MessageType.PRIMARYMESSAGE.getValue());)
```

Ik heb de methodes getest doormiddel van junit testen. Omdat ik nog geen view had heb ik junit test aangemaakt zodat ik de berichten in de database kon toevoegen zonder dat ik een view nodig had. Ook in het oog op het testen zodat ik aan het einde van de sprint nog een keer de test kon wanneer ik een aanpassing had gedaan aan de code. Voor elk type heb ik een test aangemaakt welke ieder 3 (dezelfde) ontvangers hebben. De parameters Scheduled, EndDate en MessageDate heb ik op vandaag gezet zodat er een datum staat ingevoerd. De taalondersteuning staat beschreven in de UML hoe deze in zijn werk gaat maar valt buiten de scope van het project. Ik heb wel de Java klasse Language aangemaakt. Deze kan in de toekomst gebruikt worden om de taalondersteuning toe te voegen via MySQL. Omdat dit niet tijdens mijn project is gerealiseerd heb ik Language aangemaakt als Nederlands(NL). (new Language("NL", "Nederlands", "")). Zodat dit wel al word meegenomen voor de toekomst.

De controllers heb ik getest doormiddel van junit testen hieronder een voorbeeld van een junit test welke ik heb gebruikt om het invoeren van een primair bericht te testen. Voor de leesbaarheid heb ik commentaar toegevoegd voor het invullen van de variabelen van een persoonlijk bericht.

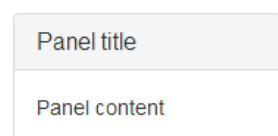
```
@Test
public void createPrimaryMessage() {
    PrimaryMessage mess = new PrimaryMessage();
    //[Set All variables]
    try {
        mess = Messages.createMessage(mess);
    } catch (Throwable e) {
        fail(e.getLocalizedMessage());
    }
    Assert.assertTrue(Messages.getMessageById(mess.getMessageid()).equals(
        mess));
}
```

5.1.3.3 View

Nadat de code volledig werkte en ik dit ook getest had met junit ben ik aan de slag gegaan met de view. Een view maken in XPages was voor mij helemaal nieuw behalve dat ik me de RSSfeed aan het begin van het project een klein beetje kennis gemaakt heb met XPages. Ik ben wel ervaren met HTML omdat ik dit vaak gebruikt heb op school of hobby projecten voor mezelf.

Ik ben begonnen met het uitzoeken hoe ik een XPage precies moest opbouwen. En welke onderdelen ik nodig had. Bij inONE wordt er een header gebruikt die ik kon gebruiken. Dit is een custom control. Een custom control is een herbruikbare “externe” pagina die op een eenvoudige wijze geïmplementeerd kan worden op meerdere pagina’s door deze aan te roepen doormiddel van de XPage custom control tag. door het element XC:[Custom control naam] aan te roepen aan te roepen kan je een custom control gebruiken op de pagina. XC staat voor XPage custom control, door dus bijvoorbeeld het element `<xc:ccHeader_beta>` kan je een custom control aanroepen

Nadat de basis van de pagina stond ben ik begonnen met het verder indelen van de pagina. De bedoeling was om over Bootstrap toe te passen. Bootstrap is een front end framework om er voor te zorgen dat er minder code nodig is om HTML pagina’s op te maken. Het werkt met CSS classes welke aangeroepen kunnen worden. Zo is het bijvoorbeeld mogelijk om kolommen aan te maken. En zijn de headers van een box al aangemaakt. (Zie Figuur 13 Bootstrap panel). En zijn de buttons al vormgegeven zodat deze met een aantal css klassen kan worden aangemaakt (met btn btn-default maakt je een standaard knop). Ook zorgt bootstrap ervoor dat de webpagina “responsive” is. Dit houdt in dat webpagina’s mee bewegen als de grootte van de browser groter of kleiner wordt.



Figuur 13 Bootstrap panel

Door bootstrap te gebruiken worden mijn pagina’s dus responsive en geoptimaliseerd, voor het mobiele platform omdat de pagina zich aanpast aan de grote van het scherm. In Figuur 14 Computer scherm is de uiteindelijke pagina geworden, op deze manier ziet de pagina er uit op een computer scherm. Figuur 15 Mobiel scherm is de weergave zoals die zou zijn op een mobiel scherm.

volledige lijst met ontvangers. Deze zijn belangrijk omdat deze parameters nodig zijn voor de werking van de custom control.

De picklist zal in de toekomst wel uitgebreid moeten worden omdat er geen zoekveld in voor komt. Hiervan is een feature request aangemaakt Wanneer een leverancier veel horeca ondernemers heeft wordt het een stuk lastiger om bepaalde ontvangers te selecteren i.v.m. de grootte van de lijst. Om deze reden heb ik geadviseerd om dit voor in de toekomst een zoekveld toe te voegen. Ook was de wens vanuit de opdrachtgever om groepen aan te kunnen maken. Beide functies heb ik een ticket voor aangemaakt in Jira zodat wanneer naar de picklist gekeken wordt deze onderdelen mee genomen kunnen worden.

Nadat dit was gedaan ben ik begonnen aan het tweede gedeelte. Er zijn 3 opties om het bericht in te vullen. Het aanmaken van een inONE bericht (Primair) was niet veel werk want hier hoeven geen extra attributen voor ingevuld te worden. Voor een mededeling heb ik in eerste instantie alleen 2 textboxes neergezet om er voor te zorgen dat de gegevens wel alvast in de database konden. Dit geldt hetzelfde voor een advertentie bericht in eerste instantie kon ik 1 product eraan koppelen door het id uit de database in te voeren. Dit heb ik puur om de reden gedaan zodat ik een eerste basis had om verder op te werken. In sprint 3 heb ik het opzoeken van 1 of meerdere producten uitgevoerd. En ook het uploaden van een PDF.

5.1.4 Black box Interface test

Met de blackbox test kwam ik erachter dat als geen titel invulden dat ik een NullPointerException kreeg. Of bij het verkeerd invullen van datum een FormatException. Om deze reden heb ik een extra story point aangemaakt voor validatie over het systeem van de ingevulde gegevens zodat dit opgevangen wordt.

De bevindingen zijn te vinden de testrapportage in Bijlage 10.

5.2 Sprint 2

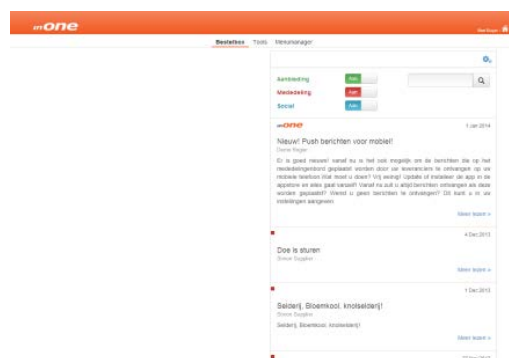
Hieronder een overzicht van de tickets welke ik heb opgelost tijdens deze sprint:

Key	Summary	Priority	Story Points
BER-26	wysiwyg editor uitzoeken en implementeren	Major	32
BER-12	Als een Stakeholder wil ik alle berichten in een "container" kunnen zien.	Critical	32
BER-10	Als een Stakeholder wil ik berichten via een online versie of een mobiele versie kunnen versturen en ontvangen	Critical	32
BER-7	Als een Stakeholder wil ik een filter mogelijkheid hebben voor de berichten zodat ik niet wordt overladen met berichten	Critical	64

Deze sprint was er voor bedoeld dat de berichten kunnen worden weergegeven. Ook als extra puntje heb ik een WYSIWYG uitgezocht welke op een goede manier kon werken met XPages en Bootstrap

De berichten van de berichtenservice moeten op het portal van inONE komen in een balk aan de rechterkant van het portal. Daarom heb ik een pagina gemaakt die helemaal wit is. Voor het idee heb ik een balk aan de rechterkant gemaakt zodat ik een beter idee kon krijgen hoe het er uiteindelijk uit komt te zien.(zie Figuur 16 Portal).

Ik heb aan de hand van de Mockups de view gemaakt met daarop alle functionaliteiten zoals het filteren, instellingen en de weergave van de berichten. Bij het maken van de Mockup heb ik rekening gehouden met de usability. Bijvoorbeeld door te kijken welke knoppen verborgen worden onder een andere knop door middel van een popup. Door gebruik te maken van Bootstrap heb ik tijdens het maken van de view niet veel rekening hoeven houden voor het mobiele platform omdat Bootstrap dit heel goed afhandelt.



Figuur 16 Portal

Nadat de portal klaar was ben ik aan de slag gegaan met een wysiwyg in de huidige situatie wordt CKEditor 3 gebruikt. Ik heb gekozen om deze te updaten naar CKEditor 4 omdat deze er meer uit ziet als Bootstrap en er nieuwe functionaliteiten biedt. In eerste instantie ben ik op zoek gegaan naar een wysiwyg die naadloos aansloot met bootstrap. En ook werkte in XPages. Ik heb een aantal editors geprobeerd en kwam uit bij een aantal editors namelijk, summernote, bootstrap-wysiwyg en wrap-bootstrap live editor. Door te proberen Word documenten en HTML documenten in de editors te plakken heb ik de beste uitgekozen. Dit was summernote. Hier ben ik mee aan de slag gegaan. De editor werkte prima. Alleen gebruikte summernote een DIV element om een editor van te maken. Helaas is het in XPages niet mogelijk om tekst uit een DIV uit te lezen. Hierdoor ben ik op zoek gegaan naar een andere editor. Uiteindelijk kwam ik uit bij CKEditor 4 omdat deze alle functionaliteiten bevat die ik wilde hebben. De functionaliteiten die ik wilde hebben waren bijvoorbeeld:

- het opmaken van tekst. (De standard opmaak zoals: Bold, Italic, underlined, heading etc.)
- Genummerd en ongenummerde lijsten maken
- HTML bewerken
- Plaatjes invoegen
- Links invoeren
- Undo en Redo functionaliteiten

Ook gebruikte CKEditor geen div maar een textarea om de tekst uit te lezen. De CKEditor heb ik een custom control voor gemaakt zodat deze op meerdere plekken hergebruikt kan worden als dit wenselijk is.

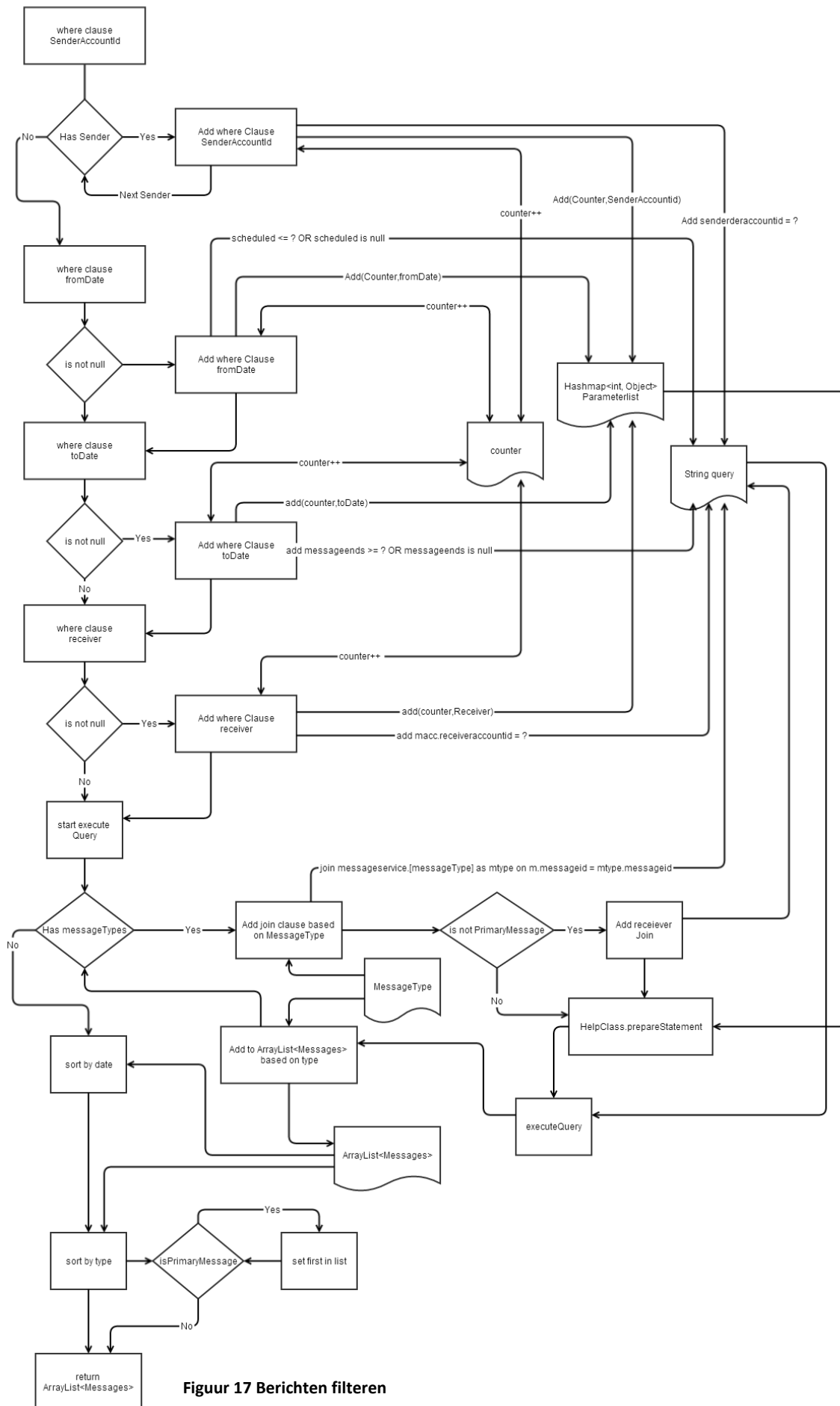
Als laatste heb ik filters aangemaakt zodat de berichten op de portal gefilterd kunnen worden naar de wens van de gebruiker. De gegevens moeten in de database opgeslagen worden voor de filters. Het opslaan van filterinstellingen heb ik in sprint 4 uitgevoerd. Deze sprint heb ik het daadwerkelijk filteren van de berichten gebouwd. Omdat MySQL snel is in het verwerken van gegevens heb ik ervoor gekozen om het filteren met MySQL te doen en niet met JAVA code. Dit was voor mij een moeilijke klus omdat dit voor mij onbekend was hoe ik dit moest aanpakken. Aan de hand van code voorbeelden van inONE ben ik ermee aan de slag gegaan. In Figuur 17 Berichten filteren (pagina 32) staat een flowchart diagram met de werking van de functie. (in Bijlage 8 staat de code die hierbij hoort)

- Stap 1: Where clause van de query opbouwen
- Stap 2: Join clause van de query opbouwen
- Stap 3: Statement vullen met de parameters a.d.h.v. prepareStatement
- Stap 4: Uitvoeren query
- Stap 5: Aan arraylist toevoegen a.d.h.v. messagetype
- Stap 6: Sorteren op datum
- Stap 7: Sorteren op type. Wanneer bericht primair is als eerste zetten.
- Stap 8: ArrayList met berichten teruggeven

5.2.1 Testen

De junit testen heb ik bijgewerkt met de nieuwe methode voor het filteren. De testen heb ik zo gemaakt dat ze herhaalbaar zijn. Mochten er zaken veranderen in de functie kan de test worden hergebruikt. De bevindingen van de junit testen zijn te vinden in de testrapportage in Bijlage 10

Deze sprint bevatte geen invoervelden die ik kon testen. Ik heb mijn test gebaseerd op wat er wordt getoond in de portal zoals: kloppen de datums aflopend/ zijn de datums ingevuld etc. De bevindingen en de test zijn te vinden in de testrapportage in Bijlage 10



Figuur 17 Berichten filteren

5.3 Sprint 3

Hieronder een overzicht van de tickets welke ik heb opgelost tijdens deze sprint:

Key	Summary	Priority	Story points
BER-32	Als een stakeholder wil ik via een type ahead een product kunnen zoeken in een nieuw bericht	Critical	128
BER-29	Als een stakeholder wil ik validatie hebben over de ingevulde gegevens	Critical	64
BER-40	Als een stakeholder wil ik een PDF kunnen uploaden	Critical	64
BER-41	Als een stakeholder wil ik oude berichten kunnen opzoeken	Major	32

De eerste stap in deze sprint was het uitzoeken hoe ik een PDF kan uploaden. Voor het uploaden heb ik gekozen om dit naar Amazon s3 te doen. Dit is een cloud service van amazon waarmee het mogelijk is om bestanden op te slaan en te delen. Ik heb voor S3 gekozen omdat PDF's soms grote bestanden zijn en dit veel tijd kan innemen om te uploaden/downloaden. S3 zorgt ervoor dat er altijd genoeg opslag is zodat de primaire servers van inONE ontlast worden.

Om bestanden te uploaden vanaf het web wordt gebruik gemaakt van de JAVA API van Amazon s3, hiermee kan ik de bestanden uploaden en een URL ophalen als ik het bestand wil aanroepen.

Het uploaden van een bestand via de API gaat als volgt voor de berichtenservice:

- Stap 1. S3Service initiëren aan de hand van Amazon credentials
- Stap 2. Key genereren
 - De key kan zelf gekozen worden en zal als bestandsnaam dienen bij amazon S3. Om de key uniek te maken heb ik gekozen voor een timestamp in combinatie met custlogid (dit is het id van de klant). Een timestamp heeft de precisie van seconden. Ik ben er vanuit gegaan dat het niet mogelijk is om 2 bestanden in dezelfde seconden te uploaden. Hiervoor zoude 2 browsers voor gebruikt moeten worden. 1 persoon heeft 1 account dus moet er heel snel geklikt worden om dit toch voor elkaar te krijgen om dezelfde seconden te uploaden.
- Stap 3. Het object sturen naar Amazon doormiddel van S3service.putObject(bucketnaam, bestand)
 - De bucketnaam is de naam voor de "bak" waarin de bestanden geupload moeten worden.
- Stap 4. De API van Amazon regelt het uploaden van de bestanden

Het ophalen van een bestand via de API gaat op de volgende wijze voor de berichtenservice:

- S3Service initiëren aan de hand van Amazon credentials
- Een url aanvragen door middel van createSignedGetUrl(bucketnaam, key uit de database, verloopdatum,
 - De link verloopt binnen 1 uur. Als je buiten de 1 uur het bestand wil aanroepen wordt er alleen een toegang geweigerd bericht weergegeven.
- De link kan gebruikt worden op de pagina.

- De url ziet er als volgt uit:
`https://inone.s3.amazonaws.com/XFO4IF/1382617874455_XS.pdf?AWSAccessKeyId=AKIAJCSLNU
 JOELNY7NQ&Expires=1389175284&Signature=olTymiZOLiDivlExS2uGKMZ2%2F4c%3D`

Na het maken van de methodes voor het ophalen en uploaden heb ik deze methodes gebruikt om een uploadfile element te maken welke het bestand upload.

Bij het testen in sprint 1 kwam ik erachter wanneer ik verkeerde informatie invulde ik errors krijg. Daarom ben ik aan de slag gegaan met (client-side) validatie van de gegevens. Hiervoor heb ik een jQuery plugin voor geselecteerd namelijk jQueryValidation. jQuery was al een bestaand onderdeel van de webpagina's

Per veld kan ik aangeven op welke manier ik deze wil valideren. Ik valideer op de volgende waardes:

- Datum velden moeten een datum zijn. (Inplannen op en verloopt op)
- URL veld moet een URL zijn (www.domeinnaam.nl) (Dit geldt alleen voor een mededeling)
- Een bericht titel is verplicht.

Door de validatie over de gegevens te doen dien ik minder af te vangen in de Java code, omdat de gegevens goed worden aangeleverd in Java doordat er al validatie over de gegevens is gedaan. Dit is voor de gebruikersvriendelijkheid ook beter omdat er sneller een reactie wordt gegeven welke gegevens niet kloppen. Als er iets niet klopt komt er een popup bij het invoer veld te staan welke niet in orde is. Zie Figuur 18 verplicht field



Figuur 18 verplicht field

Omdat leveranciers veel producten hebben was het wenselijk om een type ahead te maken voor de producten. Om dit op te lossen heb ik jQuery autocomplete widget gebruikt. Met deze widget is het mogelijk om aan de hand van JSON een type ahead te maken. De JSON wordt gebruikt voor de typeahead. JSON staat voor JavaScript Object Notation en gebruikt javascript syntax voor het omschrijven van data objecten.

De widget werkt op de volgende manier (zie Figuur 20 TypeAhead)

- Stap 1. Er wordt een waarde ingevuld in het invoerveld
- Stap 2. Bij 2 of meer tekens call naar JsonPagina.xsp met de waarde als postdata uit het invoerveld.
- Stap 3. Producten worden opgezocht in de database a.d.h.v. de waarde. Er word gezocht op naam en artikelnummer
- Stap 4. JSON word gegenereert van de gevonden producten
- Stap 5. JSON word gereturned met id, naam. Artikelnummer.
- Stap 6. JSON wordt gebruikt om het invoerveld aan te vullen met lijst van producten.
- Stap 7. Naam en artikelnummer worden weergeven in een lijst
- Stap 8. On hover op een item wordt de naam in het invoerveld weergeven.
- Stap 9. Onclick op een item wordt de naam in het invoerveld geset. Een hiddenfield word het id geset om te gebruiken voor het toevoegen.

Zodra er een waarde in het hidden field staat wordt een knop actief. Zodra je daar op klikt, wordt het product in een lijst toegevoegd voor het bericht (zie Figuur 19 Productenlijst)

Extra informatie

Zoek product

Bleekselderij +

- Andijvie holland
100001_KG
- Bleekselderij
100004_DS12ST**
- Bloemkool extra groot
100007_DS7ST
- Broccoli spanje
100014_KG
- Chinese kool klasse 1
100016_KG
- Knolraap
100024_KG
- Knolselderij
100025_KST10ST
- Bieten geel
100028_KG
- Mangold
100030_KST10KG
- Raapstelen
100037_ST

Figuur 20 TypeAhead

Extra informatie

Zoek product

Bleekselderij +

- Bleekselderij
100004_DS12ST
- Chinese kool klasse 1
100016_KG

Figuur 19 Productenlijst

5.3.1 Testen

Met de blackbox interface test heb ik gezocht op producten met artikel nummer, en op productnaam. Hier kwamen waardes uit de database uit welke ik vergeleken heb. Het uploaden van een PDF heb ik getest door een PDF te uploaden maar ook een JPEG bestanden. Hierbij kwam ik erachter dat het mogelijk was om andere bestandstypen te uploaden. Hier heb ik een storypoint voor aangemaakt zodat ik dit later op kon pakken. Het testen van de validatie heb ik uitgevoerd door verkeerde, maar ook goede waardes in te vullen. Beide gevallen verliepen zoals het zou moeten gaan.

De junit testen heb ik alleen gebaseerd op BER-41 "Als een stakeholder wil ik oude berichten kunnen opzoeken" omdat de andere story points niet met een junit kan worden getest.

De testrapportage is te vinden in Bijlage 10

5.4 Sprint 4

In sprint 4 heb ik een aantal dingen opgelost. Zo ben ik aan de slag gegaan met het opslaan van de filters en instellingen en heb ik ervoor gezorgd dat ook horecamanagers berichten konden versturen naar 1 of meerdere van zijn locaties. Als laatste onderdeel van de sprint ben ik aan de slag gegaan met het maken van de push berichten voor de app.

Hieronder een overzicht van de storypoints welke ik heb opgelost tijdens deze sprint:

Key	Summary	Priority	Story Points
BER-2	Als horecamanager wil ik een bericht kunnen sturen naar 1 of meer locaties.	Critical	32
BER-9	Als Stakeholder wil ik instellingen voor filters eenmalig in hoeven stellen, maar ook later kunnen wijzigen	Critical	64
BER-37	Als een stakeholder wil ik pagination hebben op de portal.	Critical	32
BER-36	De afbeeldingen bij de producten moeten bij een product in een aanbieding opgehaald worden. Dit ophalen uit notes database	Blocker	32

Het was als horeca manager nog niet mogelijk om berichten te sturen naar zijn locaties. Hiervoor heb ik een extra methode aangemaakt. Deze haalt de mogelijke ontvangers op van een leverancier en de mogelijke locaties van een horeca manager aan de hand van het type account.

De volgende stappen worden doorlopen:

- Stap 1: Controleer of het huidige account een leverancier is.
- Stap 2: Controleer of het huidige account een manager is.
- Stap 3: Als het account een leverancier betreft dan de horeca bedrijven ophalen
- Stap 4: Als het account een horecamanager betreft haal de sub accounts op.
- Stap 5: Geeft een lijst met ontvangers terug.

Na het ophalen van locatie ben ik in deze sprint aan de slag gegaan met het opslaan van de filters. Als eerste stap heb ik de modellen opgezet voor Filter en Settings, nadat dit gebeurt was heb ik de controllers gemaakt zodat ik gegevens kan uitlezen en opslaan in de database.

Ik heb er voor gekozen wanneer er een waarde in de database staat het filter op "aan" staat. Staat de waarde niet in de database dan betekend dit het filter op uit staat. Ik heb er bewust voor gekozen om dit op deze manier te doen omdat deze methode ruimte bespaard. Wanneer er een filter niet op "aan" staat, staat dit ook niet in de database waardoor minder ruimte nodig is. Instellingen worden wel opgeslagen in de database als een boolean omdat deze voor 1 account altijd 1 regel bevatten

Bij het initiëren van de Settings controleer ik of er instellingen bekend zijn over het huidige account. Als er voor het account geen instellingen bestaan worden alle instellingen op 'aan' gezet zodat de eerste keer als een account wordt ingelogd de instellingen altijd aan staan. Ook worden de filters op aan gezet zodat

gebruikers altijd alle berichten zien bij de eerste keer inloggen en bekijken van de portal. Met een extra functie genaamd “inDtb” controleer ik of de gegevens in de database staan. Mocht dit niet het geval zijn, dan worden de waardes in de database toegevoegd. Aan het einde van elke initiatie van instelling haal ik de gegevens op.

Ik heb de controller klassen niet gemaakt zoals de andere controllers met een functie om alle gegevens in te voeren van alle gegevens om de instelling te wijzigen. Ik heb er voor gekozen om een zogenaamde “switch” statement te maken voor alle instellingen en filters. Voor het filteren wordt er gekeken of de waarde in de database voorkomt. Staat de waarde in de database dan wordt deze verwijderd a.d.h.v. het filterid. Staat de waarde niet in de database dan wordt deze opgeslagen in de database. Voor een instelling gebruik ik het update statement, bij de instellingen geef ik een boolean mee.

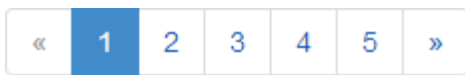


Na het opslaan van de gegevens en grondig testen of de waardes opgeslagen worden (zie blackbox test sprint 4) Ontdekte ik dat de buttons die ik gebruikte en had aangemaakt in de view niet werkte zoals ze zouden moeten werken. Dit om de reden dat het niet duidelijk was of een filter aan of uit stond. Hierdoor werd de usability minder en ben ik op zoek gegaan naar een andere oplossing. Na onderzoek op internet hoe andere websites/applicaties werkte kwam ik een afbeelding tegen met instellingen knoppen van de iPhone (zie Figuur 21 iOS Buttons) waarbij duidelijk was of de instelling aan of uit stond.

