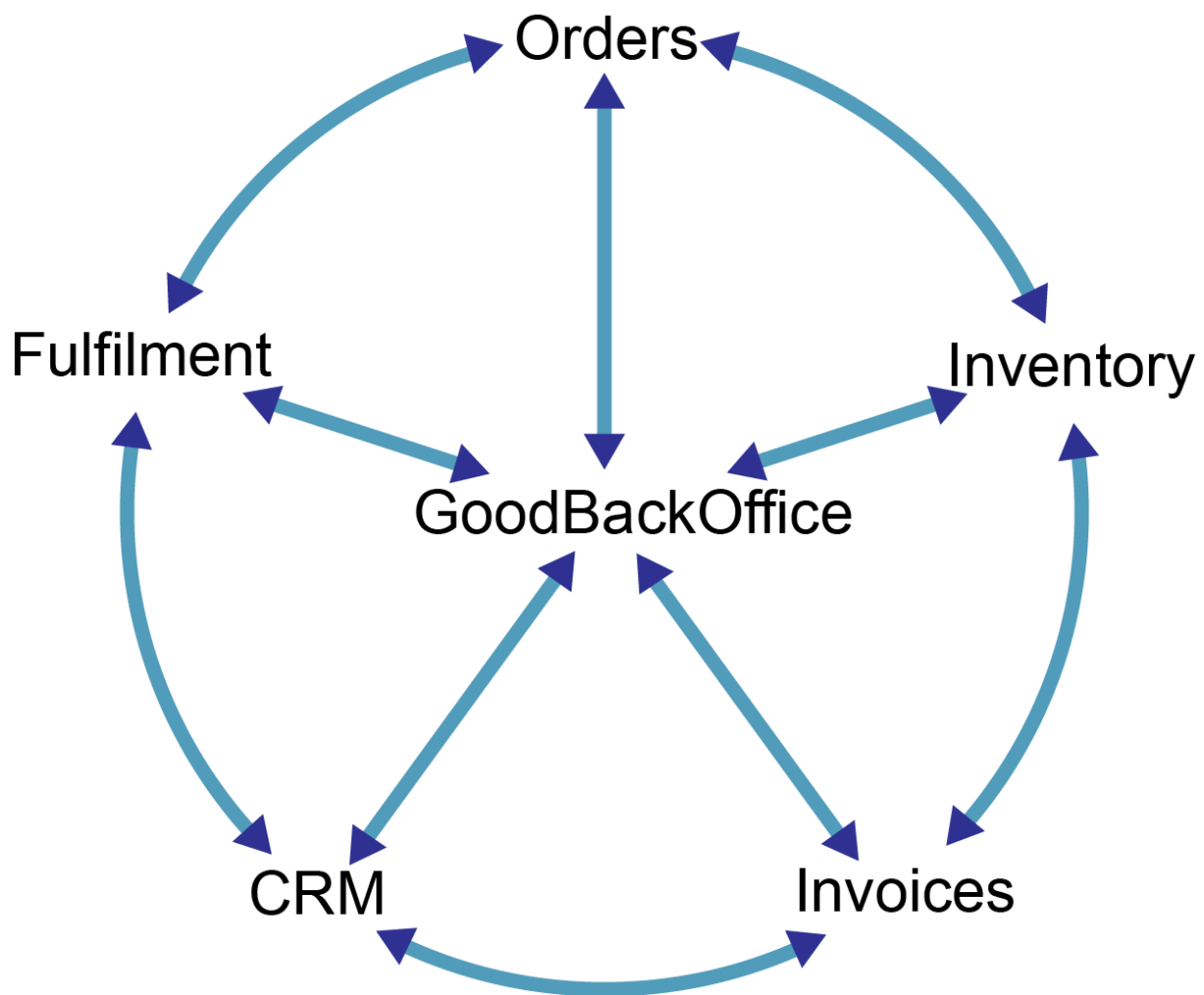


GOODBACKOFFICE

ORDER EN VOORRAAD MANAGEMENTSYSTEEM

Scriptie



Project:	GoodBackOffice
Datum:	14-10-2015
Auteur:	Ard Wijnen
Versie:	1.1
Status:	Goedgekeurd
Bedrijf:	GoodTimeGifts V.O.F.