Figuur 21 iOS Buttons Nadat ik de keuze had gemaakt om deze button te gebruiken heb ik op internet onderzoek gedaan naar de beste oplossing. Met een javascript library is het mogelijk (switch-button.js) om deze knoppen te generen vanuit een checkbox Een checkbox uitlezen werd hiermee succesvol dus heb ik deze geïmplementeerd.

Als eerste stap heb ik de buttons veranderd voor checkboxes zodat ik zeker wist dat de filter mogelijkheid nog werkt met de checkboxes. Op de “onclick” van de checkboxes voer ik het daadwerkelijk filteren uit. Zodra de checkboxes werken heb ik de javascript switcher aangeroepen. Door een DIV element om de checkbox te zetten wordt de checkbox een switch element. Daarna ben ik het switch element gaan testen of deze werkte. Helaas kwam ik erachter dat de “onclick” van de checkbox niet correct meer werd aangeroepen.

Ik heb eerst geprobeerd om via javascript de “onclick” van de checkbox aan te roepen. Alleen werkte dit niet omdat de “onclick” nu 2 maal werd aangeroepen en niet 1 keer. Waarom dit gebeurd heb ik helaas niet kunnen achterhalen. Bij de checkbox heb ik een button neergezet die ik verberg. De clientside code van de onclick wordt wel aangeroepen. Dus zodra de clientside code word aangeroepen roep ik de “onclick” aan van de button zodat de instelling of filter word aangepast in de database en de filter mogelijkheid weer werkt. Helaas is dit niet ideaal dat er een extra element bij moest komen. Alleen was er geen andere manier op dat moment om dit op te lossen.



Figuur 22 Pagination

Voor het pagination heb ik het bootstrap pagination onderdeel gebruikt. In combinatie met het XPage element pagination. Omdat ik het XPage pagination niet op de goede manier kon aanpassen zodat deze in het design paste. Het XPage element lees ik de

waardes uit hoeveel paginas er zijn en wat de huidige pagina is. Aan de hand daarvan genereer ik het html element van bootstrap. Door een loop te maken door alle paginas. Door `pager.gotoPage` aan te roepen kan er naar een bepaalde pagina genavigeerd worden in Figuur 22 Pagination is hoe het er uiteindelijk uit is komen te zien.

5.4.1 Testen

Deze sprint heb ik de jUnit testen gebaseerd op het opslaan van filteren, en het versturen naar horeca managers. Voor de blackbox test heb ik wel gekozen om de testen uit voeren voor alle story points in deze sprint. Deze heb ik gebaseerd op wat ik op de view zie. De bevindingen zijn te vinden in de testrapportage in Bijlage 10.

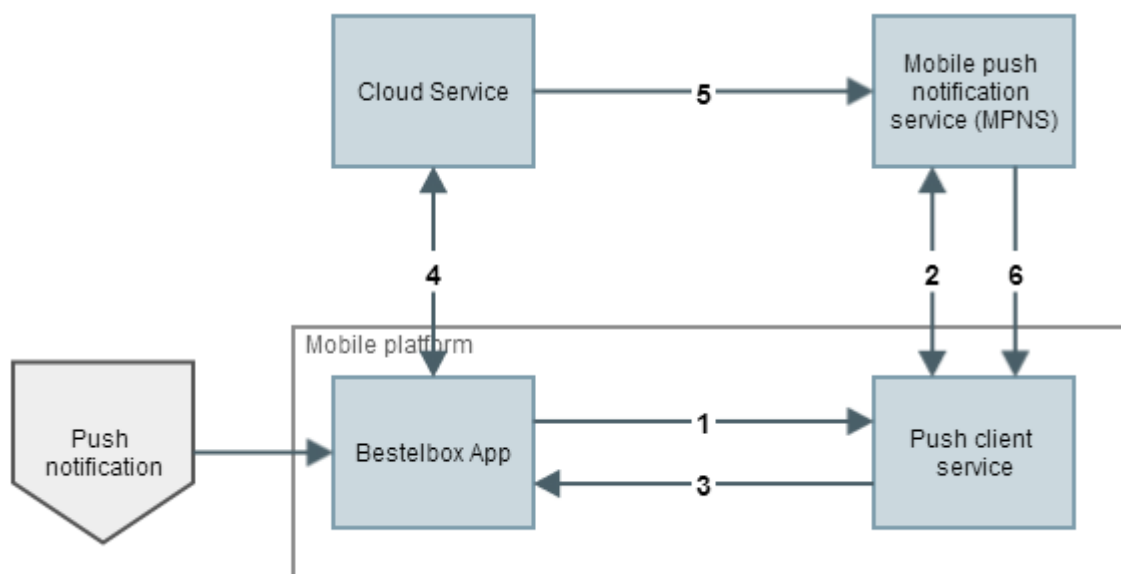
5.5 Sprint 5

Hieronder een overzicht van de storypoints welke zijn opgelost tijdens deze sprint:

Key	Summary	Priority	Story Points
BER-8	Als een bestelbox app gebruiker wil ik notificaties krijgen van belangrijke berichten op mijn mobiel apparaat en in de statusbalk van het mobiele platform weergegeven worden.	Critical	64
BER-38	BER-8 Android notificaties maken	Critical	128
BER-39	BER-8 IOS notificatie maken	Major	128

De laatste sprint stond in het teken van de push notificaties voor de mobiele app. Voordat ik de daadwerkelijke push notificaties heb gebouwd heb ik onderzocht hoe de Push services voor de iOS en Android werken.

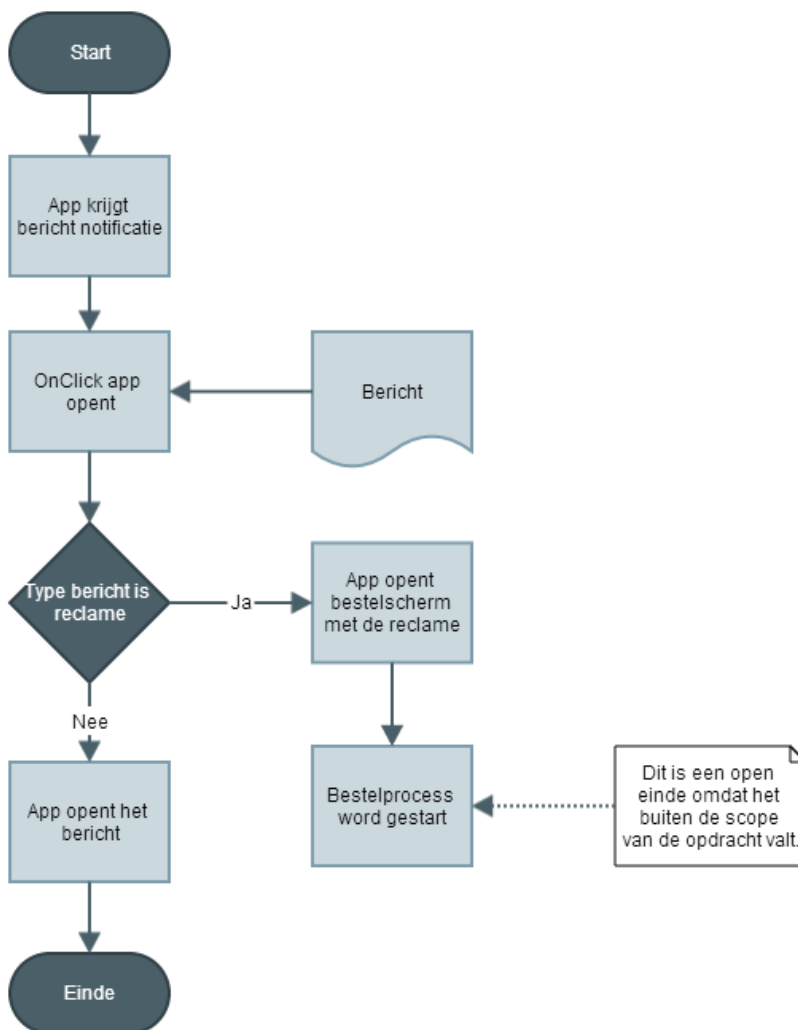
Nadat ik het heb uitgezocht kwam ik tot de conclusie dat de service op de volgende manier werkt:



Figuur 23 Werking MPNS

1. De app doet een aanvraag voor een push notificatie URI van de Push client service.
2. De Push client communiceert met de Mobile push notification service en de service geeft een notificatie URI(Uniform Resource Identifier) terug.
3. De push client geeft de URI terug naar de App.
4. De App kan een notificatie URI zenden naar de cloud service.
5. Wanneer de cloud service info heeft voor de App gebruikt hij de notificatie URI om een bericht te sturen naar de MPNS.
6. De MPNS zorgt ervoor dat de Push notificatie wordt verzonden naar de App.

Ook had ik bedacht dat zodra de app een notificatie binnen krijgt de app moet bekijken of dit een reclame bericht is of niet. Daarbij hoort de volgende flow:



Figuur 24 Flow App. Notificatie

Nadat ik dit had onderzocht ben ik aan de slag gegaan met het daadwerkelijk bouwen van de push notificatie.

Als eerste heb ik het abstract factory pattern (Zie hoofdstuk 4.5.2) uitgewerkt en opgezet zodat de model en de controller gebruikt konden worden. Ik heb voor het project alleen de Android push notificaties gemaakt. Wegens prioriteiten voor mijn stageverslag heb ik in overleg met de opdrachtgever dit onderdeel tijdelijk laten vervallen. En zal ik dit onderdeel na mijn stage verder afmaken. Dit komt omdat ik me verkeken heb op het ontwikkelen voor de iOS bij het plannen. Meer hierover staat in mijn project evaluatie hoofdstuk.

Twee methodes zijn benodigd aan de kant van de server voor het versturen van berichten, namelijk het registreren van de app zodat de gegevens in de database terecht komen. En het sturen van de notificatie naar mobiele telefoons.

Door gebruik te maken van het abstract factory pattern zijn de aanroepen van de methodes een stuk makkelijker omdat er maar eenmalig aangegeven dient te worden om welke OS het gaat. Voor de toekomst kunnen hierdoor ook meerdere platformen worden geïmplementeerd zonder te veel code te hoeven veranderen.

Voor het versturen en ontvangen van push berichten gebruikt Android gebruik van GoogleCloudMessagging (GCM) In Figuur 23 Werking MPNS staat hoe deze service in zijn werk gaat. Google heeft een JAVA API voor GCM deze heb ik daarom ook gebruikt om berichten te versturen.

De methode om een bericht te versturen werkt volgens de volgende stappen:

Stap 1. GCMSender initiëren

- a. Dit is een helper class om berichten naar de GCM service te sturen a.d.h.v de API Key.

Stap 2. GCMBuilder initiëren

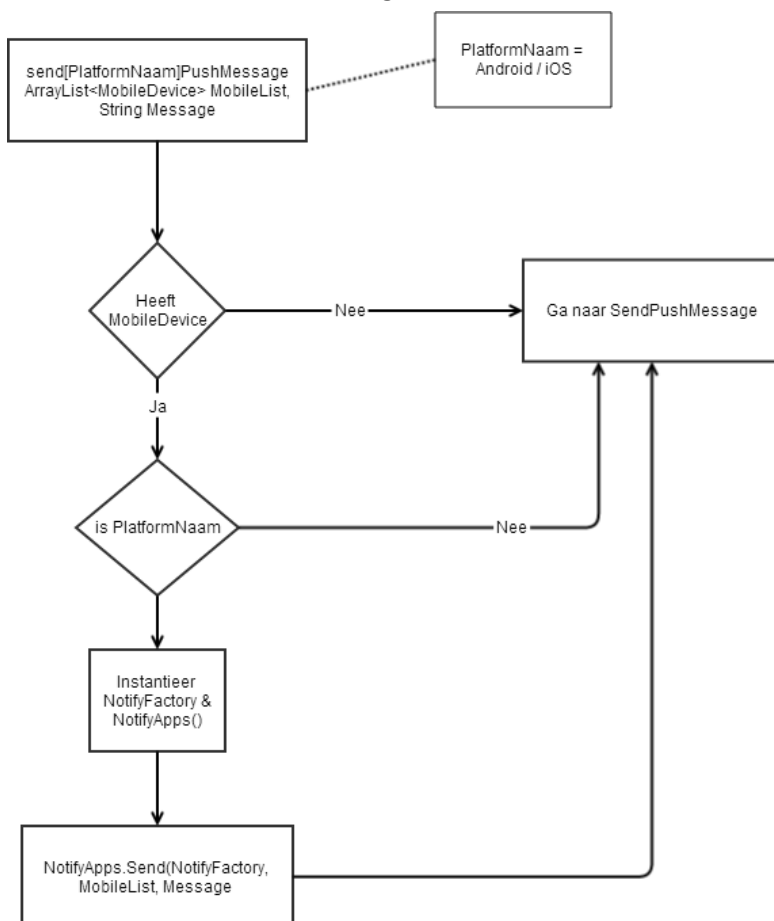
- a. Hiermee bouw je een bericht op.

Stap 3. Data toevoegen aan de GCMBuilder welke ontvangen kan worden door de app.

Stap 4. Sender.send aanroepen met als parameters de lijst van registratie ID's van de mobiele devices en een X aantal pogingen om een bericht te versturen

Om een bericht te versturen heb ik vanuit de view een aanroep naar een methode sendPushMessage met een lijst van mobiele devices en het bericht als parameters. Deze methode roept sendAndroidPushMessage en sendiOSPushMessage met dezelfde parameters als sendPushMessage

Beide methodes hebben de volgende flow:



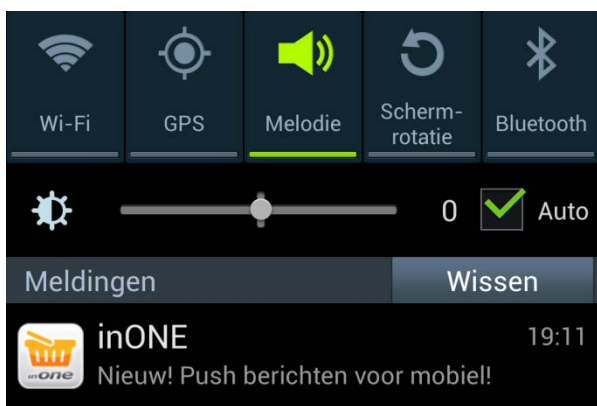
Er wordt voor elke mobile device gecontroleerd of het een Android is of een iOS en aan de hand daarvan word de servicefactory geïniteerd en het bericht verstuurd door de service factory mee te geven

Figuur 25 Flow Versturen bericht

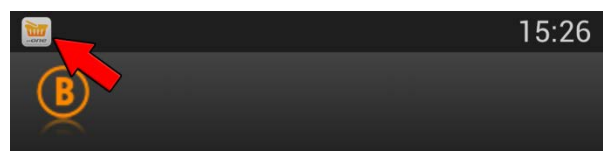
Om de app te registreren heb ik een algoritme geschreven die door middel van een POST request de applicatie registreert bij GCM en op de server van inONE. Het registreren van GCM wordt afgehandeld door de GCM API in de app. Om te registreren bij inONE heb ik een pagina aangemaakt op de server welke POST data kan ontvangen en verwerken. De app stuurt vanaf welk platform de request komt, de registratie id en de username van inONE.

De app bij inONE is in principe een website. Om deze reden moest ik op een bepaalde manier erachter komen wie er is ingelogd. Dit heb ik opgelost door zodra er ingelogd wordt, wordt de username en het wachtwoord in een cookie opgeslagen. De app vraagt de cookie van de webpagina aan zodat deze de waardes uit kan lezen. Heeft de app een username dan word de mobiele device geregistreerd bij inONE en bij GCM. Als er geen cookie bestaat zal er dus geen registratie plaatsvinden van het mobiele device

Wanneer een bericht wordt aangemaakt in de berichtenservice worden naar alle mobiele apparaten een bericht verstuurd die in de lijst staan van ontvangers als deze ontvanger heeft ingesteld om een bericht te ontvangen. Het bericht komt er uit te zien zoals in Figuur 26 Notificatie. Ook word er in Android het icoontje van de inONE app weergegeven in de status balk zoals te zien is in Figuur 27 App icoon. De berichten die ingepland staan op een later tijdstip worden pas de dag verzonden dat het bericht is ingepland.



Figuur 26 Notificatie



v6.16

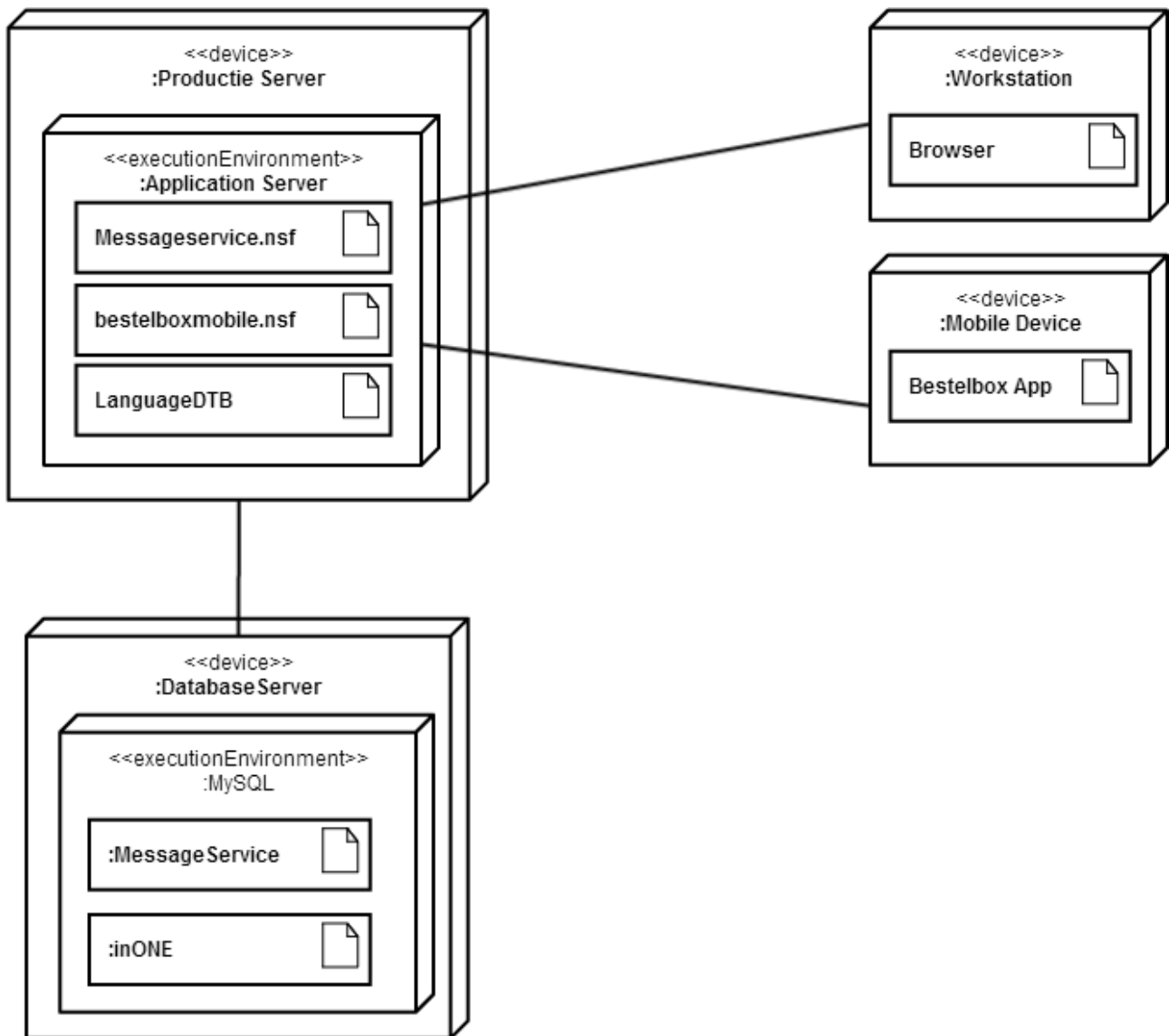
Figuur 27 App icoon

5.5.1 Testen

De junit test heb ik gebaseerd op de statuscode die GCM mij teruggeeft wanneer er een bericht verstuurd wordt. Uiteraard kan ik zelf niet kijken of het bericht daadwerkelijk is aangekomen met junit. Hiervoor heb ik de blackbox test gebruikt. De testen verliepen beide zoals het zou moeten zonder fouten. De bevindingen zijn te vinden in de testrapportage in Bijlage 10.

6 Deployment diagram

In dit hoofdstuk staat het deployment diagram. Hierin is de uiteindelijke weergave te zien van de servers in productie.



Figuur 26 Deployment diagram

7 Product evaluatie

Er staan nog een aantal story points open welke opgepakt zullen worden door de programmeurs van inONE zodat de berichtenservice nog verder door ontwikkeld wordt. Doordat de diagrammen en storypoints er liggen zal het ontwikkelen sneller gaan omdat er minder nagedacht dient te worden over de exacte werking van de functies. In hoofdstuk 4.1 staan de requirements en of de story points zijn uitgevoerd. Ik kan dus zeggen dat het project geslaagd is.

8 Project evaluatie

In dit hoofdstuk staat de evaluatie van mijn project. Hoe heb ik dit uitgevoerd. En hoe zou ik dit in het vervolg aanpakken.

iOS heb ik niet ontwikkeld in mijn afstudeer periode. Deze functionaliteit zal ik na mijn afstudeerstage oppakken om wel deze functie erin te bouwen. iOS wordt ontwikkeld in Objective-C. Ik had verwacht dat dit zou werken met Java of iets soort gelijks, hierdoor is dit voor mij onbekend terrein en zou het te veel tijd in beslag gaan nemen om dit te gaan ontwikkelen. Ook werkt iOS en Android verschillend qua implementatie waardoor er twee keer een App en twee keer Java code op de server ontwikkeld moet worden. Mijn inschatting was dat de implementatie wel verschillend zou zijn. Alleen niet dat dit heel anders zou werken.

Tijdens het project heb ik agile sprints bijgehouden. Wat ik kan verbeteren aan mijn agile sprints is het goed bijhouden van de sprints, omdat ik vrij goed weet welke functionele / niet functionele eisen er aan de applicatie gemaakt moet worden zijn er een aantal momenten geweest dat ik de sprints niet bij heb gehouden waardoor sommige sprints maar 1 dag duurde terwijl ik er wel langer mee bezig ben geweest. Omdat ik individueel werk heeft dit geen grote gevolgen voor het project. Alleen wanneer er meer in teamverband gewerkt zal worden kunnen bepaalde taken mogelijk dubbel gedaan waardoor het extra tijd kan kosten wat onnodig zou zijn omdat er dubbel tijd in gestoken is.

Aan het begin van het project heb ik alle sprints ingedeeld. Hier heb ik de feedback op gekregen dat dit niet in verhouding ligt met agile. Hierdoor heb ik goed gekeken hoe agile wel zou moeten en heb ik mij daarop aangepast zodat ik agile SCRUM beter toepas. Daarom heb ik de sprints die ik aangemaakt had en ingedeeld verwijderd zodat ik meer op de manier van Agile zou werken. Wanneer ik klaar was met een sprint heb ik de volgende sprint gemaakt in plaats van dat ik alle sprints al klaar had staan. Wel heb ik onderdelen van de waterval methode ook toegepast. Dit komt omdat de wens vanuit de opdrachtgever kwam om alle functionele eisen op tafel te krijgen voordat ik ben begonnen met het programmeren.

9 Verantwoording beroepstaken

In dit hoofdstuk staan de 3 beroepstaken welke ik heb uitgevoerd tijdens mijn afstuderen. En op welke manier ik deze behaald heb tijdens mijn project.

9.1 Beroepstaak 3.2 Ontwerpen systeemdeel

Voor de beroepstaak 3.2 ontwerpen systeemdeel heb ik een groot deel van de keuzes gemaakt hoe de werking van het systeem zal moeten worden en hoe de schermen er (ongeveer) uit komen te zien. Ook heb ik tijdens het ontwerpen een database bestand aangemaakt en laten controleren op onvolkomenheden en hoe ik het beste de implementatie kon doen. Dit is niet volledig op de manier van SCRUM welke ik gebruik voor dit project. Ik heb deze keuze gemaakt omdat de opdrachtgever graag vooraf wilde weten hoe het systeem er uit komt te zien, en wat de functionaliteiten moesten worden.

Ik heb gebruik gemaakt van de ontwerptechniek UML. Ik heb voor Unified Modeling Language(UML) gekozen als ontwerptechniek omdat ik op school ervaring heb opgedaan met UML. In overleg met de bedrijfsmentor hebben we daarom ook voor deze techniek gekozen. Dit kwam ook mede doordat bij inONE vaker UML wordt gebruikt. Om UML diagrammen te maken heb ik Gliffy gebruikt ^[Gliffy] Gliffy is een web-based diagram, hiermee is het mogelijk om flowcharts, netwerk-, UML- en andere -diagrammen te maken. Deze tool wordt gebruikt binnen inONE via een confluence^[Confluence] plugin.

Voor het klassediagram heb ik een aantal keuzes gemaakt. Ik heb gekozen voor het abstract factory pattern

Ook ik heb een aantal GUIs ontwikkeld voor de berichtenservice hiervoor heb ik gebruik gemaakt van de balsamiq plugin van confluence. Op deze manier kon ik op een gestructureerde wijze mock-ups maken omdat deze plugin er voor ontwikkeld is. Door de Mockups te maken werd het een stuk duidelijker welke onderdelen er in het systeem moesten komen. Hierdoor werden de functionaliteiten duidelijker. De Mockups waren bedoeld om meer inzicht te geven in hetgene wat ik in gedachte had en om met de stakeholders goed te kunnen overleggen over de functionele eisen van het systeem. Bij alle Mockups heb ik een beschrijving gemaakt zodat duidelijk is wat de GUI inhoudt en welke functionele eisen erin verwerkt zijn.

Voor beroepstaak 3.2 Ontwerpen systeemdeel heb ik dus gebruik gemaakt van de ontwerp techniek UML voor de klassen diagram en voor het ontwerpen van de GUI heb ik Mockups gemaakt. Deze technieken heb ik op een adequate manier toegepast door goed te kijken naar de specificaties van de ontwerptechnieken. Voor beide heb ik hiervoor dan ook een relevante tool gebruikt welke beide gebruikt worden bij inONE en daarom ook mijn keuze is gevallen op deze tools. De beschrijvingen staan bij alle Mockups en bij het klassediagram en heb de structuur en functionele eisen zo goed mogelijk doorgevoerd in de ontwerpen.

9.2 Beroepstaak 3.3 Bouwen applicatie

Het bouwen van de applicatie heb ik gedaan aan de hand van het UML Diagram en de Mockups. De Mockups zijn in grote lijnen en qua functionaliteit vrijwel hetzelfde gebleven. Ik heb dus alle functionaliteit uit de diagrammen en de Mockups gehaald. Door gebruik te maken van het MVC principe heb ik er zorg voor gedragen dat de code gemakkelijk hergebruikt, getest en wijzigbaar is omdat alle functionaliteit zo veel mogelijk gescheiden is. Het is dus volledig object georiënteerd geprogrammeerd.

Het systeemdeel werkt zoals de requirements zeggen. Ik heb er zorg voor gedragen dat dit gebeurd is door de story points een voor een af te werken. Zodat ik geen functionaliteit mis of kan vergeten. Ik heb gebruik gemaakt van XPages i.c.m JAVA door de richtlijnen te volgen heb ik dit op de juiste wijze toegepast.

Ik heb gebruik gemaakt van OO programmeren en het MVC principe. Dit zijn standaarden die vaker gebruikt worden. Ook heb hoofdletters in Java waar dit nodig was. Zoals Klassen altijd met een hoofdletter beginnen. En variabele namen met een kleine letter. Variabelen van enumeraties heb ik allemaal hoofdletters voor gebruikt omdat dit ook gebruikelijk is zodat er goed te herkennen is wat een enumeraties is.

Ik heb gebruik gemaakt van Domino Designer en Domino Notes beide zijn door IBM ontwikkeld om XPages mee te ontwikkelen en te beheren.

De applicatie is zo opgedeeld (via het MVC platform) dat nieuwe functionaliteiten gemakkelijk toegevoegd kunnen worden. Ook heb ik JavaDoc toegevoegd om het overdragen van de code makkelijker te maken en meer begrijpbaar te maken

9.3 Beroepstaak 3.5 Uitvoeren en rapporteren over het testproces

Ik heb aan de hand van de diagrammen en mockups een testbasis opgesteld en uitgewerkt. De testbasis van mijn testen zijn mijn user stories. De testontwerpen zijn op deze onderdelen gebaseerd en ik heb gebruik gemaakt van T-map next om dit maken. Het testdraaiboek is per sprint bijgewerkt en alle user stories zijn uitgewerkt zodat deze altijd up to date is. Elke sprint heb ik de testen uitgevoerd en zijn daarmee volgens de planning uitgevoerd in de bijlage staat de testrapportage.

10 Begrippenlijst

Bitbucket	Bitbucket is een repository tool. Hiermee word de code centraal opgeslagen zodat ieder teamlid bij alle code kan komen.
Jira	Jira is een ticket volgsysteem en een project management tool voor ontwikkel teams.
Confluence	<p>Confluence is een wiki waar alle documentatie van inONE is opgeslagen en waar ieder teamlid gemakkelijk bij kan. Confluence is een AddOn van Jira en is geïntegreerd hierin.</p> <p>[41.nl 2] 'Confluence is een eenvoudige, maar krachtige wiki, wat u in staat stelt om pagina's, documenten en multimedia bestanden aan te maken en te delen met uw team.'</p>
Gliffy	Online diagram en flowchart software welke in JIRA beschikbaar is via de Gliffy Confluence Plugin.
SAAS	Software as a service.
Mock-up	Een prototype van het design op ware grote.
XPage	Rapid mobile en web development technologie
JavaServer Faces (JSF) is een Java specificatie voor het bouwen van user interfaces of web applicaties	

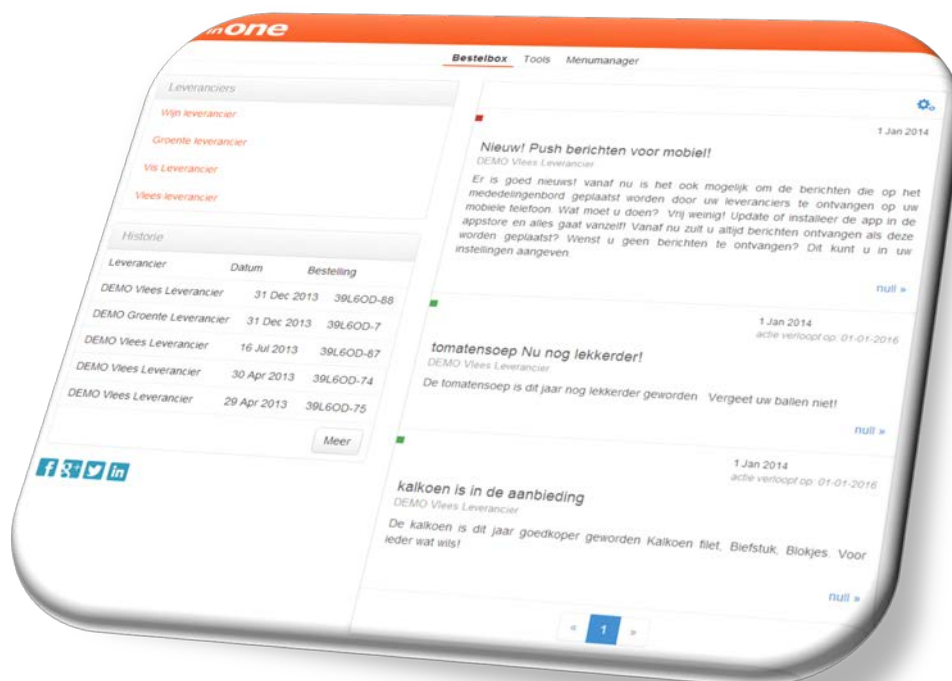
Bronnenlijst

- Notes in 9. (2012). XPages jumpstart. Geraadpleegd op <http://notesin9.com/index.php/xpages-jumpstart/>
- Atlassian (2013). Issue & Project Tracking Software | Atlassian Geraadpleegd op <https://www.atlassian.com>
- Donnelly M., McGucking T., Wallace M. (2011). Mastering XPages: A Step-by-Step Guide to XPages Application Development and the XSP Language. Geraadpleegd op <http://www.redbooks.ibm.com/Redbooks.nsf/ibmpressisbn/9780132486316>
- inONE. (z.j.). eCommerce software voor horecagroothandels | inONE.nl. Geraadpleegd op <http://inone.nl/>
- International Business Machines Corporation. (z.j.). IBM Notes and Domino Application Development wiki. Geraadpleegd op <http://www-10.lotus.com/ldd/ddwiki.nsf/dx/tutorial-intro-to-xpages.htm>
- openntf inc. (z.j.). XPages. Geraadpleegd op <http://xpagesblog.com>
- Schwaber K., Sutherland J. (2011). The Scrum Guide - The Definitive Guide to Scrum: The Rules of the Game. Geraadpleegd op <http://www.verheulconsultants.nl/Scrum%20Guide%20-%202011.pdf>
- BootstrapSwitch (2014) The switcher for Bootstrap Geraadpleegd op <http://www.bootstrap-switch.org>
- Xpage - wikipedia (z.j.) Xpage Geraadpleegd op <http://en.wikipedia.org/wiki/XPages>
- AndroidHive (2012.) Android Push Notifications using Google Cloud Messaging geraadpleegd op <http://www.androidhive.info/>
- Tcl-digitrade (2013) Providing JSON output from xPages application ... (avoiding GSON library) Geraadpleegd op <http://blog.tcl-digitrade.com/blogs/tcl-digitrade-blog.nsf/dx/28.01.2013090943DMABL6.htm>.
- StackoverFlow (z.j.) Stackoverflow geraadpleegd op <http://stackoverflow.com/>
- Oracle (z.j.) JavaServer Faces Technology geraadpleegd op <http://www.oracle.com/technetwork/java/javaee/javaserverfaces-139869.html>

Scriptie - Bijlages

Ontwikkelen berichtenservice

Marcel Fasseur



Student: Marcel Fasseur
Studentnr: 10069968
Onderwijsinstelling: De Haagse Hogeschool
Opleiding: HBO Informatica

Studieloopbaanbegeleider: Mevr. H.G.J. Bechet-Tjoonk
Begeleidend examinator: Dhr. G.A. Mijnaerends
Tweede examinator: Mevr. G.E. in 't Veld

Datum: 10 januari 2014
Versie: 1.0
Status: Final

Versiebeheer

Versie	Wijziging(en)	Datum
0.1	Mockups ingevoegd.	08-10-2013
0.2	Class diagram en SQL toegevoegd	10-10-2013
1.0	Alle bijlages samengevoegd tot 1 document	08-01-2014

Inhoudsopgave

Bijlage 1.	Afstudeerplan.....
Bijlage 2.	Plan van Aanpak.....
Bijlage 3.	Product backlog
Bijlage 4.	Notulen
Bijlage 5.	Klassediagram
Bijlage 6.	SQL DDL.....
Bijlage 7.	Mockups.....
Bijlage 8.	Code voorbeeld filteren bericht.....
Bijlage 9.	Test plan.....
Bijlage 10.	Testrapportage

Bijlage 1. Afstudeerplan

Afstudeerplan

Informatie afstudeerder en gastbedrijf

Afstudeerblok: 2013-2.1 (start uiterlijk 2 september 2013)
Startdatum uitvoering afstudeeropdracht: 2 september 2013
Inleverdatum afstudeerdossier volgens jaarrooster: 10 januari 2014

Studentnummer: 10069968
Achternaam: Dhr. Fasseur
Voorletters: M.A
Roepnaam: Marcel
Adres: Henriette van der Meestraat 18
Postcode: 2331SE
Woonplaats: Leiden
Telefoonnummer: 071-5765912
Mobiel nummer: 06-18115824
Privé emailadres: marcel.fasseur@gmail.com

Opleiding: Informatica
Locatie: Den Haag
Variant: Voltijd

Naam studieloopbaanbegeleider: · Mevr. H.G.J. Bechet-Tjoonk
Naam begeleidend examinerator: Dhr. G.A. Mijnaerends
Naam tweede examinerator: Mevr. G.E. in 't Veld

Naam bedrijf: inONE
Afdeling bedrijf: Ontwikkeling
Bezoekadres bedrijf: Lekstraat 88e
Postcode bezoekadres: 2314 VJ
Postbusnummer:
Postcode postbusnummer:
Plaats: Leiden
Telefoon bedrijf: 071 - 512 05 91
Telefax bedrijf:
Internetsite bedrijf: <http://inone.nl>

Achternaam opdrachtgever: Dhr. Belt
Voorletters opdrachtgever: L.
Titulatuur opdrachtgever:
Functie opdrachtgever:
Doorkiesnummer opdrachtgever:
Email opdrachtgever: lbelt@inone.nl