Titelpagina

Student

Naam: Wijnen, Ard A.J.H.P.
Studentnummer: 2189396
Afstudeerrichting: Software Engineering - Deeltijd
Afstudeerperiode: Februari 2015 – Oktober 2015

Bedrijf

Naam: GoodTimeGifts VOF
Afdeling: Ontwikkeling
Adres: Oceanië 8a
Plaats: Ittervoort
Telefoon: +31 (0)47 55 64 464
Email: info@goodtimegifts.nl
Web: www.goodtimegifts.nl

Begeleiders

Fontys: Meesters, Marcel M.R.J.
Bedrijf: Henssen, Marc M.

Scriptie

Titel: GoodBackOffice, order en voorraad
managementsysteem
Datum: 14-10-2015
Locatie: Ittervsoort

Goedkeuring Versie 1.1

Bedrijfsbegeleider

Naam: Marc Henssen

Datum: 14-10-2015

A handwritten signature in black ink, appearing to read 'M. Henssen', with a long horizontal stroke extending to the right.

Paraaf:

Documenthistorie

Versie	Datum	Wijzigingen
0.1	1-9-2015	Geen wijzigingen
1.0	8-9-2015	Feedback verwerkt van Marcel Meesters
1.1	7-10-2015	Feedback verwerkt van Marcel Meesters

Voorwoord

Deze scriptie is het resultaat van mijn afstudeerperiode voor de opleiding HBO-ICT Software Engineering aan Fontys Hogeschool te Eindhoven. De afstudeeropdracht is het ontwerpen en realiseren van een nieuw order- en voorraad managementsysteem voor GoodTimeGifts genaamd GoodBackOffice, hier heb ik de afgelopen 9 maanden aan gewerkt. GoodTimeGifts is een e-commerce bedrijf en is gevestigd in Ittervoort.

De volgende personen zou ik graag willen bedanken voor hun medewerking en ondersteuning:

- Peter Wijnen: voor de opdracht en de tussentijdse besprekingen over de wensen van GoodTimeGifts
- Marc Henssen: voor de support, begeleiding en voortgang besprekingen tijdens de gehele afstudeerperiode.
- Marcel Meesters: voor het beantwoorden van vragen, het geven van nuttige tips en nakijken van alle opgeleverde producten.

Ard Wijnen

1-9-2015, Ittervoort

Inhoudsopgave

Samenvatting	7
Summary	8
Verklarende woordenlijst	9
1 Inleiding.....	11
2 Het bedrijf.....	12
3 De Opdracht.....	13
3.1 Projectdoelstellingen	13
3.2 Projectaanpak	13
3.3 Producten	14
3.4 Projectorganisatiestructuur	15
4 Het onderzoek.....	17
4.1 Hoofdvraag	17
4.2 Deelvragen	17
4.3 Uitvoering	17
4.4 Conclusie	22
5 Het ontwerp.....	24
5.1 Deployment diagram.....	24
5.2 Domeinmodel.....	25
5.3 Klassediagram	26
5.3 ERD	29
5.4 Wireframes	29
5.5 Sequence Diagrammen	35
6 Implementatie.....	38
6.1 Sprint 1	40
6.2 Sprint 2	42
6.3 Sprint 3	44
6.4 Sprint 4	44
6.5 Test.....	44
7 Conclusie en Aanbevelingen	48
Evaluatie	49
Referenties.....	50
Bijlage I	50
Bijlage II	71
Bijlage III	97

Samenvatting

GoodTimeGifts is een e-commerce bedrijf met 12 webshops, opgericht in 2005 en gevestigd in Ittervoort. GoodTimeGifts wil een nieuw order- en voorraad managementsysteem omdat het oude systeem is verouderd, niet multi-user geschikt is en enkele stukken functionaliteit mist.

Het nieuwe systeem dat gebouwd is in Java is genaamd GoodBackOffice. Het systeem moet voldoen aan de wensen en eisen van GoodTimeGifts. GoodBackOffice moet beschikken over de volgende functionaliteiten: Importeren van verkooporders, bewerken van orders, maken van facturen en beheren van de voorraad.

Tijdens het onderzoek is er gezocht naar het antwoord op de volgende hoofdvraag: 'Wat zijn de eisen voor GoodBackOffice en op welke manier kunnen deze het beste gerealiseerd worden?' Tijdens het onderzoek zijn er requirements opgesteld. Uit het onderzoek bleek dat JavaFX en SOAP webservices de meest geschikte technologieën zijn.

Voor de ontwerpfase zijn er verschillende diagrammen opgesteld. Zoals een Deploymentdiagram, een Domeindiagram, een Klassediagram, verschillende Sequence diagrammen, schetsen en wireframes van de schermen.

Daarna is de implementatiefase uitgevoerd, hierin is de applicatie ontwikkeld met behulp van een variatie op de Scrum ontwikkelmethode. Er zijn in totaal vier sprints doorlopen van elk twee weken. Tijdens de sprints is er veel overleg geweest tussen de ontwikkelaar en de opdrachtgever. Na elke sprint is er een demonstratie gehouden van de gemaakte functionaliteit voor de opdrachtgever en eindgebruiker. Aan het einde van de ontwikkelperiode is de applicatie geaccepteerd door de opdrachtgever tijdens een acceptatie test.

Aanbevolen wordt dat er in de toekomst verder ontwikkeld wordt aan GoodBackOffice. Er wordt aanbevolen dat er dan een nieuw onderzoek- en ontwerpfase plaatsvindt, waarin het huidige onderzoek en ontwerp worden aangevuld.

Summary

GoodTimeGifts is an e-commerce company with twelve webshops founded in 2005 in Ittervoort. GoodTimeGifts wants a new order and stock management system called GoodBackOffice because the old system has become obsolete, is not multi user friendly and misses important functionality.

GoodTimeGifts wants GoodBackOffice to be built in Java. The system needs to meet the requirements of GoodTimeGifts. GoodBackOffice needs to have the following functions: Import orders, edit orders, create invoices and manage stock.

During the research period an answer was sought for the main question: What are the requirements for GoodBackOffice and how can it be realized best? Requirements were drawn up during the research period. The research showed that JavaFX and SOAP webservices were best suited for GoodBackOffice.

A number of diagrams were drawn up during the design period. A deployment diagram, a domain diagram and a class diagram were drawn up along with several wireframes and sequence diagrams.

The implementation period followed there after. During this de application was developed using a variation of the scrum development method. In total 4 sprints were run through of each 2 weeks. There was a lot of consultation between the developer and the client during the sprints. After each sprint a demo was held for the client and end-user. At the end of the development period an acceptance test was done by the client. During this test the client accepted the developed product.

It is recommended that GoodBackOffice be further developed in the future. When GoodBackOffice is being developed further it is recommend that the research and design are expanded.

Verklarende woordenlijst

Cross-platform	Een toepassing op een computer heet cross-platform wanneer het op meer dan één systeemplatform werkt, bijvoorbeeld met Linux, OSX en Windows.
Datepicker	Een datepicker is een Gebruikers interface element dat het makkelijk maakt om een datum te selecteren in een kalender.
Deploymentdiagram	Een deployment diagram beschrijft de relatie tussen de hardware en software van een systeem.
Domeindiagram	Een domein model in software engineering is een conceptueel model van het domein dat zowel het gedrag en de data bevat.
ERD	Het entity-relationship diagram (ERD) is een diagram voor het inzichtelijk te maken van een conceptueel datamodel. Het is een visuele weergave van de entiteiten, relaties en regels die gelden of aanwezig zijn in een logisch ontwerp.
Gebruikersinterface	Een gebruikersinterface is de interface tussen een computer en de mens die de computer gebruikt. De gebruikersinterface maakt interactie tussen mens en machine mogelijk.
JavaFX	JavaFX is een set van grafische en media-pakketten die ontwikkelaars in staat stelt om te ontwerpen, maken, testen, debuggen en implementeren van client toepassingen die consistent werken op diverse platforms.
Macro	Een macro is een klein computerprogrammaatje dat gedefinieerd is binnen een ander programma. De taak van een macro is om de werking (of de gebruiker) van het andere programma te ondersteunen, bijvoorbeeld door de automatisering van een aantal handelingen.
Multi user	Multi-user software is software die de toegang door meerdere gebruikers van een computer mogelijk maakt.
Parcelware	Parcelware is een applicatie van Postnl die het mogelijk maakt verzendlabels aan te maken.
Remote communicatie technologie	Communicatie methodes tussen losstaande applicaties
Requirement	In een requirement worden de benodigde attributen, capaciteiten, karakteristieken of kwaliteiten van een systeem geïdentificeerd, die bruikbaar zijn en meerwaarde bieden voor een gebruiker.
RMI	Java Remote Method Invocation is een remote communicatie technologie die communicatie tussen Java applicaties mogelijk maakt.