Achternaam bedrijfsmentor: Dhr. Belt
Voorletters bedrijfsmentor: L.
Titulatuur bedrijfsmentor:
Functie bedrijfsmentor:
Doorkiesnummer bedrijfsmentor:
Email bedrijfsmentor: lbelt@inone.nl

Doorkiesnummer afstudeerder:
Functie afstudeerder (deeltijd/duaal): Voltijd

Titel afstudeeropdracht:

Ontwikkelen van een berichtenservice bij inONE

Opdrachtoomschrijving**1. Bedrijf**

inONE is een software bedrijf gevestigd in Leiden en is opgericht in 2004. Ze maken eenvoudige e-commerce software voor horecaleveranciers en handige tools voor de horeca. inONE is een profit organisatie in de ICT-branche met 9 personeelsleden. De doelstelling van inONE is dat elke horecaonderneming/keuken in Nederland de mogelijkheid heeft om via één punt "Bestelbox" bij al zijn eigen leveranciers kan bestellen. Hierbij wordt gericht op horecaonderneming/keuken en leveranciers.

De afdeling waar deze opdracht voor wordt uitgevoerd is de afdeling ontwikkeling. Deze afdeling omvat 3 personeelsleden. Deze afdeling houdt zich bezig met het bedenken en maken van nieuwe onderdelen van Bestelbox. Ook zorgt de afdeling ontwikkeling ervoor dat het huidige systeem up-to-date blijft en bugs worden opgelost.

2. Probleemstelling

Hoe kunnen de leveranciers die samenwerken met inONE op een afgesproken manier berichten versturen naar specifieke horecabedrijven via zowel het mobiele platform als online versie?

In de huidige situatie kunnen blog berichten worden verstuurd via Bestelbox. Deze blog berichten kunnen alleen via de computer worden gelezen en verstuurd. De berichten kunnen naar specifieke horecabedrijven worden verzonden of naar allemaal.

In de nieuwe situatie moet het mogelijk zijn om korte reclame berichten te sturen ter illustratie: "100kg Zalm in de aanbieding voor €80, -". De berichten moeten zowel via het mobiele platform (De inONE Bestelbox app) als via de online versie verstuurd en ontvangen kunnen worden. Deze berichten kunnen korte reclame berichten of blog berichten zijn. De reclame berichten moeten gekoppeld worden aan Bestelbox zodat de korting die in het bericht staat verrekend wordt tijdens het bestellen. Dit zal een aanpassing worden aan het huidige systeem.

De inONE Bestelbox app moet via push berichten kunnen ontvangen en in de statusbalk van het mobiele platform weergegeven worden. Zodra er op de notificatie wordt geklikt moet de app tevoorschijn komen. In de app zal de bestelling geregeld kunnen worden en zal de korting die in het bericht staat verrekend worden bij de bestelling. Dit onderdeel kan complex worden omdat iedere mobiele platform anders zal werken waardoor er veel uitgezocht zal moeten worden.

Berichten moeten via Facebook gepost kunnen worden. Dit bericht moet worden opgepakt en getoond worden aan alleen de klanten van die leverancier. Daarnaast moet een leverancier ook voor een kleinere selectie groep berichten kunnen maken. Het berichten systeem moet rechtstreeks in verbinding staan met de "groeps indeling" van Bestelbox. Er kunnen een aantal typen leesrechten onderscheiden worden:

Leverancier kan:

- een bericht voor al zijn klanten maken.
- een bericht voor 1 of enkele van zijn klanten maken.

Horeca (manager) kan:

- een bericht voor alle locaties maken.
- een bericht voor 1 of enkele locaties maken.

inONE kan:

- een bericht voor alle leveranciers maken.
- een bericht voor alle horeca maken.
- een bericht voor 1 of enkele leveranciers maken.
- een bericht voor 1 of enkele horeca locaties maken.

Het GUI ontwerp en de usability is een onderdeel van de opdracht. De kleurtjes en plaatjes daar in tegen wordt gedaan door de designer van inONE.

Een korte opsomming van de requirements voor de berichtenservice:

- Blog berichten ontvangen en versturen.
- Reclame berichten ontvangen en versturen.
- Berichten via zowel de online versie als het mobiele platform versturen en ontvangen.
- De inONE Bestelbox app moet via push berichten werken op elk mobiel platform.
- De reclame berichten moeten worden gekoppeld via het order systeem van inONE
- Berichten moeten via Facebook gepost kunnen worden.

3. Doelstelling van de afstudeeropdracht

Het doel van de afstudeeropdracht is er om er voor te zorgen dat inONE een berichtenservice krijgt die aan de wensen en eisen van alle stakeholders voldoet.

4. Resultaat

Het resultaat van de afstudeeropdracht is een berichtenservice welke aan de wensen en eisen van alle stakeholders voldoet. De berichtenservice moet in productie kunnen worden genomen. Het moet dus volledig getest zijn. Door de nieuwe berichtenservice zullen de horecaleveranciers door middel van pushberichten reclame berichten kunnen sturen die bestemd zijn voor een specifieke groep horecabedrijven. De berichtenservice zal gaan werken om de producten van de leveranciers te promoten.

5. Uit te voeren werkzaamheden, inclusief een globale fasering, mijlpalen en bijbehorende activiteiten

Hieronder een overzicht van de activiteiten die vanaf de startdatum worden ondernomen om de opdracht met succes uit te kunnen voeren. Met daarbij een globale schatting van het aantal dagen dat de activiteit zal kosten. De activiteiten zullen worden opgedeeld in SCRUM sprints welke 2 tot 3 weken zullen duren. Aan het eind van de sprint zal er een werkend stuk van de software worden opgeleverd. De sprint specificatie is niet uitputtend en zal gedurende het project af kunnen wijken. Gedurende het project zal er elke week aan het afstudeerdossier gewerkt. Deze wordt aan het einde van het project helemaal in orde gemaakt.

Opzet scrum project Afstudeerweek 1;

In de eerste week zal volledig gericht worden voor het opzetten van het SCRUM project en het inlezen van Jira welke SCRUM ondersteunt. Ook zal ik deze week gebruiken om het bedrijf beter te leren kennen.

Ontwerp Afstudeerweek 2 & 3;

In de eerste sprint zal er een ontwerp worden gemaakt voor de app en online versie m.b.v. UML Ook zullen de requirements volledig gespecificeerd moeten zijn aan het einde van deze sprint. Ook zal het testplan opgezet worden

Sprint 1: Afstudeerweek 4 & 5;

Aan het einde van sprint 2 wordt de eerste versie van de online versie opgeleverd. Met deze versie is het mogelijk om korte reclame berichten en blog berichten te posten via de internet versie.

Sprint 2: Afstudeerweek 6 & 7;

De berichten kunnen via Facebook gepost worden.

Sprint 3: Afstudeerweek 8 & 9;

De app kan berichten versturen en ontvangen

Sprint 4: Afstudeerweek 10 & 11 & 12;

De app werkt op alle platformen d.m.v. push berichten.

Sprint 5: Afstudeerweek 13, 14 & 15;

De berichten moeten kunnen worden gebruikt om de aanbiedingen te verwerken in de orders.

Afronding afstudeerstage Afstudeerweek 16, 17

De laatste twee weken van het afstudeertraject zal volledig worden gericht op het afronden van het afstuderen en het opbouwen van het afstudeerdossier.

Op te leveren (tussen)producten

De volgende (tussen)producten zullen worden opgeleverd voor het bedrijf:

- *Plan van aanpak*
Het plan van aanpak zal opgenomen worden hoe ik dit project zal uitvoeren. Deze zal in de eerste week van het project opgeleverd worden.
- *Verslag literatuuronderzoek*
Het verslag voor het literatuuronderzoek zal gaandeweg het gehele project worden aangepast. Bij het begin van elke sprint zal er een literatuuronderzoek plaatsvinden. Alle gevonden informatie zal ik in het verslag opnemen.
- *Het ontwerp van de berichtenservice*
Aan het begin van het project zal er een ontwerp worden gemaakt. Het ontwerp zal worden gemaakt d.m.v UML
- *Deployment diagram*
Het deployment diagram zal gebruikt worden om beter inzicht te tonen in het algehele project.
- *Berichtenservice code*
De code die wordt gemaakt zal een product zijn van het project
- *Testplan*
Het testplan zal worden opgesteld aan het begin van het project. Elke sprint zal er getest worden, dit zal worden opgenomen in het testplan.
- *Testrapportage*
De uitkomsten van de testen zullen in een document worden bijgehouden. Om voor het hele systeem een testrapportage te hebben
- *Implementatieplan*
Hoe de implementatie zal verlopen wordt opgenomen in het implementatieplan

6. Te demonstreren competenties en wijze waarop

- 3.2 Ontwerp systeemdeel.

De complexiteit is lastig en zal zelfstandig worden uitgevoerd. Het niveau van deze beroepstaak niveau 3.

De berichtenservice wordt een objectgeoriënteerde applicatie, in de toekomst moet rekening worden gehouden met toekomstige wijzigingen, testbaarheid en hergebruik. Er zal gebruik gemaakt worden van een ontwerp methodiek en een tool.

- 3.3 Bouwen van applicatie.

De complexiteit is lastig en zal zelfstandig worden uitgevoerd. Het niveau van deze beroepstaak niveau 3.

De berichtenservice wordt een objectgeoriënteerde applicatie, waarbij geavanceerde concepten van de gebruikte programmeertaal aan de orde komen. Verder wordt rekening gehouden met toekomstige wijzigingen, testbaarheid en hergebruik. Het bouwen gebeurt in een ontwikkelomgeving.

- 3.5 Uitvoeren van en rapporteren over het testproces

De complexiteit is lastig en zal zelfstandig worden uitgevoerd. Het niveau van deze beroepstaak niveau 3.

Bij het opstellen van het logisch testontwerp wordt gebruik gemaakt van een testontwerptechniek. Er is aandacht voor herhaalbaarheid van de testen. Het betreft hoofdzakelijk het testen van de functionaliteit. Testrapportage betreft het volledige systeem.

Bijlage 2. Plan van Aanpak

Plan van aanpak

Ontwikkelen berichtenservice

in **One**

Student: Marcel Fasseur
Studentnr: 10069968
Onderwijsinstelling: De Haagse Hogeschool
Opleiding: HBO Informatica

Studieloopbaanbegeleider: Mevr. H.G.J. Bechet-Tjoonk
Begeleidend examiner: Dhr. G.A. Mijnaerends
Tweede examiner: Mevr. G.E. in 't Veld

Datum: 10 January 2014
Versie: 1.0
Status: Final

Versiebeheer

Versie	Wijziging(en)	Datum
0.1	Opstellen document; Aantal hoofdstukken ingevuld.	20-08-2013
0.2	Spellingscontrole, Feedback verwerkt	21-08-2013
0.3	Plan uitgewerkt, Hoofdstuk product herschreven	22-08-2013
0.4	Plan afgemaakt, Status veranderd naar "klaar voor revisie"	23-08-2013
0.5	Verbetering aan software ontwikkelmethode, begrippenlijst.	26-08-2013
1.0	Feedback verwerkt van gesprek, Status veranderd naar "Final"	28-08-2013

Inhoudsopgave

Versiebeheer	1
Inhoudsopgave	2
Inleiding	3
1 Organisatie	4
1.1 Missie	5
1.2 Processen	5
2 De opdracht.....	6
2.1 Probleemstelling	6
2.1.1 Oude situatie	6
2.1.2 Nieuwe situatie.....	6
2.2 Doelstelling	9
2.3 Resultaat	9
2.4 Producten.....	9
2.5 Stakeholders	9
3 Risico's	11
4 Randvoorwaarden.....	12
5 Beperkingen	12
6 Software ontwikkelmethode.....	13
7 Planning	14
8 Begrippenlijst	15
8.1 Stipulatieve betekenissen	15
9 Bronnenlijst	16

Inleiding

Dit is het plan van aanpak over de berichtenservice die ontwikkeld wordt voor inONE en is geschreven door Marcel Fasseur. Dit document is geschreven naar aanleiding van een aantal gesprekken met de opdrachtgever inONE te Leiden.

Dit plan is gemaakt om de berichtenservice te kunnen ontwikkelen en inzicht te geven in hoe het ontwikkeld zal gaan worden. Naast dat dit PvA een basis legt voor de ontwikkeling is het voor inONE een document waar inzicht in wordt gegeven hoe het ontwikkel proces vorm is gegeven en hoe het wordt ontwikkeld. Dit is dus een grondslag voor het gehele project.

In het eerste hoofdstuk wordt er ingegaan op de organisatie hoe deze in elkaar steekt en wat inONE doet. In hoofdstuk 2 staat beschreven wat de opdracht inhoudt wat de doelstellingen zijn, de stakeholders en weke producten er worden opgeleverd. Hoofdstuk 3 staan de risico's die gelden voor dit project. In hoofdstuk 4 staan de randvoorwaarden beschreven en in hoofdstuk 5 staan de beperkingen van het project. In hoofdstuk 6 wordt er in gegaan op de software ontwikkelmethode welke gebruikt gaat worden voor dit project. hoofdstuk 7 en 8 staan de planning voor het project en tot slot de begrippenlijst, de bronnenlijst is hoofdstuk 9.

1 Organisatie

inONE is een software bedrijf gevestigd in Leiden. Ze maken eenvoudige e-commerce software voor horecaleveranciers en handige tools voor de Horeca. inONE is een klein en jong bedrijf en bestaat sinds 2004. Ze opereert vanaf 1 locatie in Leiden. Er werken op dit moment 9 medewerkers en 1 stagiair. De afdeling waar dit project word uitgevoerd (de ontwikkelafdeling) bestaat uit 3 medewerkers exclusief de stagiair.

Er zijn vier producten ontwikkeld bij inONE te noemen Leveranciers, Bestelbox, Tools en Menumanager. Bij het in gebruik nemen van het systeem wordt er bij alle klanten een speciale training gegeven zodat zij zonder problemen online kunne bestellen. Ook bij eventuele latere vragen is er klantenondersteuning, als een specifiek verlengstuk van de service. Alle producten van inONE worden aangeboden d.m.v. het SAAS principe. De producten hieronder staat beschreven wat de tools inhouden:

Leveranciers

In deze webwinkel staan alle producten van een leverancier op een rij. De horecaklanten kunnen de producten bestellen via een afgeschermd en beveiligde online omgeving of via de bestelbox App. De webwinkel is een schakel tussen hun huidige softwarepakket en de webwinkel. Het huidige pakket wordt aangesloten aan het bestelbox hierdoor hoeft het software pakket van de leverancier niet aangepast te worden.

bestellen

Met bestellen is het mogelijk dat de horecaonderneming via internet en de app rechtstreeks bij de vaste leverancier die zijn aangesloten bij inONE een bestelling kunnen plaatsen. Hierdoor hebben de. Bestellen wordt aangeboden door de leverancier, er zijn dus geen kosten aan verbonden voor horeca ondernemers.

Tools

Wanneer er met Bestelbox wordt gewerkt, kan er gebruik worden gemaakt van inONE Tools. Een pakket van programma's voor de inventarisatie, controle en analyse van de voorraad, inkoop, verkoop, goederenontvangst en facturen.

inONE Tools geeft toegang tot de modules:

- Voorraad: inventarisatie en waarde van de voorraad.
- Goederenontvangst: komen het geleverde en het bestelde overeen?
- Facturen: alle facturen overzichtelijk bij elkaar en gemakkelijk te koppelen aan de grootboeknummers en boekhoudpakket van de horeca ondernemer
- Analyse: maak een analyse op basis van factuurgegevens, voorraad, goederenontvangst en inkoop/verkoop.
- Mail bestellen: handige tool als een van de leveranciers nog niet bij inONE is aangesloten.
- Import en export van bestellingen die in het eigen systeem zijn ingevoerd van de horeca ondernemer.

Menumanager

Een tool voor online receptenbeheer, menu engineering en kostprijsberekeningen. Deze tool is op het moment van schrijven nog in ontwikkeling.

1.1 Missie

De missie van het inONE is als volgt: “Dat elke horecaonderneming/keuken in Nederland de mogelijkheid heeft om via één punt, Bestelbox, bij al zijn eigen leveranciers kan bestellen.”

1.2 Processen

Het primaire proces van inONE is het aanbieden en onderhouden van een software pakket voor leveranciers en zijn klanten (horecaonderneming/keuken) De software moet ervoor zorgen dat de klanten via één punt, bestelbox, zijn bestellingen kan plaatsen.

2 De opdracht

2.1 Probleemstelling

Hoe kunnen de leveranciers die samenwerken met inONE op een afgesproken manier berichten versturen naar specifieke horecabedrijven via zowel het mobiele platform als online versie?

2.1.1 Oude situatie

In de huidige situatie kunnen blog berichten worden verstuurd via Bestelbox. Deze blog berichten kunnen worden gelezen en verstuurd en er is weinig interactie behalve een link naar een pagina. De berichten kunnen naar specifieke horecabedrijven worden verzonden of naar allemaal. Met de huidige bestelbox app kunnen bestellingen geplaatst worden.

2.1.2 Nieuwe situatie

In de nieuwe situatie met het mogelijk zijn om een aantal type berichten te versturen namelijk primair, sociaal, aanbiedingen en mededelingen meer over de verschillende types is later te lezen in dit hoofdstuk. De berichten moeten zowel via het mobiele platform (De inONE Bestelbox app) als via de online versie verstuurd en ontvangen kunnen worden. Er moet nog wel gekeken worden hoe er onderscheid tussen de verschillende types wordt gemaakt. De berichten moeten naar 1 of meerdere klanten verstuurd kunnen worden. De berichtenservice moet daarom een verbinding hebben met de “groepen indeling” van Bestelbox. Er zal voor zowel de mobiele versie als de online versie 1 en dezelfde programma moeten worden. Er moet dus goed gelet worden op de usability dat deze voor beide geschikt is. De berichten die in het huidige mededelingenboard staat zullen niet worden meegenomen maar zal langs de nieuwe berichtenservice erbij blijven staan voor een bepaalde tijd (nog nader te bepalen).

Er kunnen hiervoor een aantal typen leesrechten onderscheiden worden:

Leverancier kan:

- Een bericht voor al zijn klanten maken.
- Een bericht voor 1 of enkele van zijn klanten maken.

Horeca (manager) kan:

- Een bericht voor alle locaties maken.
- Een bericht voor 1 of enkele locaties maken.
- Een bericht voor zijn leveranciers maken

inONE kan:

- Een bericht voor alle leveranciers maken.
- Een bericht voor alle horeca maken.
- Een bericht voor 1 of enkele leveranciers maken.
- Een bericht voor 1 of enkele horeca locaties maken.

Ook zal er de mogelijkheid zijn om te reageren op berichten die zichtbaar zijn voor de gebruiker om meer interactie te krijgen op deze manier.

Er zijn een aantal kanalen waar de berichten binnen zullen komen. Dit kan zijn via de portal van inONE via sociale media of een RSS feed. De berichten zullen worden getoond in een soort container. In deze container moet er een mogelijkheid komen om berichten te filteren. De berichten filteren is een belangrijk onderdeel van de service. Doordat er veel verschillende kanalen worden gebruikt voor berichten moet er goed gekeken worden hoe het te voorkomen wordt dat het niet een overvolle service wordt met te veel onzin waardoor er ergernis zou kunnen ontstaan. Om het via verschillende kanalen binnen te halen zal gekeken moeten worden of dit zelf geprogrammeerd moet worden of dat er mogelijk een tool bestaat waarbij dat mogelijk is.

De inONE Bestelbox app moet via push berichten kunnen ontvangen en in de statusbalk van het mobiele platform weergegeven worden. Zodra er op de notificatie wordt geklikt moet de app tevoorschijn komen. Als het gaat om een reclame bericht zal in de app zal de bestelling geregeld kunnen worden en zal de korting die in het bericht staat verrekend worden bij de bestelling. Dit onderdeel kan complex worden omdat iedere mobiele platform anders zal werken waardoor er veel uitgezocht zal moeten worden

Hieronder in het kort wat de verschillende types inhouden:

Primaire berichten

De primaire berichten zullen voornamelijk door inONE gebruikt worden en zijn de urgente berichten zoals storingen of updates. Het type primair zal altijd getoond worden via het mededelingenboard

Sociale berichten

Berichten moeten via social media gepost kunnen worden. De berichten moeten worden opgepakt en getoond worden aan alleen de klanten van die leverancier. Dit is nog erg onduidelijk hoe het wordt geïmplementeerd omdat er mogelijk te veel berichten binnen komen als een klant meerdere leveranciers hebt en deze vrij veel posten.

Aanbiedingen

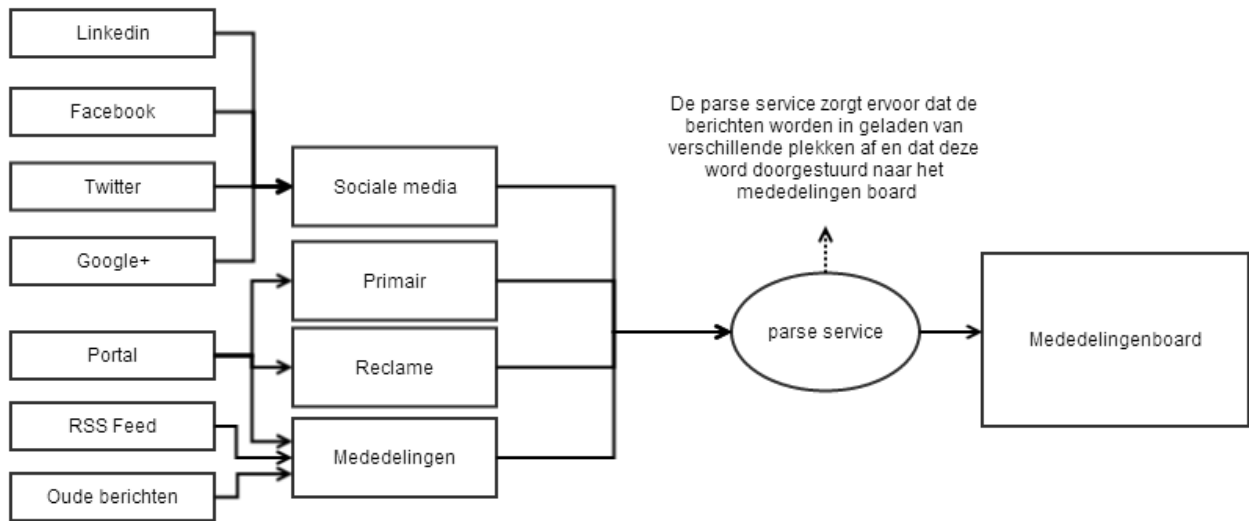
In de nieuwe situatie moet het mogelijk zijn om korte reclame berichten sturen ter illustratie: “100kg Zalm in de aanbieding voor €80, -“. De reclame berichten moeten gekoppeld worden aan Bestelbox zodat de korting die in het bericht staat verrekend wordt tijdens het bestellen. Dit zal een aanpassing worden aan het huidige systeem.

Mededelingen

De mededelingen zijn de algemenere berichten zoals bijvoorbeeld de nieuwsbrieven van de leveranciers en blog berichten van inONE.

Er zijn 2 onderdelen gewenst om in het systeem erbij te hebben. Met het ontwikkelen van het systeem moet hier rekeningen mee worden gehouden. het betreft het bijhouden van statistieken over de berichten (bijvoorbeeld ...x gelezen) of het aanmaken van custom groepen voor leveranciers. Ook moet er rekeningen worden gehouden met windows mobile voor in de toekomst.

Dit is een kort concept schema met de werking van de verschillende kanalen en types.



2.2 Doelstelling

De doelstelling van het project is er om er voor te zorgen dat inONE een werkende berichtenservice krijgt die aan de wensen en eisen van alle stakeholders voldoen. Voor het einde van de afstudeerperiode moet de berichtenservice worden opgeleverd en volledig getest zijn.

2.3 Resultaat

Het resultaat van de afstudeeropdracht is een berichtenservice welke aan de wensen en eisen van alle stakeholders voldoen. De berichtenservice moet in productie kunnen worden genomen. Het moet dus volledig getest zijn. Door de nieuwe berichtenservice zullen de horecaleveranciers door middel van pushberichten reclame berichten kunnen sturen die bestemd zijn voor een specifieke groep horecabedrijven. De berichtenservice zal gaan werken om de producten van de leveranciers te promoten. en de communicatie via de berichtenservice te laten lopen

2.4 Producten

De volgende (tussen)producten zullen worden opgeleverd voor het bedrijf tijdens het ontwikkelen van de berichtenservice:

- **Plan van aanpak**
Het plan van aanpak zal opgenomen worden hoe ik dit project zal uitvoeren. Deze zal in de eerste week van het project opgeleverd worden.
- **Het ontwerp van de berichtenservice**