RPC	Een remote procedure call, kortweg RPC, is een technologie die een computerprogramma op één bepaalde computer toestaat om code uit te voeren op een andere machine zonder dat de programmeur de code expliciet hiervoor geschreven heeft.
Scrum	Scrum is een flexibele manier om (software)producten te maken.
Scene Builder	JavaFX Scene Builder is een visuele omgeving waarmee gebruikersinterfaces (UI) voor JavaFX applicaties ontworpen kunnen worden zonder enige code te schrijven.
Sequence diagram	Een Sequence diagram is een interactie grafiek die laat zien hoe processen met elkaar functioneren en in welke volgorde.
SOAP	SOAP (Simple Object Access Protocol) is een computerprotocol dat wordt gebruikt voor communicatie tussen verschillende componenten.
Sprint	Binnen SCRUM is een sprint een periode van 1 tot 4 weken waarin men een aantal zaken van de productbacklog oppakt. Dit is een time-box.
Storypoint	Een Storypoint is een aanduiding van tijd en moeilijkheidsgraad in punten voor een requirement
Swing	Swing is de API voor het maken van grafische gebruikersinterfaces in Java
Trello	Trello is een web-based project management applicatie
Webservice	Een webservice kan omschreven worden als een interface van een applicatiecomponent die toegankelijk is via standaard webprotocollen en waarbij meestal wordt gecommuniceerd via XML zonder menselijke tussenkomst (bijvoorbeeld SOAP).
Wireframes	Een wireframe is een illustratie van een gebruikersinterface.

1 Inleiding

GoodTimeGifts is een e-commerce bedrijf dat is opgericht in 2005 en is gevestigd in Ittervoort. Op dit moment heeft GoodTimeGifts 12 verschillende webshops in verschillende productcategorieën. GoodTimeGifts wil een nieuw order- en voorraad managementsysteem genaamd GoodBackOffice omdat het oude systeem is verouderd, niet multi-user geschikt is en enkele stukken functionaliteit mist. De opzet en uitwerking van dit project wordt in dit document toegelicht. Het project is opgedeeld in de volgende fases: Onderzoek, Ontwerp, Implementatie, Oplevering en Afronding.

De volgende hoofdstukken worden behandeld in deze scriptie:

- **Het bedrijf:** Dit hoofdstuk geeft achtergrond informatie over GoodTimeGifts en de opdracht.
- **De opdracht:** Hierin wordt de opdracht en aanpak uitgebreid omschreven.
- **Het onderzoek:** Beschrijving van het uitgevoerde onderzoek met resultaten en conclusie.
- **Het ontwerp:** Dit hoofdstuk beschrijft de gemaakte keuzes tijdens de ontwerpfase van het project.
- **Implementatie:** Hierin worden de gemaakte keuzes tijdens de Implementatiefase beschreven met bijbehorende argumentatie.
- **Conclusie:** Hierin wordt het project afgerond.
- **Aanbevelingen:** Dit hoofdstuk beschrijft de aanbevelingen voor de toekomst van GoodBackOffice.
- **Evaluatie:** In dit hoofdstuk wordt reflectie gegeven over het project en de uitvoering

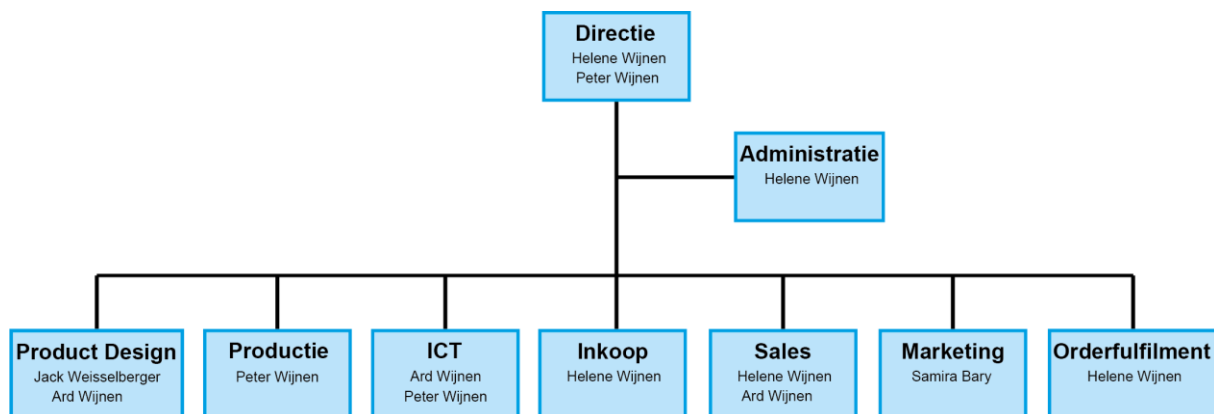
2 Het bedrijf

GoodTimeGifts is een e-commerce bedrijf met 12 webshops, in verschillende productcategorieën zoals klokken, waxjassen, kapstokken en keukengerei. Het bedrijf is opgericht in 2005 en is gevestigd in Ittervoort. GoodTimeGifts verkoopt hun producten via verschillende verkoopkanalen: eigen webshops, Bol.com en Amazon.de.

Het huidige order- en voorraad managementsysteem is inmiddels flink verouderd en heeft enkele nadelen. Zo is het niet mogelijk om met meerdere personen het systeem tegelijk te gebruiken. Er missen ook verschillende stukken functionaliteit zoals het maken van een creditnota. GoodTimeGifts heeft besloten om zelf een nieuw order- en voorraad managementsysteem, "GoodBackoffice", te maken om deze problemen op te lossen. Het oude systeem is gebouwd in Microsoft Excel en Outlook met verschillende macro's.

Mijn positie binnen het bedrijf

Ik ben begonnen bij GoodTimeGifts in januari 2011. In mijn rol als productdesigner ontwerp ik klokken die door de productie gemaakt worden. In mijn rol als salesmedewerker zorg ik ervoor dat de websites up-to-date blijven. In mijn rol als ICT 'er ontwikkel ik aan de websites en nu ook aan het nieuwe order- en voorraad managementsysteem. In figuur 1 is de gehele organisatie afgebeeld in een organigram.



Figuur 1 Organigram

3 De Opdracht

Dit hoofdstuk is gebaseerd op het PID verslag (bijlage I).

GoodTimeGifts heeft ervoor gekozen om het systeem in Java te bouwen. Dit zodat de gebruikers ongeacht welk systeem(Windows, OSX) de applicatie kunnen gebruiken en omdat er al ervaring met Java is binnen GoodTimeGifts. Verder moet het systeem multi user zijn. Het systeem moet dus meerdere gebruikers tegelijk aankunnen en zorgen dat deze gebruikers dezelfde informatie kunnen zien. Er zal een database moeten worden gerealiseerd om de gegevens op te kunnen slaan. Daarnaast moet ervoor gezorgd worden dat de applicatie snel werkt, dit wordt gecontroleerd volgens de hiervoor opgestelde performance eisen. Bij het maken van het ontwerp zal er rekening gehouden moeten worden met de performance.

Binnen de scope van het afstudeerproject vallen de volgende functionaliteiten:

- Importeren van verkooporders;
- Maken van facturen;
- Beheren van de voorraad.

Buiten de scope het afstudeerproject vallen de volgende functionaliteiten:

- Maken van creditnota's;
- Exporteren orderstatus;
- Updates orderstatus versturen naar klant;
- Order fulfilment;
- CRM;
- Inkoop.

Er is gekozen voor deze verdeling omdat in de scope de belangrijkste functionaliteit zit. De functionaliteit binnen de scope zorgt voor een goede basis, wanneer deze af is zal de applicatie goed uit te breiden zijn met de functionaliteit die tijdens dit project buiten de scope gevallen is.

3.1 Projectdoelstellingen

Het doel is dat GoodBackOffice aan het einde van het project beschikt over de gewenste functionaliteit.