Aan het begin van het project zal er een ontwerp worden gemaakt. Het ontwerp zal worden gemaakt d.m.v. UML in het programma Gliffy.
- **Deployment diagram**
Het deployment diagram zal gebruikt worden om beter inzicht te tonen in het algehele project.
- **Berichtenservice code**
De code die wordt gemaakt zal een product zijn van het project
- **Testplan**
Het testplan zal worden opgesteld aan het begin van het project. Elke sprint zal er getest worden, dit zal worden opgenomen in het testplan.
- **Testrapportage**
De uitkomsten van de testen zullen in een document worden bijgehouden. Om voor het hele systeem een testrapportage te hebben

2.5 Stakeholders

Er zijn verschillende doelgroepen welke zijn eigen eisen en wensen hebben hieronder staat een lijst van de doelgroepen welke te maken hebben met het berichtenservice en op welke manier:

Leveranciers

Een leverancier moet berichten kunnen beheren/versturen naar al zijn klanten of naar 1 of enkele klanten

Horeca Manager

Een horeca manager moet berichten kunnen beheren/versturen naar al zijn locaties of naar 1 of enkele locaties.

inONE

inONE moet berichten kunnen beheren/versturen naar alle leveranciers/horeca of 1 of enkele leveranciers/horeca

Er moet goed geluisterd worden naar alle doelgroepen om zo de eisen en wensen goed in kaart te brengen. inONE zal de leveranciers en horeca managers vertegenwoordigen om de wensen en eisen vast te stellen.

3 Risico's

In dit hoofdstuk wordt beschreven welke risico's bekend zijn bij aanvang van dit project.

- De app-stores kan te hoge eisen stellen aan de app om de applicatie te kunnen publiceren. Hierdoor kan de app niet worden toegelaten in de app-stores, om dit op te vangen moeten de eisen van de app-stores tijdig worden onderzocht en deze naast de eisen en wensen leggen zodat dit tijdig kan worden opgevangen.
- Beter inzicht in het project kan leiden tot veranderingen in de requirements. uiteindelijk kan hierdoor vertraging van het project ontstaan wanneer er grote veranderingen doorgevoerd moeten worden. Daarom moet er tijdens het project veel overleg zijn met de opdrachtgever zodat dit tijdig opgevangen kan worden en de veranderingen van de requirements niet achteraf komen.
- Tijdens het project kunnen er meer requirements bijkomen waardoor het project te groot wordt voor de beschikbare tijd.
- Wanneer besluiten (te) traag worden genomen kan dit tot vertraging van het project leiden. Op het moment dat besluiten niet snel genoeg worden genomen moet er overleg zijn met de opdrachtgever hoe dit probleem opgelost kan worden
- Miscommunicatie met de stakeholders kan ervoor zorgen dat de requirements verkeerd zijn opgesteld of dat er ergernissen ontstaan. Hiervoor moet worden gewaakt. Door veel overleg te hebben met de stakeholders kan dit snel en tijdig worden opgevangen.

4 Randvoorwaarden

Het project omvat de volgende randvoorwaarden:

- Voor het ontwikkeling is een Android toestel en een iOS toestel noodzakelijk om te kunnen testen dit mag een mobiel of tablet zijn.
- Er moet een Apple developer licentie aanwezig zijn.
- Er moet genoeg tijd vrij gemaakt worden door de bedrijfsmentor / andere werknemers van inONE voor de begeleiding van het project

5 Beperkingen

In dit hoofdstuk staat beschreven wat de beperkingen van het project zijn.

- Het onderhoud van de berichtenservice valt buiten de scope van dit project.
- Het GUI ontwerp en de usability is een onderdeel van de opdracht. De kleurtjes en plaatjes daar in tegen wordt gedaan door de designer van inONE.
- Afspraken naast het project: Het kan voorkomen dat er naast het project afspraken staan ingepland waardoor projecttaken mogelijk stil komen te liggen. Hier zal rekening mee gehouden moeten worden in de planning. Omdat de afspraken later in het project gemaakt zouden kunnen worden moet de planning met voortschrijdend inzicht gemaakt worden en waar nodig aangepast worden.
- Omdat SCRUM vereist dat er veel overleg is met de stakeholders zullen er tijdig afspraken gemaakt moeten worden zodat het project geen vertraging op loopt. Wanneer door omstandigheden toch vertraging wordt opgelopen zal de planning bijgesteld moeten worden.
- Afspraken op school zullen tussendoor voor kunnen komen. Het zou dus kunnen zijn dat het werk een dag of een deel van de dag stil komt te liggen.

6 Software ontwikkelmethode

De software ontwikkelmethode die zal worden gebruikt is SCRUM. SCRUM is een framework voor agile management van softwareontwikkeling. Omdat de projecten die worden uitgevoerd bij inONE worden gemanaged met deze ontwikkelmethode is de keuze hierop gevallen. Voor het project zullen daarom de daarbij behorende documenten worden opgeleverd zoals de product backlog voor alle functionele en niet functionele eisen, de sprint backlog voor de eisen van de sprint, de sprint burndown chart om de voortgang goed bij te houden en optioneel een impediment list. Ook zal er een Definition of "Done" worden aangemaakt.

Elke sprint zal een vaste lengte van 2 weken hebben. Aan het eind van elke sprint zal een deel opgeleverd worden als een prototype. Er zal niet worden gewerkt in teamverband maar door alles goed vast te leggen kan wel een goed overzicht gehouden worden over het werk door de bedrijfsmentor en de opdrachtgever. De ontwikkeling zal alleen door Marcel Fasseur worden gedaan maar mochten er vragen zijn of uitleg nodig is hier altijd de mogelijkheid voor bij de bedrijfsmentor en overige programmeurs. Voor elke spint zal aan het begin vast gesteld worden welke onderdelen er behandeld gaan worden. De requirements worden vastgelegd in de product backlog in de vorm van user stories. De onderdelen die behandeld zullen gaan worden per sprint wordt vastgelegd in de sprint backlog. Ook wordt er een sprint burndown chart bij gehouden per sprint hierin wordt alles goed gedocumenteerd wat de voortgang is van de sprint. Wanneer er een blokkade wordt geconstateerd zal het worden vastgelegd in het impediment list en zal de stand van de oplossing hiervoor worden bijgehouden. De Definition of "Done" wordt bijgehouden zodat er geen onduidelijkheden kunnen ontstaan zodra een user story gemarkeerd staat als klaar hierin staat wat "klaar" precies inhoudt.

Als ondersteuning van SCRUM wordt het SAAS pakket JIRA gebruikt. JIRA is een project Alle documentatie en gemaakte beslissingen zullen hier in opgenomen worden. Ook zit er een module in om modellen en diagrammen te maken. Om deze reden zal ik ook gebruik gaan maken van deze module om het makkelijker te maken.

7 Planning

In deze planning staat aangegeven wanneer welke onderdelen worden uitgevoerd. Ook staan de afspraken met school zodat hier rekening mee gehouden kan worden. Elke week zal er een halve dag besteed worden aan het afstudeerdossier. De laatste twee weken van de afstudeerperiode zullen gebruikt worden voor het afstudeerdossier

Week	Sprint	Taken
1		<ul style="list-style-type: none"> - SCRUM Project opzetten. - Inlezen in JIRA. - Inlezen in XPages. - Plan van aanpak maken.
2		<ul style="list-style-type: none"> - Ontwerp maken voor de berichtenservice m.b.v. UML. - Eisen en wensen specificeren. - Testplan opzetten.
3	1	<ul style="list-style-type: none"> - Eerste prototype beginnen. Met deze versie moeten er berichten kunnen worden gemanaged
4		<ul style="list-style-type: none"> - Eerste prototype opleveren - Voor de week erop volgend afspraak plannen met begeleidende examiner
5	2	<ul style="list-style-type: none"> - De verschillende types opzetten - Uitzoeken hoe onderscheid te maken van de verschillende types
6		<ul style="list-style-type: none"> - Uitwerken van "rechten" systeem als beschreven in het PvA
7	3	<ul style="list-style-type: none"> - Alle kanalen koppelen aan berichtenservice (m.u.v socials) - Voor de week erop volgend een voortgangsverslag opstellen en versturen
8		<ul style="list-style-type: none"> - Afmaken van koppelingen van kanalen
9	4	<ul style="list-style-type: none"> - Push berichten Android ontwikkelen
10		<ul style="list-style-type: none"> - Push berichten iOS ontwikkelen
11	5	<ul style="list-style-type: none"> - De berichten moeten kunnen worden gebruikt om de aanbiedingen te verwerken in de orders - Bespreking van het concept afstudeerdossier
12		<ul style="list-style-type: none"> -
13	6	<ul style="list-style-type: none"> - Social media kanalen koppelen
14		<ul style="list-style-type: none"> - Laatste versie opleveren en volledig getest
15	Uitloop	<ul style="list-style-type: none"> - Deze week is ingepland voor mogelijke uitloop van het project - Tussentijds assessment
16		<ul style="list-style-type: none"> - Afstudeerdossier
17		<ul style="list-style-type: none"> - 10 januari laatste stage dag - 11 januari uiterste inleverdatum afstudeerdossier

8 Begrippenlijst

SAAS = Software as a service.

Gliffy = Online diagram en flowchart software welke in JIRA beschikbaar is via de Gliffy Confluence Plugin.

Mock-up = Een prototype van het design op ware grote.

8.1 Stipulatieve betekenissen

App stores = Apple store, Google Store.

Alle platformen = Android / iOS.

Social media = Facebook, Twitter, Google+, LinkedIn.

9 Bronnenlijst

Notes in 9. (2012). Xpages jumpstart. Geraadpleegd op <http://notesin9.com/index.php/xpages-jumpstart/>

Atlassian (2013). Issue & Project Tracking Software | Atlassian Geraadpleegd op <https://www.atlassian.com>

Donnelly M., McGucking T., Wallace M. (2011). Mastering XPages: A Step-by-Step Guide to XPages Application Development and the XSP Language. Geraadpleegd op <http://www.redbooks.ibm.com/Redbooks.nsf/ibmpressisbn/9780132486316>

inONE. (z.j.). eCommerce software voor horecagroothandels | inONE.nl. Geraadpleegd op <http://inone.nl/>
International Business Machines Corporation. (z.j.). IBM Notes and Domino Application Development wiki. Geraadpleegd op <http://www-10.lotus.com/ldd/ddwiki.nsf/dx/tutorial-intro-to-xpages.htm>

openntf inc. (z.j.). Xpages. Geraadpleegd op <http://xpagesblog.com>

Schwaber K., Sutherland J. (2011). The Scrum Guide - The Definitive Guide to Scrum: The Rules of the Game. Geraadpleegd op <http://www.verheulconsultants.nl/Scrum%20Guide%20-%202011.pdf>

Bijlage 3. Product backlog

Doelen

- Berichten service ontwikkelen

Aannames

- De gebruikers weten hoe de app werkt.

Niet-functionele eisen

De berichtenservice moet een verbinding krijgen met de groepen indeling van bestelbox

Er moet 1 versie komen voor alle platformen

Requirements

#	User Story Beschrijving	Prioriteit	Opmerkingen
1	Als een leverancier wil ik een bericht kunnen sturen naar 1 of meer klanten.	Critical	
2	Als een horecamanager wil ik een bericht kunnen sturen naar 1 of meer locaties.	Critical	
3	Als inONE wil ik een bericht kunnen sturen naar 1 of meerdere leveranciers/horecamanagers	Critical	
4	Als een Stakeholder wil ik een viertal types in kunnen stellen:Primair, Sociaal, Aanbiedingen en mededelingen	Critical	
5	Als inONE wil ik het type Primair gebruiken om belangrijke berichten te versturen zoals storingen of updates.	Critical	
6	Als een leverancier wil ik mijn sociale media berichten automatisch laten doorsturen naar de berichtenservice.	Critical	
7	Als een Stakeholder wil ik een filter mogelijkheid hebben voor de berichten zodat ik niet wordt overladen met berichten	Critical	
8	Als een bestelbox app gebruiker wil ik notificaties krijgen van belangrijke berichten op mijn mobiel apparaat en in de statusbalk van het mobiele platform weergegeven worden.	Critical	
9	Als een Stakeholder wil ik instellingen voor filters eenmalig in hoeven stellen, maar ook later kunnen wijzigen	Critical	
10	Als een Stakeholder wil ik berichten via een online versie of een mobiele versie kunnen versturen en ontvangen	Critical	Rekening houden met usability
11	Als een Stakeholder wil ik berichten ontvangen via verschillende kanalen zoals social media, een RSS feed of via de inONE portal	Critical	
12	Als een Stakeholder wil ik alle berichten in een "container" kunnen zien.	Critical	Onder container word verstaan dat daar alle berichten in worden getoond als een "grote bak"

13	Als een Stakeholder wil ik kunnen klikken op een notificatie en op die manier de app openen met het bericht. Bij een reclamebericht wil ik dat het bestelscherm opent	Major	
14	Als een Stakeholder wil ik wanneer het bestelscherm opent gelijk de aanbieding naar voren komt als ik op een aanbieding geklikt heb,	Minor	
15	Als een Stakeholder wil ik persoonlijke berichten kunnen sturen. Dit zijn simpele tekstuele berichten.	Major	
16	Als een Stakeholder wil ik een email notificatie krijgen wanneer ik een persoonlijk bericht ontvangen heb.	Major	
17	Als een Stakeholder wil ik statistieken inzien van mijn berichten zoals bijvoorbeeld hoe vaak een bericht gelezen is.	Minor	
18	Als een Stakeholder wil ik kunnen reageren op berichten	Major	
19	Als een Stakeholder wil ik een status van een persoonlijk bericht in kunnen stellen.	Minor	
20	Als een Stakeholder wil ik dat er de mogelijkheid is voor meertaligheid	Trivial	Voor korte naamnotatie van talen gebruik maken van ISO 639-1

Vragen

Hieronder een aantal vragen die behandeld moeten worden aan de hand van de requirements

Question	Outcome
Hoe zorgen we ervoor dat de gebruikers niet worden overladen met berichten	Zorgen dat de filters op een juiste manier werken die 1 keer ingesteld hoeven worden en later aangepast kunnen worden. waar gewenst
Hoe wordt er onderscheid gemaakt tussen de verschillende berichten	
Word het ophalen van verschillende kanalen zelf geprogrammeerd of word hier een "standaard" pakket voor gebruikt	Dit wordt zelf geprogrammeerd, de API's worden gebruikt van de verschillende social medias
Hoe worden berichten verstuurd als een gebruiker niet is ingelogd.	
Voor notificaties moet gekeken of dit zelf geprogrammeerd word of dat dit een standaard pakket word gebruikt.	Hier wordt zelf voor geprogrammeerd d.m.v GoogleCloudMessaging service en de Apple Push Notification Service
Wie kan de status van een persoonlijk bericht aanpassen. De maker, Degene die hem verstuurd of beide. moet een status bij gesloten nog aangepast kunnen worden of is dit final? moeten er berichten nog verstuurd kunnen worden in een gesloten thread?	
Hoe gaan we een PDF/URL weergeven in het overzicht? hier kan je niet makkelijk een "snippit" van maken.	

Niet doen

- Onderhoud van de berichtenservice is buiten de scope van het project.

Bijlage 4. Notulen

Notulen 15-10-2013 10:00

Aanwezig: Leon belt. Rogier Bruin, Marcel Fasseur, Gerard Mijnaerends.

Genotuleerd door: Marcel Fasseur

- Maak vergelijkingen met verschillende bestaande systemen voor de verdediging. wat is goed en wat onderscheidt zich van andere systemen.. waarom gebruik je niet een standaard systeem?.
- Beschrijf alle ideeën op en benoem deze. Al implementeer je ze niet wel beschrijven.
- Waarom is het SCRUM en hoe pas je het toe. Gebruik je geen waterval.
- Een deployment diagram moet je hebben. Door dit te doen wordt het duidelijk hoe het is opgebouwd voor butenstandaars.
- Leg uit wat xpages is. en welke tools je gebruikt. Hoe werkt het nou allemaal precies?
- Welke IDE gebruik je, leg dit uit.
- Een prototype is dit de kwaliteitseis van SCUM? of is dit anders?
- Leg het proces van in productie nemen uit. Wie doet wat en hoe gaat dit in zijn werk.
- Testen beroepstaak: Neem testtechnieken en speciale testen.. ga er dieper op in dan alleen kleine onderdelen te testen dmv alleen de GUI
- Leg uit dat je afspraken maakt en waar deze toe dienen. Een de feedback momenten beschrijven. Er zijn grote mijlpalen.
- Als je afwijkt van de theorie moet je dit benoemen. Waarom je afwijkt.
- Het verslag opzicht moet je snappen zonder achtergrond informatie te hebben. Het moet op zichzelf staan.
- Houdt notulen bij en neem dit op in je verslag als bijlage
- op 60% een concept verslag opleveren.
- De scriptie mag ongeveer 40 á 50 pagina's zijn.
- Benoem een aantal belangrijke backlog items. Niet allemaal dit wordt te uitgebreid.
- Als er iets mis gaat dit beschrijven of als je ergens tegen aan loopt dit opnemen in het rapport.
- In het voortgangsverslag kort benoemen waar ik ongeveer sta met mijn project.
- Op 80% een gesprek waarin de laatste keer feedback wordt gegeven. Zorg dat het verslag zo goed als af is!.

Genotuleerd door: Leon Belt.

Vragen van mij:

1. Bericht aanbieding ook in domino de artikelen markeren als aanbieding.
 2. Personal message >> de close button aangeven aan iedereen dat het bericht gesloten is ("de leverancier heeft het bericht gesloten").
-
- Opmerkingen:
 - Waarom thread>> uitleggen.
 - Extra mogelijkheden laten zien, deze hoeven niet gemaakt te worden maar wel aantonen dat je er rekening mee hebt gehouden (agile).
 - Waarom een sprint van 128 punten? Hoelang? Wat zit er in een sprint? Legenda van punten geven? Welke tools gebruik je (jira confluence etc).
 - Je hebt de sprints ingedeeld, hoe verhoudt dit met agile, je gebruikt waterval, waarom? Wat kan er achteraf nog wijzigen?
 - Design pattern omschrijven.
 - Laat zien wat je vooraf gedaan hebt, voordat je aan de eerste sprint bent begonnen.
 - Beschrijf de totstandkoming van je klassendiagram, deel deze op en verdedig waarom je lege tabellen zoals primary & Advertisement message leeg hebt gelaten.
 - Laat de complexiteit van de berichten service zien.
Laat zien dat je verder denkt in je applicatie (selecteer horeca 5 klanten en selecteer horeca uit 100 klanten.
 - Leg de gelaagdheid van je applicatie uit in het begin van je verslag, werk vanuit daar verder.
 - Leg de ontwikkelsuite uit en geef aan wat het verschil is met andere suites.

Notulen 21-8-2013 10:00

Aanwezig: Leon belt. Rogier Bruin, Marcel Fasseur

Genotuleerd door: Marcel Fasseur

Onderdelen

Inleiding

- Toevoegen aan de inleiding: --> nieuw hoofdstuk aangemaakt
 - wat is inONE.
 - welke producten er zijn
 - wat houden de producten in.
- In de inleiding staat vrij veel Plan of plan van aanpak. Zoek naar vervangende woorden
Voorbeeld vervangen: PvA Of dit document.

Probleemstelling

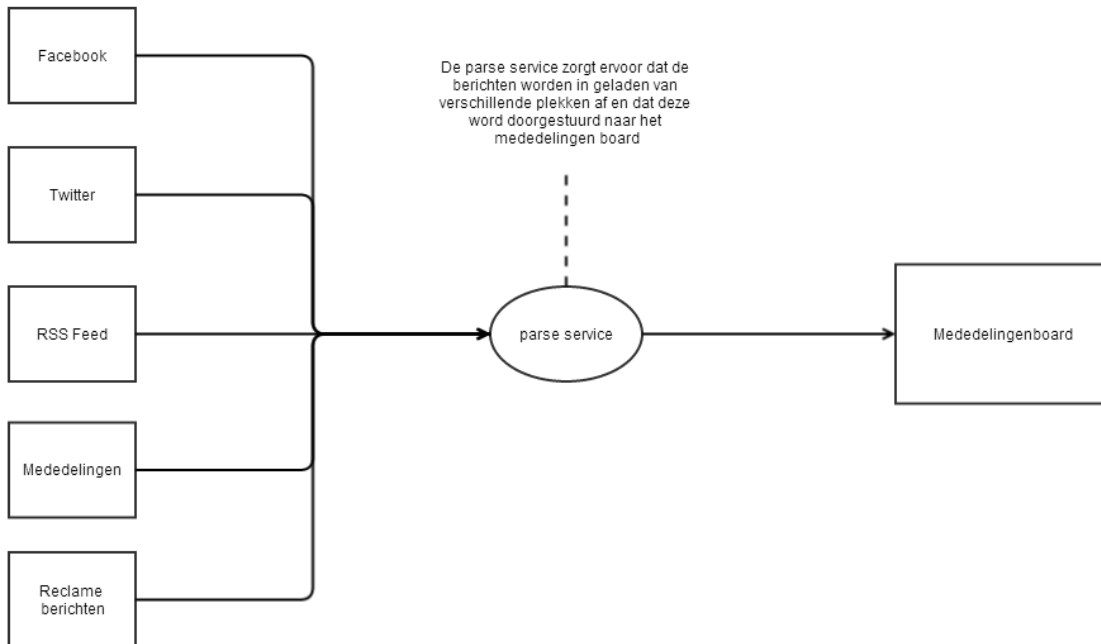
- Een probleemstelling is: Wat is het probleem. in welke omgeving.
 - Toevoegen aan probleemstelling
 - o Er worden te weinig klant gerichte berichten geplaatst door leveranciers. Dit gaat wel via FB maar niet via inONE
 - o Meer interactiviteit toevoegen.
 - o In klantengroep op een gestructureerde manier berichten versturen
 - o Social media meer eraan koppelen
 - o Kunnen we een bestaande tool gebruiken voor de koppeling voor verschillende kanalen of moet dit zelf geprogrammeerd worden
- Uit de probleemstelling halen wat er wordt gemaakt. Daar heeft de probleemstelling niks mee te maken hoofd probleemstelling maken met daaronder deel probleemstellingen.
 - onderste deel van de probleemstelling is geen onderdeel van de probleemstelling.

Wat is/wordt de berichten service

- Puur voor interne communicatie tussen leveranciers en zijn klanten, dus een Interne berichten service
- het huidige mededelingen board wordt vervangen door de berichtenservice
- Bij het versturen van berichten moet er een specificatie komen wat voor type bericht het is.
 - Primair --> altijd via mededelingen board --> Alleen voor inONE
 - Sociaal --> Berichten uit Facebook Twitter en andere social media --> Dit word geautomatiseerd
 - Aanbiedingen
 - Mededelingen
- Hoe moet er onderscheid gemaakt gaan worden onder de verschillende types
- waarheen het word verstuurd moet aangegeven kunnen worden
 - Naar de APP.
 - Alleen op het board (in de portal).
 - Het zal altijd op het board geplaatst worden.
- De app word een notificeer app er komt namelijk maar 1 keer de bouw, deze moet geschikt zijn voor zowel mobiel is web.

- Het board moet een container worden waar verschillende kanalen aan gelinkt zijn
- Berichten moeten gefilterd kunnen worden.
- Een vraag is: Hoe voorkom je dat het niet een overvolle service wordt met te veel onzin waardoor het tot ergernis zou kunnen leiden
- Er moet gekeken worden of er een tool wordt gebruikt om de gegevens in te laden. of word alles zelf geprogrammeerd

Een kleine schets wat een onderdeelje wordt van de service



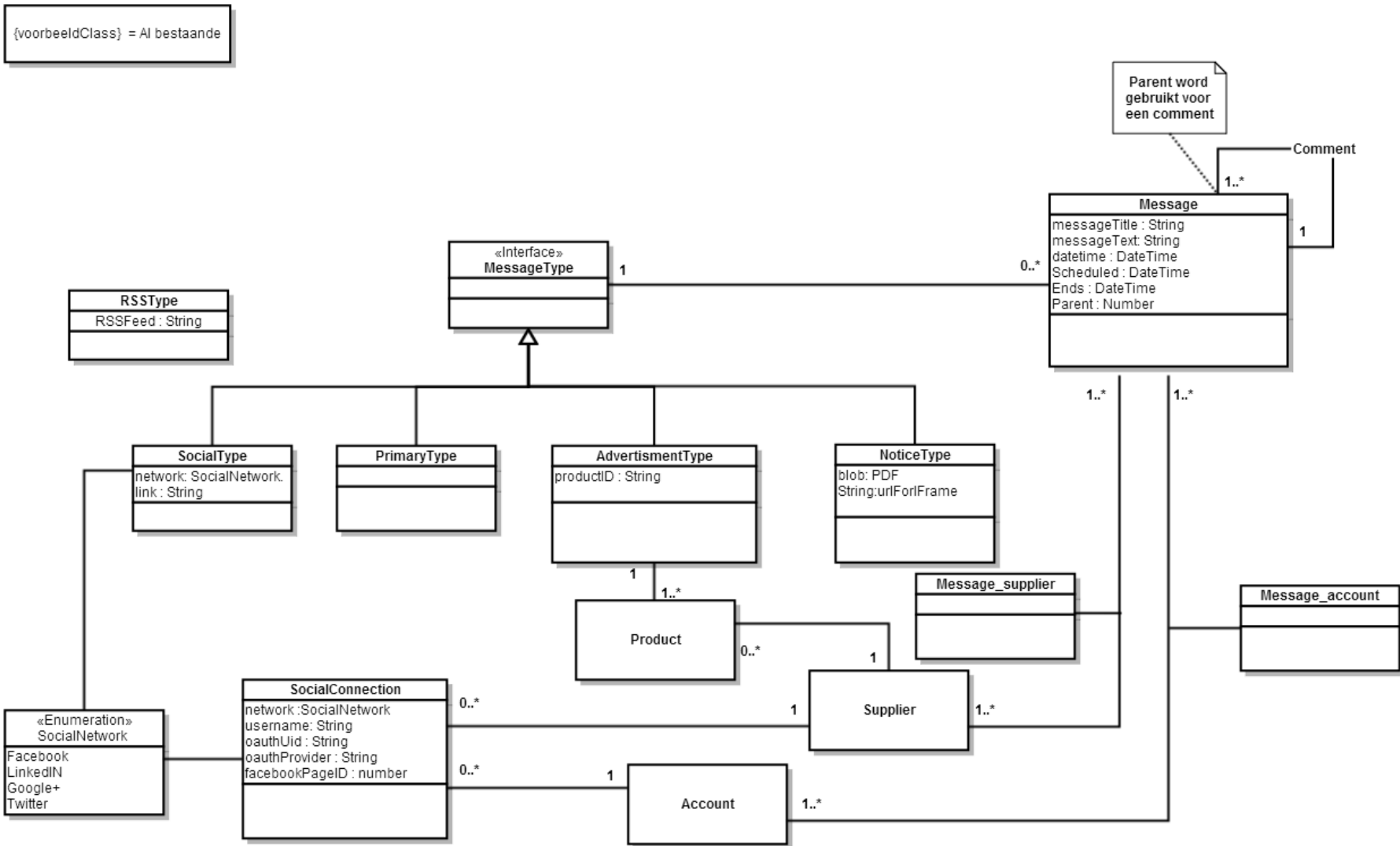
Notulen 27-8-2013 10:00

Aanwezig: Leon belt. Rogier Bruin, Marcel Fasseur

Genotuleerd door: Marcel Fasseur

- In mededelingen niet de oudere berichten meenemen. Dit moet aangepast worden in het PvA. De oude berichten worden als een "tweede" systeem ernaast gehouden en later weggegooid.
- In de inleiding staat hierna, en daarna. Dit veranderen naar hoofdstuk nummers.
- Het onderdeel van inONE minder commercieel maken en maar naar je eigen hand zetten.
- De namen van producten: Leveranciers word Bestelbox leverancier en Bestellen Bestelbox
- Vertel wat over de portal m.b.t. tot het project.
- De abonnementen kunnen verwijderd worden: Geeft geen meerwaarde
- Vertel dat inONE SAAS software ontwikkeld
- In de nieuwe situatie
 - o reacties versturen van leveranciers naar horecaondernemers en visa versa.
 - o Addon:
 - Statistieken bijhouden van wie wat leest etc.
 - Custom groepen aanmaken > Is dit gewenst bij de stakeholder leverancier?
- Omschrijven hoe de risico's getackeld worden
- windows mobile moet rekening mee gehouden worden
- Begin aan de UI d.m.v. mockups en kijk hoe het er uit moet komen te zien.
- Maak een horeca afspraak met Martijn
- Maak een leveranciers afspraak met Rogier.
- Kijken wat mogelijk is qua support en rick ondersteunen daarin.

Bijlage 5. Klasse iagram







Bijlage 6. SQL DDL

```
CREATE DATABASE messageservice;

use messageservice;

CREATE TABLE messagetype (
    messagetypeid INT(11) NOT NULL UNIQUE,
    messagedescription VARCHAR(255),
    PRIMARY KEY (messagetypeid)
);
INSERT INTO messagetype
VALUES      (1, "primarymessage"),
            (2, "advertisementmessage"),
            (3, "noticemessage"),
            (4, "socialmessage"),
            (5, "rssfeedmessage");
CREATE TABLE language (
    languageshort VARCHAR(2) NOT NULL,
    languagelong VARCHAR(100) NOT NULL,
    flag VARCHAR(255) NULL,
    PRIMARY KEY (languageshort)
);
CREATE TABLE message (
    messageid INT(11) NOT NULL auto_increment,
    messagetitle VARCHAR(255) NOT NULL,
    messagetext TEXT NOT NULL,
    messagedate DATETIME NOT NULL,
    scheduled DATETIME NULL,
    messageends DATETIME NULL,
    senderaccountid VARCHAR(6) NOT NULL,
    messagetypeid INT(2) NOT NULL,
    language varchar(2),
    PRIMARY KEY (messageid),
    FOREIGN KEY (senderaccountid)
        REFERENCES inone.account (custlogid),
    FOREIGN KEY (messagetypeid)
        REFERENCES messagetype (messagetypeid)
);
/*
readStatus
0 = unread
1 = read
*/
CREATE TABLE message_account (
    message_accountid INT(11) NOT NULL auto_increment,
    messageid INT(11) NOT NULL,
    receiveraccountid VARCHAR(6) NOT NULL,
    readstatus INT(2) NULL,
    readstatusdate DATETIME NULL,
    PRIMARY KEY (message_accountid),
    FOREIGN KEY (messageid)
        REFERENCES message (messageid),
    FOREIGN KEY (receiveraccountid)
        REFERENCES inone.account (custlogid)
);
CREATE TABLE messagecomment (
    messagecommentid INT(11) NOT NULL auto_increment,
    messageid INT(11) NOT NULL,
```