3.2 Projectaanpak

Het project is opgedeeld in de volgende 5 fases:

- **Onderzoek**

Tijdens het onderzoek wordt de volgende onderzoeksvraag beantwoord: Wat zijn de eisen voor GoodBackOffice en op welke manier kunnen deze het beste gerealiseerd worden?

- **Ontwerp**

Het ontwerp door middel van het opstellen van verschillende diagrammen(ERD, Domeindigram, Deploymentdiagram) en Wireframes.

- **Implementatie**

Tijdens de implementatiefase wordt de daadwerkelijke applicatie gemaakt. Dit gebeurt met behulp van een variatie op de Scrum ontwikkelmethode (Schwaber & Sutherland, 2013).

- **Oplevering**

Deze fase begint met een grote test waarin het gehele applicatie getest wordt. Ook vindt er een acceptatietest plaats met de opdrachtgever. Indien de opdrachtgever de applicatie accepteert wordt de applicatie opgeleverd.

- **Afronding**

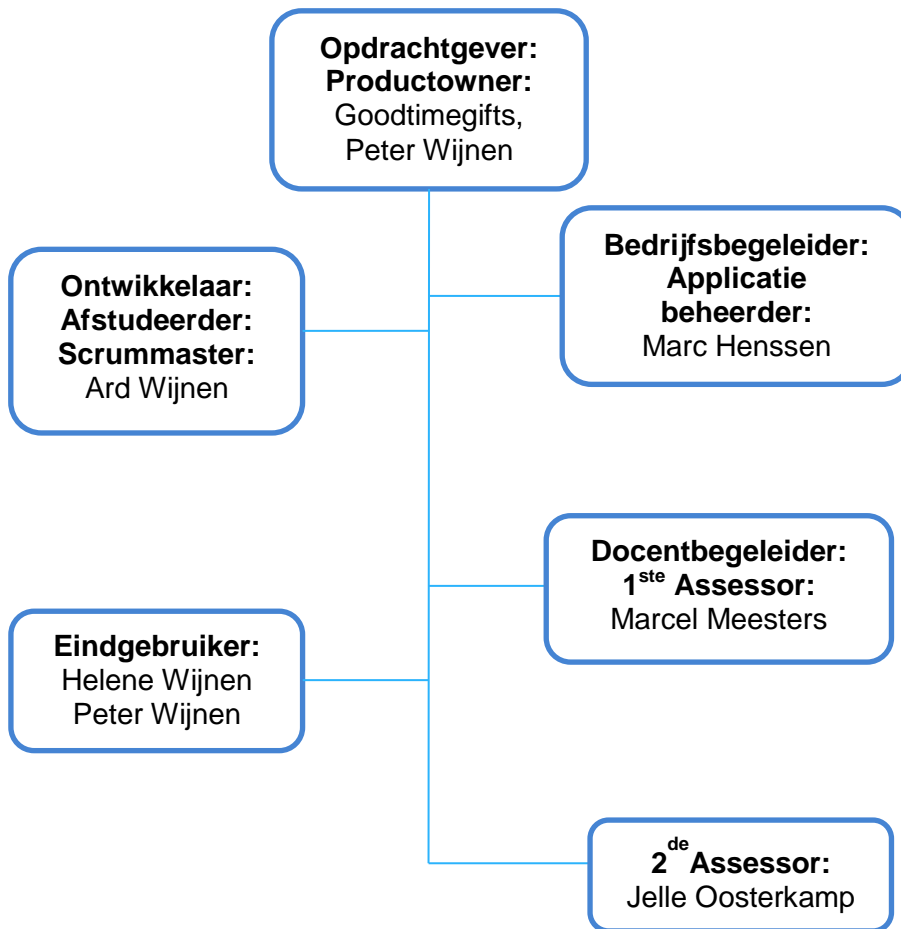
Tijdens deze fase zal de scriptie afgerond worden en ingeleverd worden. De eindpresentatie zal ook in deze fase gegeven worden. De fase wordt afgesloten met een reflectie over het gehele project.

3.3 Producten

Het project is opgedeeld in de volgende hoofdproducten en tussenproducten:

- Projectdocumentatie
 - Scriptie
 - PID
 - Onderzoekverslag
- Applicatie documentatie
 - Test rapportage
- Ontwerp
- Applicatie
 - Factuur Management
 - Order Management
 - Voorraad Management

3.4 Projectorganisatiestructuur



Figuur 2 Projectorganisatiestructuur

Opdrachtgever, Productowner

De opdrachtgever heeft het project bedacht en gecreëerd. Hij overziet het project en zorgt ervoor dat het eindresultaat voldoet aan de eisen en overeenkomt met wat origineel was gepland. Wijzigingen aan het project worden overlegd met de opdrachtgever. In de rol als productowner doet hij mee aan de dagelijkse stand-up meetings, en mee bepalen welke requirements in de sprints meegenomen worden.

Ontwikkelaar, Afstudeerder, Scrummaster

De afstudeerder is verantwoordelijk voor het uitvoeren van het gehele project. Hij is verantwoordelijk voor het inplannen en regelen van afspraken met alle eindgebruikers. Zoals de interviews met de opdrachtgever en eindgebruikers om de wensen, eisen en huidige werkwijzen in kaart te brengen. In de rol als scrummaster moet hij ervoor zorgen dat het scrumproces correct verloopt.

Eindgebruiker

De eindgebruikers zijn de personen die de applicatie in productie gaan nemen wanneer deze voltooid is. Zij moeten de afstudeerder zo goed mogelijk informeren over de wensen, tevens moeten zij aanwezig zijn bij de tweewekelijkse demonstratie.

Bedrijfsbegeleider, Applicatiebeheerder

De bedrijfsbegeleider is het eerste aanspreekpunt voor vragen en feedback over het project vanuit de afstudeerder. Hij adviseert de docentbegeleider over de beoordelingen van de afstudeerproducten. De functie van de applicatiebeheerder zal pas ingaan als de gehele applicatie voltooid is. Er is tijdens het onderzoek met de applicatiebeheerder gesproken over de wensen betreffende de configuratie.

Docentbegeleider, 1^{ste} Assessor

De docentbegeleider is het tweede aanspreekpunt voor vragen en feedback over het project vanuit de afstudeerder. Hij beoordeelt samen met de 2^{de} assessor de afstudeerproducten. De docentbegeleider begeleidt de afstudeerder ook tijdens het schrijven van de scriptie.

2^{de} Assessor

De 2^{de} assessor beoordeelt samen met de 1^{ste} assessor de afstudeerproducten.

4 Het onderzoek

Dit hoofdstuk is gebaseerd op het onderzoeksverslag (bijlage II).

4.1 Hoofdvraag

Het onderzoek wordt uitgevoerd om antwoord te krijgen op de volgende hoofdvraag:

Wat zijn de eisen voor GoodBackOffice en op welke manier kunnen deze het beste gerealiseerd worden?

4.2 Deelvragen

De hoofdvraag wordt beantwoord door middel van het onderzoeken van de volgende deelvragen:

1. Hoe ziet het oude systeem eruit, zowel de architectuur als de gebruikte werkwijze?
2. Wat zijn de functionele systeemeisen voor het nieuwe systeem?
3. Wat zijn de performance eisen voor het nieuwe systeem?
4. Wat zijn de gebruikersinterface eisen voor het nieuwe systeem?
5. Welke bestaande oplossingen zijn er die aan de criteria en eisen voldoen?
6. Welke gebruikersinterface technologieën moeten er gebruikt worden om aan de eisen te voldoen?
7. Welke remote technologieën moeten er gebruikt worden om aan de eisen te voldoen?
8. Zijn de performance eisen haalbaar?

4.3 Uitvoering

Het oude systeem is onderzocht door middel van interviews met de eindgebruikers en opdrachtgever. Ook is er meegekeken hoe de applicatie gebruikt wordt. Tijdens het interview zijn er open vragen gesteld zodat de geïnterviewde ook extra informatie konden vertellen en hierop doorgevraagd kon worden. Per onderdeel zijn vragen gesteld zoals:

- Kun je kort het proces beschrijven?
- Welke overzichten zijn er beschikbaar?
- Waar liggen de beslissingsmomenten?
- Zijn er dingen die sneller of beter kunnen in jouw ogen?