```

        messagecommenttext TEXT NOT NULL,
        messagecommentdate DATETIME NOT NULL,
        senderaccountid INT(11) NOT NULL,
        PRIMARY KEY (messagecommentid),
        FOREIGN KEY (messageid)
            REFERENCES message (messageid)
    );
CREATE TABLE noticemessage (
    noticemessageid INT(11) NOT NULL auto_increment,
    messageid INT(11) NOT NULL,
    pdfurl VARCHAR(255) NULL,
    urlforiframe VARCHAR(255) NULL,
    PRIMARY KEY (noticemessageid),
    FOREIGN KEY (messageid)
        REFERENCES message (messageid)
);
CREATE TABLE rssfeedmessage (
    rssfeedmessageid INT(11) NOT NULL auto_increment,
    messageid INT(11) NOT NULL,
    link VARCHAR(255) NOT NULL,
    guid VARCHAR(255) NOT NULL,
    comments VARCHAR(255) NULL,
    category VARCHAR(255) NULL,
    PRIMARY KEY (rssfeedmessageid),
    FOREIGN KEY (messageid)
        REFERENCES message (messageid)
);
CREATE TABLE advertisementmessage (
    advertisementmessageid INT(11) NOT NULL auto_increment,
    messageid INT(11) NOT NULL,
    PRIMARY KEY (advertisementmessageid),
    FOREIGN KEY (messageid)
        REFERENCES message (messageid)
);
CREATE TABLE advertisementmessage_product (
    advertisementmessage_productid INT(11) NOT NULL auto_increment,
    advertisementmessageid INT(11) NOT NULL,
    productid VARCHAR(40) NOT NULL,
    PRIMARY KEY (advertisementmessage_productid),
    FOREIGN KEY (productid)
        REFERENCES inone.product (productid)
);
CREATE TABLE primarymessage (
    primarymessageid INT(11) NOT NULL auto_increment,
    messageid INT(11) NOT NULL,
    PRIMARY KEY (primarymessageid),
    FOREIGN KEY (messageid)
        REFERENCES message (messageid)
);
CREATE TABLE socialnetwork (
    socialnetworkid INT(2) NOT NULL UNIQUE,
    networkname VARCHAR(255) NOT NULL,
    PRIMARY KEY (socialnetworkid)
);
INSERT INTO socialnetwork
VALUES
    (1, "facebook"),
    (2, "linkedin"),
    (3, "googleplus"),
    (4, "twitter");
CREATE TABLE socialmessage (
    socialmessageid INT(11) NOT NULL auto_increment,
    socialnetworkid INT(11) NOT NULL,
    link VARCHAR(255) NOT NULL,

```



```

        messageid INT(11) NOT NULL,
        PRIMARY KEY (socialmessageid),
        FOREIGN KEY (messageid)
            REFERENCES message (messageid),
        FOREIGN KEY (socialnetworkid)
            REFERENCES socialnetwork (socialnetworkid)
    );
CREATE TABLE socialconnection (
    socialconnectionid INT(11) NOT NULL auto_increment,
    socialnetwork INT(2) NOT NULL,
    username VARCHAR(255) NOT NULL,
    oauthuid VARCHAR(255) NOT NULL,
    oauthprovider VARCHAR(255) NOT NULL,
    defaultpublish BOOLEAN NOT NULL DEFAULT 0,
    accountid VARCHAR(6) NOT NULL,
    PRIMARY KEY (socialconnectionid),
    FOREIGN KEY (accountid)
        REFERENCES inone.account (custlogid)
);
CREATE TABLE personalmessagethread (
    personalmessagethreadid INT(11) NOT NULL auto_increment,
    senderaccountid VARCHAR(6) NOT NULL,
    receiveraccountid VARCHAR(6) NOT NULL,
    readstatus INT(2) NOT NULL,
    startdate DATETIME NOT NULL,
    closedate DATETIME NULL,
    PRIMARY KEY (personalmessagethreadid),
    FOREIGN KEY (senderaccountid)
        REFERENCES inone.account (custlogid),
    FOREIGN KEY (receiveraccountid)
        REFERENCES inone.account (custlogid)
);
CREATE TABLE personalmessage (
    personalmessageid INT(11) NOT NULL auto_increment,
    personalmessagethreadid INT(11) NOT NULL,
    messagetext VARCHAR(255) NOT NULL,
    personalmessagedate DATETIME NOT NULL,
    senderaccountid VARCHAR(6) NOT NULL,
    PRIMARY KEY (personalmessageid),
    FOREIGN KEY (personalmessagethreadid)
        REFERENCES personalmessagethread (personalmessagethreadid)
);
CREATE TABLE messageservicesettings (
    messageservicesettingsid INT(11) NOT NULL auto_increment,
    accountid VARCHAR(6) NOT NULL,
    rssfeedurl VARCHAR(255) NULL,
    showexpiredmessages BOOLEAN NOT NULL DEFAULT 0,
    sendmobilenotification BOOLEAN NOT NULL DEFAULT 1,
    PRIMARY KEY (messageservicesettingsid),
    FOREIGN KEY (accountid)
        REFERENCES inone.account (custlogid)
);
CREATE TABLE messageservicefilter (
    messageservicefilterid INT(11) NOT NULL auto_increment,
    messageservicesettingsid INT(11) NOT NULL,
    rangedate INT(11) NULL,
    PRIMARY KEY (messageservicefilterid),
    FOREIGN KEY (messageservicesettingsid)
        REFERENCES messageservicesettings (messageservicesettingsid)
);
CREATE TABLE messageservicefilter_messagetype (
    messageservicefilter_messagetypeid INT(11) NOT NULL auto_increment,
    messageservicefilterid INT(11) NOT NULL,

```

```

        messagetypeid INT(11) NOT NULL,
        PRIMARY KEY (messageservicefilter_messagetypeid),
        FOREIGN KEY (messageservicefilterid)
            REFERENCES messageservicefilter (messageservicefilterid),
        FOREIGN KEY (messagetypeid)
            REFERENCES messagetype (messagetypeid)
    );
CREATE TABLE messageservicefilter_language (
    messageservicefilter_languageid INT(11) NOT NULL auto_increment,
    messageservicefilterid INT(11) NOT NULL,
    languageshort VARCHAR(2) NOT NULL,
    PRIMARY KEY (messageservicefilter_languageid),
    FOREIGN KEY (messageservicefilterid)
        REFERENCES messageservicefilter (messageservicefilterid),
    FOREIGN KEY (languageshort)
        REFERENCES language (languageshort)
);
CREATE TABLE messageservicefilter_account (
    messageservicefilter_accountid INT(11) NOT NULL auto_increment,
    messageservicefilterid INT(11) NOT NULL,
    accountid VARCHAR(6) NOT NULL,
    PRIMARY KEY (messageservicefilter_accountid),
    FOREIGN KEY (messageservicefilterid)
        REFERENCES messageservicefilter (messageservicefilterid),
    FOREIGN KEY (accountid)
        REFERENCES inone.account (custlogid)
);
CREATE TABLE messageservicefilter_socialnetwork (
    messageservicefilter_socialnetwork INT(11) NOT NULL auto_increment,
    messageservicefilterid INT(11) NOT NULL,
    socialnetworkid INT(2) NOT NULL,
    PRIMARY KEY (messageservicefilter_socialnetwork),
    FOREIGN KEY (messageservicefilterid)
        REFERENCES messageservicefilter (messageservicefilterid),
    FOREIGN KEY (socialnetworkid)
        REFERENCES socialnetwork (socialnetworkid)
);
/*
mobileos
Android = 1
IOS = 2
*/
CREATE TABLE mobiledevice (
    mobiledeviceid INT(11) NOT NULL auto_increment,
    registrationid VARCHAR(255) NOT NULL,
    mobileos INT(2) NOT NULL,
    created DATETIME NOT NULL,
    accountid VARCHAR(6) NOT NULL,
    PRIMARY KEY (mobiledeviceid),
    FOREIGN KEY (accountid)
        REFERENCES inone.account (custlogid)
);
CREATE TABLE statistics (
    statisticsid INT(11) NOT NULL auto_increment,
    messagereadamount INT(11) NOT NULL default 0,
    messageid INT(11) NOT NULL,
    PRIMARY KEY (statisticsid),
    FOREIGN KEY (messageid)
        REFERENCES message (messageid)
);

```

Bijlage 7. Mockups

In dit hoofdstuk staan mijn ontwerpen beschreven met de daarbij horende uitleg. Het project ben ik begonnen met het ontwerpen van het systeem documentatie. Ik ben als eerste begonnen met het maken van Mockups aan de hand van de Mockups heb ik een Klasse diagram gemaakt. Nadat deze 2 klopte en volledig waren heb ik de SQL DDL gemaakt zodat de database gelijk volledig klaar stond van te voren.

1.2 Interface - Mockups

1.2.1 Mededelingenboard

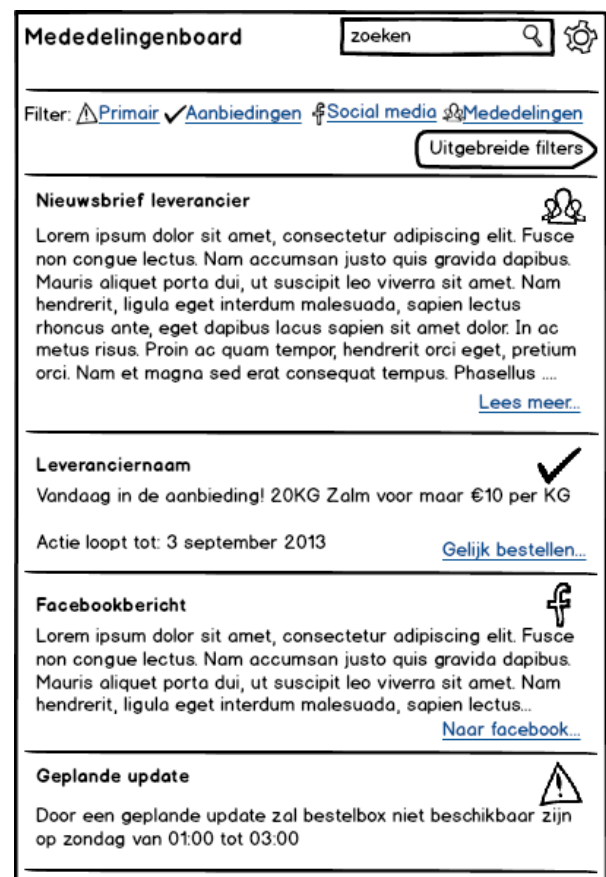
Het mededelingenboard zal onderdeel worden van de portal. en is daarom een smalle balk zodat deze in de portal geïmplementeerd kan worden. In deze balk komt een overzicht van de meest recente berichten.

1.2.2 Mededelingenboard overzicht

De berichten zullen op de volgende wijze worden weergegeven:

- Primair: Bovenaan. Deze worden altijd weergegeven omdat dit belangrijke berichten zijn zoals geplande updates, storingsen etc.
- Aanbiedingen. Hier zullen de meest recente aanbiedingen worden weergegeven welk zijn ingevoerd in de berichtenservice.
- Social Media berichten - De social media berichten zullen als een scroller worden weergegeven zodat er meerdere berichten op 1 plek kunnen worden weergegeven
- Mededelingen - Ook de mededelingen worden als een scroller weergegeven.

Hoeveel berichten worden weergegeven moet nog bekeken worden hoeveel ruimte de berichten service in gaat nemen met een aantal berichten.



Figuur 1 Mockup mededelingenboard

1.2.3 Filteren

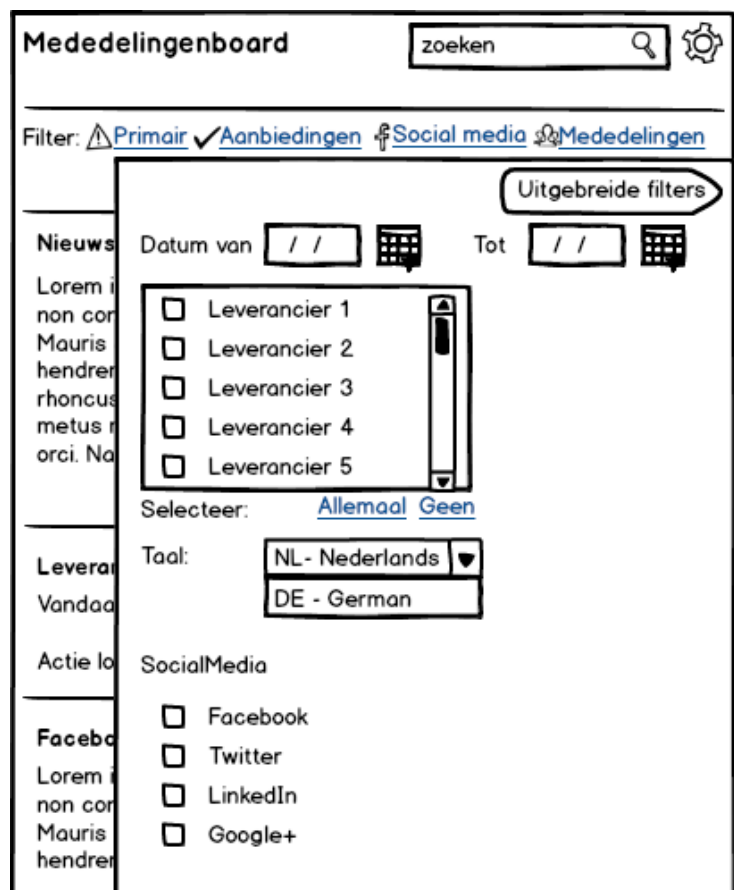


Figuur 2 Mockup filteren

De berichten kunnen op een simpele wijze worden gefilterd door op primair, Aanbiedingen, Social media, of mededelingen geklikt wordt. Door op een van deze knoppen te drukken zullen de berichten worden verborgen/weergeven. Of een bericht wordt verborgen of weergeven hang af van de status van de knop. Standaard staan de filters uit zodat alle berichten worden weergeven.

1.2.4 Uitgebreid filteren

Door op de knop uitgebreid filteren te klikken wordt er een popup geopend welke de mogelijkheid biedt om de berichten te filteren op datum, leverancier, taal of sociaal netwerk. De uitgebreide filters staan standaard op uit zodat alle berichten worden weergeven



Figuur 3 Mockup uitgebreid filteren

1.2.5 Instellingen

Er kunnen voor de berichtenservice een aantal instellingen worden ingesteld

- App notificaties: Als het gewenst is kunnen app notificaties aan of uit gezet worden. Welke app notificaties binnen komen kunnen hier ook worden ingesteld.
- Account bericht notificaties kunnen worden uitgeschakeld door "stuur notificaties naar mijn bestelbox App" uit te vinken
- Toon verlopen aanbiedingen. Deze staat standaard uit. Als het gewenst is kunnen oude verlopen aanbiedingen aangezet worden zodat je deze alsnog kan bekijken

Mededelingen

Filter: ⚠

Filter app notificaties:

- ☒ Stuur notificaties naar mijn bestelbox App
- ☒ Primair
- ☒ Aanbiedingen
- ☐ Social Media
- ☐ Mededelingen

☐ Toon verlopen aanbiedingen

Figuur 4 Mockup instellingen

1.3 Nieuw bericht

Er zijn een aantal opties welke standaard zijn voor alle berichten. Zo is het mogelijk om een bericht in te plannen of te laten verlopen, te publiceren op de sociale media en om naar een bepaald aantal accounts een bericht te versturen (leveranciers/horecalocaties).

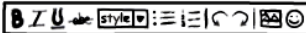
Voor elke bericht is het ook mogelijk om een titel en een omschrijving in te voeren. De beschrijving is alleen niet altijd mogelijk om te gebruiken bij mededeling, meer hierover bij het hoofdstuk van de mededelingen

1.3.1 Primair bericht

Een primair bericht is een "simpel" bericht. Deze berichten zijn bedoeld als een info bericht zonder speciale toevoegingen. Deze berichten kunnen alleen door inONE verstuurd worden en worden gebruikt voor storings updates e.d.


☒ Primair
☐ Reclame
☐ Mededeling


Titel



Versturen

Bericht instellingen

Inplannen op: / 

Verloopt op: / 

Publiceren op:
☒ Facebook
☒ Google+
☐ LinkedIn
☒ Twitter

Versturen naar

Zoeken

☒ Horeca 1
☒ Horeca 2
☒ Horeca 3
☒ Horeca 4
☒ Horeca 5
☒ Horeca 6
☐ Horeca 11
☐ Horeca 15

Selecteer Alles / Niks

Figuur 5 Mockup Primair bericht

1.3.2 Reclame bericht

Een reclame bericht kan worden gebruik om 1 of meerde producten te promoten. Bij een bericht moet altijd een product worden ingevoerd dit kan door middel van het productnummer.

☐ Primair
☒ Reclame
☐ Mededeling

Titel

B *I* U

Aanbieding betreft:
Productnummer: +

Een gevonden product word hier weergeven
En kunnen "oneindig" veel worden ingevuld
Zoveel als gewenst is.

Versturen

Bericht instellingen

Inplannen op: /

Verloopt op: /

Publiceren op:

☒ Facebook
☒ Google+
☐ LinkedIn
☒ Twitter

Versturen naar

Zoeken

☒ Horeca 1
☒ Horeca 2
☒ Horeca 3
☒ Horeca 4
☒ Horeca 5
☒ Horeca 6
☐ Horeca 11
☐ Horeca 15

Selecteer Alles / Niks

Figuur 6 Mockup Reclame bericht

1.3.3 Mededeling bericht

Het mededelingenbericht kan worden gebruikt om berichten te versturen die niet onder de andere types vallen. Dit kunnen bijvoorbeeld nieuwsbrieven zijn. Bij een mededeling kan een PDF bijgevoegd worden of een URL worden ingevoerd welke moeten worden weergeven.

☐ Primair
☐ Reclame
☒ Mededeling

Titel

B *I* U

Versturen

Bericht instellingen

Inplannen op: /

Verloopt op: /

Publiceren op:

☒ Facebook
☒ Google+
☐ LinkedIn
☒ Twitter

URL invoeren http://

PDF uploaden

Als PDF/URL is ingevuld word bericht schrijven uitgeschakeld
PDF en URL kunnen ook niet tegelijk werken

Versturen naar

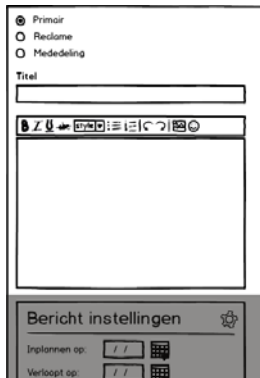
Zoeken

☒ Horeca 1
☒ Horeca 2
☒ Horeca 3
☒ Horeca 4
☒ Horeca 5
☒ Horeca 6
☐ Horeca 11
☐ Horeca 15

Selecteer Alles / Niks

Figuur 7 Mockup Mededeling bericht

1.3.4 Weergave op de App



Op deze wijze moet het invoeren van berichten worden weergegeven op de mobiele app. Dit zal er bijna identiek er uit komen te zien behalve dat de kolom aan de rechterkant onder het invoeren van een bericht worden weergegeven. Het grijze gedeelte in de mockup is als zijnde buiten het scherm.

Figuur 8 Mockup nieuw bericht op de app

1.4 Bestelbox App

De app zal een balk aan de linkerkant erbij krijgen zodat daar een plek is om de verschillende "Nieuwe" functionaliteiten neer te zetten. Dit zal eerst alleen voor de berichtenservice gelden. In de toekomst kunnen hier meerdere functionaliteiten toegevoegd worden. De balk kan je verbergen door naar links te swippen. Of weer te geven door aan de linker kant naar rechts te swippen



Figuur 9 Bestelbox app

1.5 PersonalMessage

1.5.1 Statusbalk

Bovenin bestelbox zal een menu knopje komen om de persoonlijke berichten te kunnen bekijken of openen in een kort overzicht van de laatste 4 berichten. Zodra je op een bericht klikt, opent zich een uitgebreider scherm waarmee je toegang krijgt tot het versturen van berichten.



Figuur 10 Mockup Persoonlijk bericht

1.5.2 Berichten overzicht

Door onderin een bericht op te stellen en te drukken op versturen wordt het mogelijk om met een van de eigen contacten (Lees leveranciers/horeca locaties) te chatten via de bestelbox. Ook kan er een status worden bijgehouden worden van de berichten. Wanneer een bericht geopend wordt verandert de status automatisch naar "open" er kan ook gekozen worden om een bericht weer op ongelezen te zetten of op gesloten. Zodra het bericht op gesloten staat zal het niet meer mogelijk zijn om berichten via de chatsessie te versturen. Om berichten naar de leverancier te versturen zal een nieuw bericht aan gemaakt moeten worden om op deze manier een nieuw bericht te versturen. Aan de linkerkant van het scherm staan alle

chatsessies welke zijn gestart met daarbij een icoon van de status. Aan de rechterkant staat de chatsessie. boven het invulveld is de mogelijkheid om de status te veranderen van de chatsessie.

Nieuw bericht

Leverancier 1

15-sept-2013 8:03

Geen punt. De vis doen we bij de...

Leverancier 4

12-sept-2013 15:33

Het is voor jullie geregeld. alles k...

Leverancier 12

9-sept-2013 12:46

Onder welke naam staat dit pro...

Leverancier 22

29-aug-2013 17:54

Het staat onder ingrediënten en d...

Aan: Leverancier 1

15-sept-2013 8:03

Horeca ondernemer:

Mijn vis is bijna op. Kan ik een spoedbestelling plaatsen bij jullie zodat de vis geleverd kan worden met de levering van vandaag?!

Leverancier 1

15-sept-2013 8:11

Dat is geen probleem, Zet maar door via bestelbox en wij regelen dat dit in orde komt!

Horeca ondernemer:

15-sept-2013 8:13

We gaan gelijk een bestelling plaatsen, bvd!

Horeca ondernemer:

15-sept-2013 8:29

De bestelling is geplaatst. Mijn dank is groot!

Leverancier 1

15-sept-2013 8:32

Geen punt. De vis doen we bij de lading van vandaag. Volgende keer graag wel letten op de voorraden :-)

Horeca ondernemer:

15-sept-2013 8:03

Mijn vis is bijna op. Kan ik een spoedbestelling plaatsen bij jullie zodat de vis geleverd kan worden met de levering van vandaag?!

Leverancier 1

15-sept-2013 8:11

Dat is geen probleem, Zet maar door via bestelbox en wij regelen dat dit in orde komt!

Horeca ondernemer:

15-sept-2013 8:13

We gaan gelijk een bestelling plaatsen, bvd!

Horeca ondernemer:

15-sept-2013 8:29

De bestelling is geplaatst. Mijn dank is groot!

Leverancier 1

15-sept-2013 8:32

Geen punt. De vis doen we bij de lading van vandaag. Volgende keer graag wel letten op de voorraden :-)

Unread Open Closed

Versturen

Figuur 11Mockup Persoonlijk bericht overzicht

1.5.3 Weergave op de app

De werking op de app werkt exact hetzelfde. Alleen is de chatsessie met het overzicht van de berichten los van elkaar gehaald.



Figuur 12 Mockup Persoonlijk bericht op de telefoon

Bijlage 8. Code voorbeeld filteren bericht

```
public static ArrayList<Message> getMessages(ArrayList<Account> sender, ArrayList<MessageType>
messageType, Timestamp fromDate, Timestamp toDate,
        ArrayList<SocialNetwork> socialNetwork, Account receiver) {
    ArrayList<Message> returnValue = new ArrayList<Message>();
    ResultSet rs = null;
    PreparedStatement pst = null;
    String query = "";
    String where = "";
    String joinString = "";
    String dateFromString = "";
    String dateToString = "";
    String senderString = "";
    String socialString = "";
    String receiverString = "";
    int receiverParameterKey = -1;
    HashMap<Integer, Object> parameterList = new HashMap<Integer, Object>();
    int counter = 1;
    for (Account send : sender) {
        if (where.isEmpty()) {
            where = " WHERE ";
        }
        if (senderString.isEmpty()) {
            senderString += " senderderaccountid = ? ";
        }
        else {
            senderString += " OR senderderaccountid = ? ";
        }
        parameterList.put(counter, send.getCustlogId());
        counter++;
    }

    if (fromDate != null) {
        if (where.isEmpty()) {
            where = " WHERE (";
        }
        else {
            dateFromString += " AND (";
        }
        dateFromString += " scheduled <= ? OR scheduled is null)";
        parameterList.put(counter, fromDate);
        counter++;
    }

    if (toDate != null) {
        if (where.isEmpty()) {
            where = " WHERE ";
        }
        else {
            dateToString += " AND (";
        }
        dateToString += " messageends >= ? OR messageends is null)";
        parameterList.put(counter, toDate);
        counter++;
    }

    if (receiverString != null) {
        if (where.isEmpty()) {
            where = " WHERE ";
        }
        else {
            receiverString += " AND ";
        }
        receiverString += " macc.receiveraccountid = ? ";
        parameterList.put(counter, receiver.getCustlogId());
        receiverParameterKey = counter;
        counter++;
        for (MessageType messageType : messageType) {
            if (messageType == MessageType.PRIMARYMESSAGE) {
                joinString = " join messageservice.primarymessage as pm on m.messageid =
pm.messageid ";
            }
            else if (messageType == MessageType.NOTICEMESSAGE) {
                joinString = " join messageservice.noticemessage as nm on m.messageid =
nm.messageid ";
            }
            else if (messageType == MessageType.ADVERTISEMENTMESSAGE) {
                joinString = " join messageservice.advertisementmessage as am on m.messageid
= am.messageid ";
            }
            else if (messageType == MessageType.SOCIALMESSAGE) {
                joinString = " join messageservice.socialmessage as sm on m.messageid =
sm.messageid ";
            }
            for (SocialNetwork soc : socialNetwork) {
                if (socialString.isEmpty()) {
                    socialString += " socialnetworkid = ?";
                }
                else {
                    socialString += " AND socialnetworkid = ?";
                }
                parameterList.put(counter, soc.getValue());
                counter++;
            }
        }
    }
    else if (messageType == MessageType.RSSFEEDMESSAGE) {
```

```

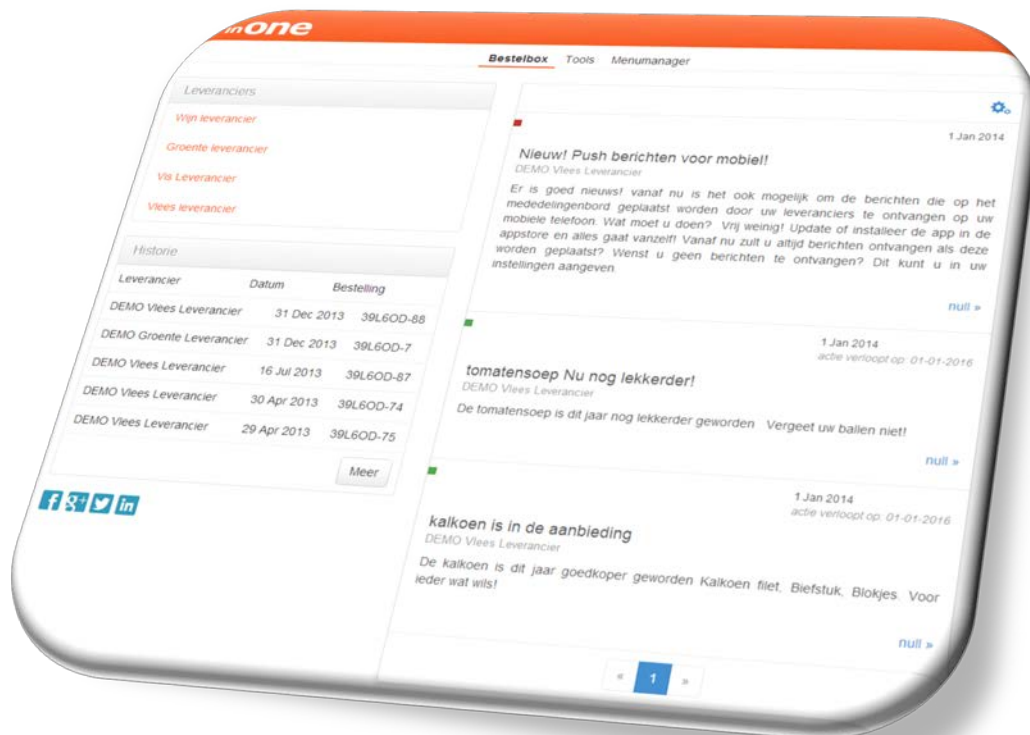
joinString = " join messageservice.rssfeedmessage as rsm on m.messageid =
rsm.messageid ";
    }
    if (mType != MessageType.PRIMARYMESSAGE) {
joinString += " join messageservice.message_account as macc on
macc.messageid = m.messageid ";
        parameterList.put(receiverParameterKey, receiver.getCustlogId());
        query = "SELECT * FROM messageservice.message as m " + joinString + where +
senderString + dateFromString + dateToString
+ socialString + receiverString + " order by messagedate desc;";
    }
    else {
        query = "SELECT * FROM messageservice.message as m " + joinString + where +
dateFromString + dateToString
" order by messagedate desc;";
        parameterList.remove(receiverParameterKey);
    }
    try {
pst = SQLQuery.getDbConn().prepareStatement(query);
pst = HelpClass.setParametersPreparedStatement(pst, parameterList);
rs = pst.executeQuery();
        while (rs.next()) {
            if (mType == MessageType.PRIMARYMESSAGE) {
returnValue.add(new PrimaryMessage(rs));
            }
            else if (mType == MessageType.NOTICEMESSAGE) {
returnValue.add(new NoticeMessage(rs));
            }
            else if (mType == MessageType.ADVERTISEMENTMESSAGE) {
returnValue.add(new AdvertisementMessage(rs));
            }
            else if (mType == MessageType.SOCIALMESSAGE) {
returnValue.add(new SocialMessage(rs));
            }
            else if (mType == MessageType.RSSFEEDMESSAGE) {
returnValue.add(new RssFeedMessage(rs));
            }
            else if (mType == MessageType.GHOSTMESSAGE) {
returnValue.add(new RssFeedMessage(rs));
            }
        }
    }
    catch (SQLException e) {
        Debug.consoleWriteln(e.getLocal izedMessage());
        Debug.consoleWriteln("<b>getMessage - messageType</b>" +
mType.toString() + " error: " + e);
    }
    finally {
        SQLQuery.close(rs);
        SQLQuery.close(pst);
    }
}
Collections.sort(returnValue, new DateComperator());
Collections.sort(returnValue, new TypeComperator());
return returnValue; }

```

Bijlage 9. Test plan

Testplan

Berichtenservice
Marcel Fasseur



Inhoudsopgave

Inhoudsopgave	2
Inleiding	1
1. Uitvoering van de testen	2
2. Sprint 1	3
2.1 Whitebox test testscript	4
2.2 Blackbox test testscript	5
3. Sprint 2	6
3.1 Whitebox test testscript	6
3.2 Blackbox test testscript	7
4. Sprint 3	8
4.1 Whitebox test testscript	8
4.2 Blackbox test testscript	9
5. Sprint 4	10
5.1 Whitebox test testscript	10
5.2 Blackbox test testscript	11
6. Sprint 5	12
6.1 Whitebox test testscript	12
6.2 Blackbox test testscript	12

Inleiding

Dit test plan is geschreven om een goede basis neer te leggen voor de testen welke worden uitgevoerd voor de berichtenservice. Gaandeweg het project zal dit document worden aangepast en aangevuld per sprint.

In het eerste hoofdstuk ga ik in op welke manier de testen moeten worden uitgevoerd, De daarop volgende hoofdstukken zijn alle sprints van het project. Deze sprints zijn opgedeeld in 2 hoofdstukken namelijk een hoofdstuk whitebox testscript en een hoofdstuk blackbox testscript

1. Uitvoering van de testen

Elke sprint wordt een junit test uitgevoerd waar dit mogelijk is. Zodra een sprint wordt gestart en het duidelijk is welke story points in de sprint worden uitgevoerd moet het test plan bijgewerkt worden met de nieuwe sprint.

De testen dienen te worden uitgevoerd op de testomgeving van inONE en moeten beginnen met een lege database voor de berichtenservice. Dit geldt alleen voor de berichtenservice database, voor de inONE database geldt het dus niet omdat deze data gelijk is aan de productie omgeving. De whitebox test zal worden uitgevoerd door gebruik te maken van junit en de blackbox test wordt via een browser getest.

Voor elke sprint zal een testscript aangemaakt moeten worden zodat het duidelijk is welke gegevens en punten getest moeten worden. De bevindingen moeten worden opgeslagen in de testrapportage. Wanneer een script het invoeren van berichten bevat moeten de waardes die zijn ingevoerd opgehaald worden uit de database zodat deze kan worden vergeleken met hetgene wat is ingevoerd. Hiervoor kan in java bij de modellen die getest moeten worden een "equals"methode worden bijgevoegd.

Voor het testscript dient de volgende template gebruikt te worden.

TestNr.	Storypoint Key	Stap	Test omschrijving	Verwacht resultaat

2. Sprint 1

In deze sprint worden de volgende story points getest:

Key	Summary
BER-5	Als inONE wil ik het MessageType Primair gebruiken om belangrijke berichten te versturen zoals storingen of updates.
BER-4	Als een Stakeholder wil ik een drietal types in kunnen stellen: Primair, Aanbiedingen en Mededelingen
BER-3	Als inONE wil ik een bericht kunnen sturen naar 1 of meerdere leveranciers/horecamanagers en horeca
BER-1	Als een leverancier wil ik een bericht kunnen sturen naar 1 of meer klanten.

De volgende gegevens dienen te worden gebruikt voor alle testen.

- Voor alle berichten:
 - Titel: Bericht titel voor test
 - Tekst: Bericht tekst voor de testen van sprint 1
 - Begin datum: 1-1-2013
 - Eind datum: 1-1-2015
 - ontvangers(CustlogID):
 - VSV01
 - 5L3LMM
 - XFO4IF
 - Afzender(CustlogID):
 - W3FUE6
- Voor een advertentie bericht
 - Product (ID): 01L9R2_100001_KG
- Voor een mededeling bericht.
 - PDFURL: XFO4IF/messageservice/1382604849080.pdf
 - Iframe Url: <http://www.google.nl>

2.1 Whitebox test testscript

De volgende parameters dienen te worden gebruikt voor alle testen

TestNr.	Storypoint Key	Stap	Test omschrijving	Verwacht resultaat
1	BER-4/ BER-5	1	Bericht aanmaken met type Primair	PrimaryMessage is geïnitieerd
	BER-4/ BER-5	2	PrimaryMessage Opslaan	PrimaryMessage staat in de dtb
	BER-4	3	Bericht aanmaken met type Advertisment	AdvertismentMessage is geïnitieerd
	BER-4	4	AdvertismentMessage opslaan	AdvertismentMessage staat in de database
	BER-4	5	Bericht aanmaken met type Notice	NoticeMessage is geïnitieerd
	BER-4	6	NoticeMessage opslaan	NoticeMessage staat in de database
2	BER-3	1	GhostMessage aanmaken	GhostMessage is geïnitieerd
	BER-3	2	Ontvangers leveranciers en horecamanagers invoeren	Lijst met ontvangers bestaat
	BER-3	3	Bericht inclusief ontvangers opslaan	Het bericht en de ontvangers staan in de database
3	BER-1	1	GhostMessage aanmaken	
	BER-1	2	Horeca bedrijven als ontvangers instellen	
	BER-1	3	Bericht inclusief ontvangers opslaan	Bericht en ontvangers staan in database

2.2 Blackbox test testscript

Omdat er nog geen view beschikbaar is zal er in de My SQL database gekeken worden of de waardes goed zijn ingevoerd in de database.

TestNr.	TestNr.	Storypoint Key	Stap	Test omschrijving	Verwacht resultaat
4	BER-4/ BER-5	BER-4/ BER-5	1	Gegevens van bericht invoeren met primair bericht geselecteerd	
		BER-4/ BER-5	2	Klikken op Bericht versturen	primair bericht aangemaakt in de dtb
	BER-4	BER-4	3	Gegevens van bericht invoeren met advertentie bericht geselecteerd	
		BER-4	4	Klikken op Bericht versturen	Advertentie bericht aangemaakt in de dtb
		BER-4	5	Gegevens van bericht invoeren met mededeling bericht geselecteerd	
		BER-4	6	Klikken op Bericht versturen	Mededeling bericht aangemaakt in de dtb
5	BER-3	BER-3	1	Inloggen als zijnde inONE	
		BER-3	2	Gegevens van bericht invoeren met primair bericht geselecteerd	
		BER-3	3	Alle mogelijke ontvangers selecteren	In de lijst moeten leveranciers en horeca voorkomen
		BER-3	4	Bericht naar ontvangers versturen	Alle ontvangers staan in de dtb
6	BER-1	BER-1	1	Inloggen als zijnde Leverancier	Ingelogd als leverancier
		BER-1	2	Gegevens van bericht invoeren met primair bericht geselecteerd	
		BER-1	3	Alle mogelijke ontvangers selecteren	In de lijst moeten alleen horeca voorkomen
		BER-1	4	Bericht naar ontvangers versturen	Alle ontvangers staan in de dtb
7	BER-4	BER-4	1	Klik op versturen zonder waardes in te vullen	Er gebeurt niks.
		BER-4	2	Alle waardes invullen zoals staat beschreven.	
		BER-4	3	Versturen op veld "asdd" invullen	Er gebeurt niks.
		BER-4	4	Bericht versturen	Er gebeurt niks.

3. Sprint 2

In deze sprint worden de volgende story points getest:

Key	Summary
BER-26	wysiwyg editor uitzoeken en implementeren
BER-12	Als een Stakeholder wil ik alle berichten in een "container" kunnen zien.
BER-10	Als een Stakeholder wil ik berichten via een online versie of een mobiele versie kunnen versturen en ontvangen
BER-7	Als een Stakeholder wil ik een filter mogelijkheid hebben voor de berichten zodat ik niet wordt overladen met berichten

3.1 Whitebox test testscript

Deze test is alleen gebaseerd op BER-7 omdat de overige story points geen Java aanpassing zal zijn.

De gegevens van een bericht die zijn beschreven in sprint 1 kunnen worden hergebruikt. Voor elk type moet 1 bericht worden aangemaakt. Hiervoor kan de junit worden gedraaid uit sprint 1 om de gegevens in de database op te slaan.

TestNr.	Storypoint Key	Stap	Test omschrijving	Verwacht resultaat
8	BER-7	1	Filter op advertentie bericht	1 bericht wordt terug gegeven
	BER-7	2	Filter op primair bericht	1 bericht wordt terug gegeven
	BER-7	3	Filter op mededeling bericht	1 bericht wordt terug gegeven
	BER-7	4	Filter op alle types	3 berichten worden terug gegeven

3.2 Blackbox test testscript

TestNr.	Storypoint Key	Stap	Test omschrijving	Verwacht resultaat
9	BER-26	1	HTML document plakken in de wysiwyg	Resultaat ziet er hetzelfde uit als de bron
	BER-26	2	Word document plakken in de wysiwyg	Resultaat ziet er hetzelfde uit als de bron
	BER-26	3	Bericht opslaan	
	BER-26/ BER-12	4	Bericht bekijken	Ingevuld bericht wordt hetzelfde weergegeven in de container als hoe deze eruit zag in de wysiwyg. Ook worden de gegevens die zijn beschreven in sprint 1 weergegeven
10	BER-10	1	Venster op 1024x768 instellen	De panels worden weergegeven zoals in de mockup staat beschreven
	BER-10	2	Venster smaller maken	De panels worden onder elkaar weergegeven zoals in de mockup staat beschreven
11	BER-7	1	Zet alleen filter primair bericht aan	1 bericht wordt weergegeven
	BER-7	2	Zet alleen filter advertentie bericht aan	1 bericht wordt weergegeven
	BER-7	3	Zet alleen filter mededeling bericht aan	1 bericht wordt weergegeven
	BER-7	4	Zet alle filters uit	3 berichten worden weergegeven

4. Sprint 3

In deze sprint worden de volgende story points getest:

Key	Summary
BER-32	Als een stakeholder wil ik via een type ahead een product kunnen zoeken in een nieuw bericht
BER-29	Als een stakeholder wil ik validatie hebben over de ingevulde gegevens
BER-40	Als een stakeholder wil ik een PDF kunnen uploaden
BER-41	Als een stakeholder wil ik oude berichten kunnen opzoeken

4.1 Whitebox test testscript

De whitebox testen heb ik alleen gebaseerd op BER-41 “Als een stakeholder wil ik oude berichten kunnen opzoeken” omdat de andere story points in deze sprint niet met een whitebox test kan worden getest omdat alleen front-end (javascript) code zal moeten worden toegevoegd bij deze punten.

De gegevens van een bericht die zijn beschreven in sprint 1 kunnen worden hergebruikt. Voor elk type moet 1 bericht worden aangemaakt. Hiervoor kan de junit worden gedraaid uit sprint 1 om de gegevens in de database op te slaan.

	Storypoint Key	Stap	Test omschrijving	Verwacht resultaat
12	BER-41	1	Voer zoektem “sprint 1” in	
	BER-41	2	Voer zoeken uit	3 berichten worden terug gegeven.
	BER-41	3	Voer zoektem “vind niks” in	
	BER-41	4	Voer zoeken uit	0 berichten worden terug gegeven.
	BER-41	5	Voer zoektem “titel” in	
	BER-41	6	Voer zoeken uit	3 berichten worden terug gegeven.

4.2 Blackbox test testscript

	Storypoint Key	Stap	Test omschrijving	Verwacht resultaat
13	BER-32	1	Voer "100001_KG" in in de typeahead textbox in	1 Product namelijk "andijvie" word weergeven.
	BER-32	2	Voer "andijvie"	1 Product namelijk "andijvie" word weergeven met id 100001_KG
14	BER-29	1	Bij het invoeren geen titel invoeren	Bij "lost focus" van het titel veld moet een popup worden weergeven met de melding "This field is required".
	BER-29	2	In het "versturen op" veld "SDFSf" invullen	Bij "lost focus" van het versturen op veld moet een popup worden weergeven met de melding "Date is not the correct format"
	BER-29	3	In het "Verloopt op" veld "SDFSf" invullen	Bij "lost focus" van het versturen op veld moet een popup worden weergeven met de melding "Date is not the correct format"
	BER-29	4	In het veld Iframe URL "SDFSf" invullen	Bij "lost focus" van het versturen op veld moet een popup worden weergeven met de melding "URL is not the correct format"
	BER-29	5	Klik op bericht versturen	Alle validatie meldingen worden weergeven.
	BER-29	6	Bij het invoeren de titel "Dit is een bericht titel voor test" invoeren	Bij "lost focus" van het titel veld moet de popup niet meer zichtbaar zijn
	BER-29	7	In het "versturen op" veld "1-1-2014" invullen	Bij "lost focus" van het versturen op veld moet de popup niet meer zichtbaar zijn
	BER-29	8	In het "Verloopt op" veld "1-1-2015" invullen	Bij "lost focus" van het versturen op veld moet de popup niet meer zichtbaar zijn
	BER-29	9	In het veld Iframe URL "www.inone.nl" invullen	Bij "lost focus" van het versturen op veld moet de popup niet meer zichtbaar zijn
	BER-29	10	Klik op bericht versturen	Het bericht is opgeslagen
15	BER-40	1	Een JPEG bestand uploaden	Er gebeurt niks.
	BER-41	2	Een PDF bestand uploaden	PDF succesvol geüpload.

5. Sprint 4

In deze sprint worden de volgende story points getest:

Key	Summary
BER-2	Als horecamanager wil ik een bericht kunnen sturen naar 1 of meer locaties.
BER-9	Als Stakeholder wil ik instellingen voor filters eenmalig in hoeven stellen, maar ook later kunnen wijzigen
BER-37	Als een stakeholder wil ik pagination hebben op de portal.
BER-36	De afbeeldingen bij de producten moeten bij een product in een aanbieding opgehaald worden. Dit ophalen uit notes database

5.1 Whitebox test testscript

Deze sprint heb ik de whitebox test gebaseerd op het opslaan van filteren (BER-9), en het versturen naar horeca managers (BER-2).

Voor story point BER-2 moet "VSV01" worden gebruikt als huidig account. Als ontvangers moeten de volgende custlogIds gebruikt worden:

- HS6W7
- JQUD8
- 1PSDM7

TestNr.	Storypoint Key	Stap	Test omschrijving	Verwacht resultaat
16	BER-9	1	Settings initiëren	Alle filters staan op "aan"
	BER-9	2	Zet de filter primair bericht uit	De regel voor primair bericht staat niet meer in de database
	BER-9	3	Zet de filter advertentie bericht uit	De regel voor advertentie staat niet meer in de database
	BER-9	4	Zet de filter mededeling bericht uit	De regel voor mededeling bericht staat niet meer in de database
	BER-9	5	Zet de filter primair bericht uit	De regel voor primair bericht staat niet meer in de database
	BER-9	6	Zet de filter advertentie bericht aan	De regel voor advertentie staat in de database
	BER-9	7	Zet de filter mededeling bericht aan	De regel voor mededeling bericht staat in de database
	BER-9	8	Zet de filter mededeling bericht	De regel voor mededeling bericht staat in de database

			aan
17	BER-2	1	Stel de waardes in zoals staat beschreven in Sprint 1
	BER-2	2	Gebruik de custlogids als ontvangers en afzender welke staan omschreven in hoofdstuk 5.1

5.2 Blackbox test testscript

Voor de blackbox test kunnen de junit testen gebruikt worden uit sprint 1. Deze moeten 5 keer worden uitgevoerd om meerdere berichten in de database te hebben om de pagination te kunnen testen.

TestNr.	Storypoint Key	Stap	Test omschrijving	Verwacht resultaat
18	BER-2	1	Bericht invullen zoals staat beschreven in Sprint 1	
	BER-2	2	Lijst met ontvangers selecteren	Lijst met ontvangers bestaat en er zijn een aantal locaties aanwezig in de lijst
	BER-2	3	Bericht versturen	Bericht is ontvangen bij de horeca locaties
19	BER-9	1	Portal openen voor de eerste keer	Alle filters staan op "aan"
	BER-9	2	Zet de filter primair bericht uit	Na herladen van de pagina staat deze nog steeds uit
	BER-9	3	Zet de filter advertentie bericht uit	Na herladen van de pagina staat deze nog steeds uit
	BER-9	4	Zet de filter mededeling bericht uit	Na herladen van de pagina staat deze nog steeds uit
	BER-9	5	Zet de filter primair bericht uit	Na herladen van de pagina staat deze nog steeds uit
	BER-9	6	Zet de filter advertentie bericht aan	Na herladen van de pagina staat deze nog steeds aan
	BER-9	7	Zet de filter mededeling bericht aan	Na herladen van de pagina staat deze nog steeds aan
	BER-9	8	Zet de filter mededeling bericht aan	Na herladen van de pagina staat deze nog steeds aan
20	BER-37	1	Open de portal	Op de portal worden 3 pagina's weergegeven (dus het ziet er zo uit < 1 2 3 >)
	BER-37	2	Navigeer naar pagina 2	de volgende aantal berichten worden weergegeven
21	BER-36	1	Een advertentie bericht openen	Er staat een afbeelding van het product i.p.v. een 404 afbeelding

6. Sprint 5

In deze sprint worden de volgende story points getest:

Key	Summary
BER-8	Als een bestelbox app gebruiker wil ik notificaties krijgen van belangrijke berichten op mijn mobiel apparaat en in de statusbalk van het mobiele platform weergegeven worden.
BER-38	BER-8 Android notificaties maken

6.1 Whitebox test testscript

Voordat de whitebox test uitgevoerd kan worden moet eerst de blackbox test worden uitgevoerd voor de registratie van de mobiele device.

TestNr.	Storypoint Key	Stap	Test omschrijving	Verwacht resultaat
22	BER-38	1	Verstuur bericht naar account "VSV01"	Statusbericht "Ok" word teruggegeven

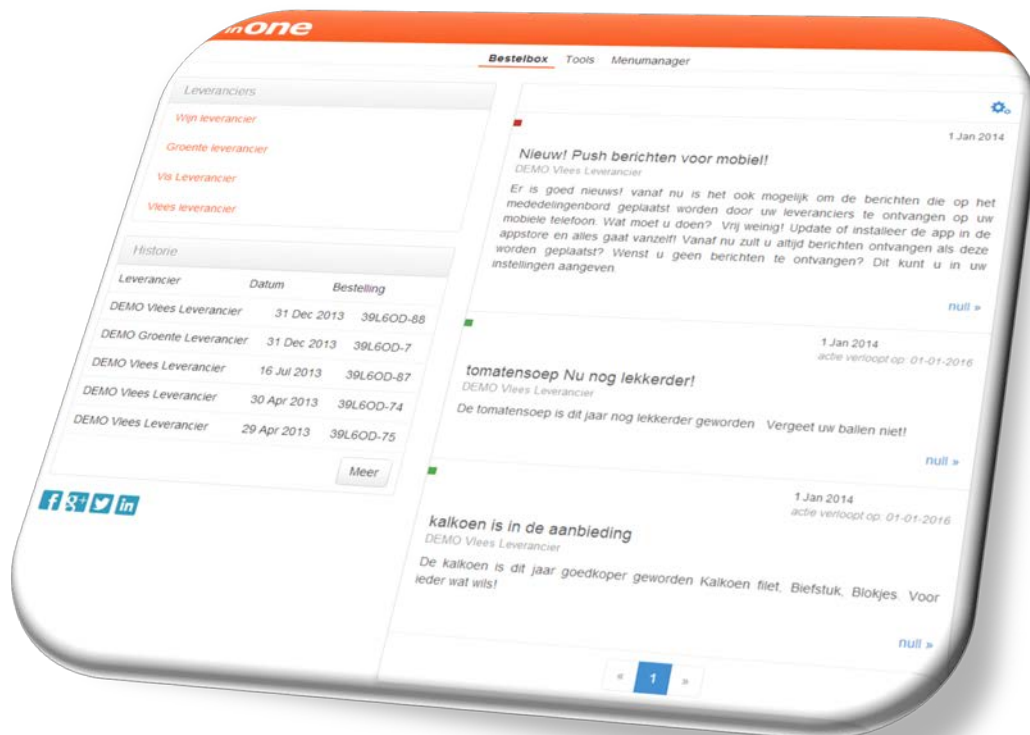
6.2 Blackbox test testscript

TestNr.	Storypoint Key	Stap	Test omschrijving	Verwacht resultaat
23	BER-38	1	Open de bestelbox App	
	BER-38	2	Log in op de bestelbox App als VSV01	
	BER-38	3	Maak een bericht aan met de gegevens uit sprint 1	
	BER-38	4	Bekijk de mobiele telefoon	Er moet een bericht binnen zijn gekomen op het mobiele device

Bijlage 10. Testrapportage

Testrapportage

Berichtenservice
Marcel Fasseur



Inhoudsopgave

Inhoudsopgave	1
Inleiding	1
1. Sprint 1	2
2.1 Whitebox test rapportage	2
2.2 Blackbox test rapportage	3
2. Sprint 2	4
3.1 Whitebox test rapportage	5
3.2 Blackbox test rapportage	6
3. Sprint 3	7
4.1 Whitebox test rapportage	7
4.2 Blackbox test rapportage	7
4. Sprint 4	9
5.1 Whitebox test rapportage	9
5.2 Blackbox test rapportage	9
5. Sprint 5	11
6.1 Whitebox test rapportage	11
6.2 Blackbox test rapportage	11

Inleiding

Dit is de testrapportage van de berichtenservice. De indeling van de hoofdstukken in dit document is hetzelfde als in het test plan zodat terugkoppeling gemaakt kan worden. De hoofdstukken zijn per sprint beschreven en in de sprint hoofdstukken staan een hoofdstuk whitebox test rapportage en een hoofdstuk blackbox test rapportage

1. Sprint 1

1.1 Whitebox test rapportage

Datum uitgevoerd: 2 oktober 2013

TestNr.	Storypoint Key	Stap	Test omschrijving	Verwacht resultaat	Daadwerkelijk resultaat	Geslaagd
1	BER-4/ BER-5	1	Bericht aanmaken met type Primair	PrimaryMessage is geïnitieerd	PrimaryMessage is geïnitieerd	Ja
	BER-4/ BER-5	2	PrimaryMessage Opslaan	PrimaryMessage staat in de dtb	PrimaryMessage is aanwezig in de dtb	Ja
	BER-4	3	Bericht aanmaken met type Advertisment	AdvertismentMessage is geïnitieerd	AdvertismentMessage is geïnitieerd	Ja
	BER-4	4	AdvertismentMessage opslaan	AdvertismentMessage staat in de database	AdvertismentMessage is aanwezig in de dtb	Ja
	BER-4	5	Bericht aanmaken met type Notice	NoticeMessage is geïnitieerd	NoticeMessage is geïnitieerd	Ja
	BER-4	6	NoticeMessage opslaan	NoticeMessage staat in de database	NoticeMessage is aanwezig in de dtb	Ja
2	BER-3	1	GhostMessage aanmaken	GhostMessage is geïnitieerd	GhostMessage is geïnitieerd	Ja
	BER-3	2	Ontvangers leveranciers en horecamanagers invoeren	Lijst met ontvangers bestaat	Lijst met ontvangers bestaat	Ja
	BER-3	3	Bericht inclusief ontvangers opslaan	Het bericht en de ontvangers staan in de database	Het bericht en de ontvangers zijn aanwezig in de dtb	Ja
3	BER-1	1	GhostMessage aanmaken			Ja
	BER-1	2	Horeca bedrijven als ontvangers instellen			Ja
	BER-1	3	Bericht inclusief ontvangers opslaan	Bericht en ontvangers staan in database	Bericht en ontvangers zijn aanwezig in de dtb	Ja

1.2 Blackbox test rapportage

Datum uitgevoerd: 2 oktober 2013

TestNr.	Storypoint Key	Stap	Test omschrijving	Verwacht resultaat	Daadwerkelijk resultaat	Geslaagd
4	BER-4/ BER-5	1	Gegevens van bericht invoeren met primair bericht geselecteerd			
	BER-4/ BER-5	2	Klikken op Bericht versturen	primair bericht aangemaakt in de dtb	primair bericht is aanwezig in de dtb	Ja
	BER-4	3	Gegevens van bericht invoeren met advertentie bericht geselecteerd			Ja
	BER-4	4	Klikken op Bericht versturen	Advertentie bericht aangemaakt in de dtb	Advertentie bericht is aanwezig in de dtb	Ja
	BER-4	5	Gegevens van bericht invoeren met mededeling bericht geselecteerd			Ja
	BER-4	6	Klikken op Bericht versturen	Mededeling bericht aangemaakt in de dtb	Mededeling bericht is aanwezig in de dtb	Ja
5	BER-3	1	Inloggen als zijnde inONE			Ja
	BER-3	2	Gegevens van bericht invoeren met primair bericht geselecteerd			Ja
	BER-3	3	Alle mogelijke ontvangers selecteren	In de lijst moeten leveranciers en horeca voorkomen	In de lijst komen leveranciers en horeca voor	Ja
	BER-3	4	Bericht naar ontvangers versturen	Alle ontvangers staan in de dtb	Alle ontvangers zijn aanwezig in de dtb	Ja
6	BER-1	1	Inloggen als zijnde Leverancier	Ingelogd als leverancier	Ingelogd als leverancier	Ja
	BER-1	2	Gegevens van bericht invoeren met primair bericht geselecteerd			Ja
	BER-1	3	Alle mogelijke ontvangers selecteren	In de lijst moeten alleen horeca voorkomen	In de lijst komt alleen horeca voor	Ja

	BER-1	4	Bericht naar ontvangers versturen	Alle ontvangers staan in de dtb	Alle ontvangers z	Ja
7	BER-4	1	Klik op versturen zonder waardes in te vullen	Er gebeurt niks.	NullPointerException	Gefaald
	BER-4	2	Alle waardes invullen zoals staat beschreven.			Ja
	BER-4	3	Versturen op veld "asdd" invullen	Er gebeurt niks.	Er gebeurt niks.	Ja
	BER-4	4	Bericht versturen	Er gebeurt niks.	FormatException	Gefaald

Test nummer 7 stap 1 en 4 zijn Gefaald want deze gaven een foutmelding terug. In de code heb ik nog geen rekening gehouden met validatie over de gegevens om deze reden moet er een story point worden aangemaakt zodat dit onderdeel uitgewerkt kan worden later in het project.

2. Sprint 2

2.1 Whitebox test rapportage

Datum uitgevoerd: 27 oktober 2013

TestNr.	Storypoint Key	Stap	Test omschrijving	Verwacht resultaat	Daadwerkelijk resultaat	Geslaagd
8	BER-7	1	Filter op advertentie bericht	1 bericht wordt terug gegeven	1 bericht wordt terug gegeven	Ja
	BER-7	2	Filter op primair bericht	1 bericht wordt terug gegeven	1 bericht wordt terug gegeven	Ja
	BER-7	3	Filter op mededeling bericht	1 bericht wordt terug gegeven	1 bericht wordt terug gegeven	Ja
	BER-7	4	Filter op alle types	3 berichten worden terug gegeven	3 berichten worden terug gegeven	Ja

2.2 Blackbox test rapportage

Datum uitgevoerd: 27 oktober 2013

TestNr.	Storypoint Key	Stap	Test omschrijving	Verwacht resultaat	Daadwerkelijk resultaat	Geslaagd
9	BER-26	1	HTML document plakken in de wysiwyg	Resultaat ziet er hetzelfde uit als de bron	Het resultaat ziet er hetzelfde uit als de bron	Ja
	BER-26	2	Word document plakken in de wysiwyg	Resultaat ziet er hetzelfde uit als de bron	Het resultaat ziet er hetzelfde uit als de bron	Ja
	BER-26	3	Bericht opslaan			Ja
	BER-26/ BER-12	4	Bericht bekijken	Ingevuld bericht wordt hetzelfde weergegeven in de container als hoe deze eruit zag in de wysiwyg. Ook worden de gegevens die zijn beschreven in sprint 1 weergegeven	Ingevuld bericht wordt hetzelfde weergegeven in de container als hoe deze eruit ziet in de wysiwyg. Ook worden de gegevens die zijn beschreven in sprint 1 weergegeven	Ja
10	BER-10	1	Venster op 1024x768 instellen	De panels worden weergegeven zoals in de mockup staat beschreven	De panels worden weergegeven zoals in de mockup staat beschreven	Ja
	BER-10	2	Venster smaller maken	De panels worden onder elkaar weergegeven zoals in de mockup staat beschreven	De panels worden onder elkaar weergegeven zoals in de mockup staat beschreven	Ja
11	BER-7	1	Zet alleen filter primair bericht aan	1 bericht wordt weergegeven	1 bericht wordt weergegeven	Ja
	BER-7	2	Zet alleen filter advertentie bericht aan	1 bericht wordt weergegeven	1 bericht wordt weergegeven	Ja
	BER-7	3	Zet alleen filter mededeling bericht aan	1 bericht wordt weergegeven	1 bericht wordt weergegeven	Ja
	BER-7	4	Zet alle filters uit	3 berichten worden weergegeven	3 berichten worden weergegeven	Ja

3. Sprint 3

3.1 Whitebox test rapportage

Datum uitgevoerd: 13 november 2013

	Storypoint Key	Stap	Test omschrijving	Verwacht resultaat	Daadwerkelijk resultaat	Geslaagd
12	BER-41	1	Voer zoektem "sprint 1" in			Ja
	BER-41	2	Voer zoeken uit	3 berichten worden terug gegeven.	3 berichten worden terug gegeven.	Ja
	BER-41	3	Voer zoektem "vind niks" in			Ja
	BER-41	4	Voer zoeken uit	0 berichten worden terug gegeven.	0 berichten worden terug gegeven.	Ja
	BER-41	5	Voer zoektem "titel" in			Ja
	BER-41	6	Voer zoeken uit	3 berichten worden terug gegeven.	3 berichten worden terug gegeven.	Ja

3.2 Blackbox test rapportage

Datum uitgevoerd: 13 november 2013

	Storypoint Key	Stap	Test omschrijving	Verwacht resultaat	Daadwerkelijk resultaat	Geslaagd
13	BER-32	1	Voer "100001_KG" in in de typeahead textbox in	1 Product namelijk "andijvie" word weergeven.	1 Product namelijk "andijvie" word weergeven.	Ja
	BER-32	2	Voer "andijvie"	1 Product namelijk "andijvie" word weergeven met id 100001_KG	1 Product namelijk "andijvie" word weergeven met id 100001_KG	Ja
14	BER-29	1	Bij het invoeren geen titel invoeren	Bij "lost focus" van het titel veld moet een popup worden weergeven met de melding "This field is required".	Het veld This field is required wordt weergeven	Ja
	BER-29	2	In het "versturen op" veld "SDFSf" invullen	Bij "lost focus" van het versturen op veld moet een popup worden weergeven met de melding "Date is not	De melding "Date is not the correct format" wordt weergeven	Ja

			the correct format"			
	BER-29	3	In het "Verloopt op" veld "SDFSf" invullen	Bij "lost focus" van het versturen op veld moet een popup worden weergegeven met de melding "Date is not the correct format"	De melding "Date is not the correct format" wordt weergegeven	Ja
	BER-29	4	In het veld Iframe URL "SDFSf" invullen	Bij "lost focus" van het versturen op veld moet een popup worden weergegeven met de melding "URL is not the correct format"	De melding "URL is not the correct format" word weergegeven	Ja
	BER-29	5	Klik op bericht versturen	Alle validatie meldingen worden weergegeven.	Alle validatie meldingen worden weergegeven.	Ja
	BER-29	6	Bij het invoeren de titel "Dit is een bericht titel voor test" invoeren	Bij "lost focus" van het titel veld moet de popup niet meer zichtbaar zijn	De popup is niet meer zichtbaar	Ja
	BER-29	7	In het "versturen op" veld "1-1-2014" invullen	Bij "lost focus" van het versturen op veld moet de popup niet meer zichtbaar zijn	De popup is niet meer zichtbaar	Ja
	BER-29	8	In het "Verloopt op" veld "1-1-2015" invullen	Bij "lost focus" van het versturen op veld moet de popup niet meer zichtbaar zijn	De popup is niet meer zichtbaar	Ja
	BER-29	9	In het veld Iframe URL "www.inone.nl" invullen	Bij "lost focus" van het versturen op veld moet de popup niet meer zichtbaar zijn	De popup is niet meer zichtbaar	Ja
	BER-29	10	Klik op bericht versturen	Het bericht is opgeslagen	Het bericht is opgeslagen	Ja
15	BER-40	1	Een JPEG bestand uploaden	Er gebeurt niks.	JPEG kan geüpload worden	Gefaald
	BER-41	2	Een PDF bestand uploaden	PDF succesvol geüpload.	PDF succesvol geüpload.	Ja

Testnummer 15 stap 1 is gefaald omdat het niet mogelijk zou moeten zijn om een ander bestandstype te uploaden dan PDF. Na wat onderzoek werk ben ik erachter gekomen dat de check mist om welk bestandstype het gaat. Nadat ik de check erin heb verwerkt was het alleen nog maar mogelijk om een PDF te uploaden

4. Sprint 4

4.1 Whitebox test rapportage

Datum uitgevoerd: 2 december 2013

TestNr.	Storypoint Key	Stap	Test omschrijving	Verwacht resultaat	Daadwerkelijk resultaat	Geslaagd
16	BER-9	1	Settings initiëren	Alle filters staan op "aan"	Alle filters staan op "aan"	Ja
	BER-9	2	Zet de filter primair bericht uit	De regel voor primair bericht staat niet meer in de database	De regel voor primair bericht staat niet meer in de database	Ja
	BER-9	3	Zet de filter advertentie bericht uit	De regel voor advertentie staat niet meer in de database	De regel voor advertentie staat niet meer in de database	Ja
	BER-9	4	Zet de filter mededeling bericht uit	De regel voor mededeling bericht staat niet meer in de database	De regel voor mededeling bericht staat niet meer in de database	Ja
	BER-9	5	Zet de filter primair bericht uit	De regel voor primair bericht staat niet meer in de database	De regel voor primair bericht staat niet meer in de database	Ja
	BER-9	6	Zet de filter advertentie bericht aan	De regel voor advertentie staat in de database	De regel voor advertentie staat in de database	Ja
	BER-9	7	Zet de filter mededeling bericht aan	De regel voor mededeling bericht staat in de database	De regel voor mededeling bericht staat in de database	Ja
	BER-9	8	Zet de filter mededeling bericht aan	De regel voor mededeling bericht staat in de database	De regel voor mededeling bericht staat in de database	Ja
17	BER-2	1	Stel de waardes in zoals staat beschreven in Sprint 1			Ja
	BER-2	2	Gebruik de custlogids als ontvangers en afzender welke staan omschreven in hoofdstuk 5.1			Ja

4.2 Blackbox test rapportage

Datum uitgevoerd: 2 december 2013

TestNr.	Storypoint Key	Stap	Test omschrijving	Verwacht resultaat	Daadwerkelijk resultaat	Geslaagd
18	BER-2	1	Bericht invullen zoals staat beschreven in Sprint 1			Ja
	BER-2	2	Lijst met ontvangers selecteren	Lijst met ontvangers bestaat en er zijn een aantal locaties aanwezig in de lijst	Lijst met ontvangers bestaat en er zijn een aantal locaties aanwezig in de lijst	Ja
	BER-2	3	Bericht versturen	Bericht is ontvangen bij de horeca locaties	Bericht is ontvangen bij de horeca locaties	Ja
19	BER-9	1	Portal openen voor de eerste keer	Alle filters staan op "aan"	Alle filters staan op "aan"	Ja
	BER-9	2	Zet de filter primair bericht uit	Na herladen van de pagina staat deze nog steeds uit	Na herladen van de pagina staat deze nog steeds uit	Ja
	BER-9	3	Zet de filter advertentie bericht uit	Na herladen van de pagina staat deze nog steeds uit	Na herladen van de pagina staat deze nog steeds uit	Ja
	BER-9	4	Zet de filter mededeling bericht uit	Na herladen van de pagina staat deze nog steeds uit	Na herladen van de pagina staat deze nog steeds uit	Ja
	BER-9	5	Zet de filter primair bericht uit	Na herladen van de pagina staat deze nog steeds uit	Na herladen van de pagina staat deze nog steeds uit	Ja
	BER-9	6	Zet de filter advertentie bericht aan	Na herladen van de pagina staat deze nog steeds aan	Na herladen van de pagina staat deze nog steeds aan	Ja
	BER-9	7	Zet de filter mededeling bericht aan	Na herladen van de pagina staat deze nog steeds aan	Na herladen van de pagina staat deze nog steeds aan	Ja
	BER-9	8	Zet de filter mededeling bericht aan	Na herladen van de pagina staat deze nog steeds aan	Na herladen van de pagina staat deze nog steeds aan	Ja
20	BER-37	1	Open de portal	Op de portal worden 3 pagina's weergegeven(dus het ziet er zo uit < 1 2 3 >)	Op de portal worden 3 pagina's weergegeven (Ja
	BER-37	2	Navigeer naar pagina 2	de volgende aantal berichten worden weergegeven	de volgende aantal berichten worden weergegeven	Ja
21	BER-36	1	Een advertentie bericht openen	Er staat een afbeelding van het product i.p.v. een 404 afbeelding	Er staat een afbeelding van het product i.p.v. een 404 afbeelding	Ja

5. Sprint 5

5.1 Whitebox test rapportage

Datum uitgevoerd: 2 januari 2014

TestNr.	Storypoint Key	Stap	Test omschrijving	Verwacht resultaat	Daadwerkelijk resultaat	Geslaagd
22	BER-38	1	Verstuur bericht naar account "VSV01"	Statusbericht "Ok" word teruggegeven	Statusbericht OK wordt teruggegeven	Ja

5.2 Blackbox test rapportage

Datum uitgevoerd: 2 januari 2014

TestNr.	Storypoint Key	Stap	Test omschrijving	Verwacht resultaat	Daadwerkelijk resultaat	Geslaagd
23	BER-38	1	Open de bestelbox App			Ja
	BER-38	2	Log in op de bestelbox App als VSV01			Ja
	BER-38	3	Maak een bericht aan met de gegevens uit sprint 1			Ja
	BER-38	4	Bekijk de mobiele telefoon	Er moet een bericht binnen zijn gekomen op het mobiele device	Een notificatie komt binnen met de berichtstitel als bericht	Ja