Toen er een duidelijk beeld was van het oude systeem zijn er interviews gehouden met de eindgebruikers en opdrachtgever om erachter te komen wat de wensen en eisen zijn voor GoodBackOffice. Tijdens het interview zijn er open vragen gesteld zodat de geïnterviewde ook extra informatie konden vertellen en hierop doorgevraagd kon worden. Tijdens het interview zijn er vragen gesteld als:

- Vanuit hoeveel verschillende webshop systemen moeten orders geïmporteerd worden?
- Welke gegevens moeten op de factuur staan?
- Voor welke onderdelen zijn er performance eisen en wat is hierbij de gestelde performance eis?
- Welke overzichten moeten er komen?

Tijdens het onderzoek waarin er gekeken werd of er bestaande oplossingen waren die aan de criteria en eisen voldeden, is er eerst gezocht naar welke beschikbare oplossingen er zijn. Daarna is er een checklijst opgesteld met de hoofd requirements van het systeem (figuur 3).

Pakketten / Hoofd Criteria (Max. Punten)	Unit 4	Scherpa	Microsoft Dynamics NAV	Embedded Erp	Suite commerce
1. Het pakket beschikt over een order management gedeelte.(50)	50	50	50	50	50
2. Het pakket beschikt over een factuur management gedeelte.(50)	50	50	50	50	50
3. Het pakket beschikt over een voorraad beheer gedeelte.(50)	50	50	50	50	50
4. Het pakket beschikt over een bestaande koppeling met Magento.(20)	20	20	20	20	20
5. Het pakket beschikt over een bestaande koppeling met Joomla Virtuemart.(20)	0	0	0	0	0
6. Het pakket werkt cross-platform.(40)	40	0	40	40	40
7. Het pakket is multi-user.(20)	20	20	20	20	20
8. Het pakket beschikt over een bestaande Parcelware integratie.(10)	0	10	0	0	0
9. Het pakket moet updates versturen naar klant.(10)	10	10	10	10	10
10. Het pakket moet status updates versturen naar Magento.(10)	10	10	10	10	10
11. Het pakket moet status updates versturen naar Joomla Virtuemart.(10)	0	0	0	0	0
12. Prijsindicatie zonder aanpassingen (eenmalig / per jaar).	3500 euro/jaar	7620 euro/jaar + 3100 eenmalig	1536 euro/jaar + 3500 euro eenmalig	1340 euro eenmalig	2000 euro/jaar
Prijs in punten(100)	5	0	35	100	20
Totaal punten (390)	255	220	285	350	270

Figuur 3 Bestaande applicaties

Tijdens het onderzoek welke gebruikers interface technologie is er een checklijst (figuur 5) en pros-and-cons lijst (figuur 4) opgesteld. JavaFX is in vergelijking met Swing een veel nieuwere techniek (Weaver, Gao, Chin, Iverson, & Vos, 2014). In de toekomst komen er nog functionele uitbreidingen in JavaFX (JavaFX-Jira.kenai, 2015), Swing wordt niet meer uitgebreid. Doordat JavaFX de nieuwere techniek is zitten er nog veel bugs in. Wel staan veel van deze bugs in de planning om vóór de Java SE 9 release opgelost te worden (JavaFX-Jira.kenai, 2015). JavaFX beschikt over meer functionaliteit dan Swing. Een voordeel van JavaFX is de beschikking van Touch integratie. Swing is momenteel nog de meest gebruikte technologie. Voor beide technologieën is er nog volledige support. Voor Swing zal de support eerder stoppen dan voor JavaFX. De ontwikkelsnelheid van JavaFX zal iets lager liggen dan die voor Swing door de steilere leercurve (Stafford, 2013).

JavaFX	Swing
Pro	Pro
Langere support termijn	Momenteel meest gebruikt
Uitgebreidere functionaliteit	Hogere ontwikkelsnelheid
Uitbreidingen in de toekomst	
Con	Con
Meer bugs	Korte support termijn
Lagere ontwikkelsnelheid	Minder functionaliteit
	Wordt niet meer uitgebreid

Figuur 4 pros and cons lijst JavaFX & Swing

Onderdeel	JavaFX	Swing
Buttons	X	X
Tabel met slider	X	X
Tickboxes	X	X
Pagination	X	-
Text-edit-single-line-field	X	X
Multiple-line-textfield	X	X
Single-line-textfield	X	X
Text-edit-multiple-line-field	X	X
Dropdown-selector	X	X
Radio-buttons	X	X
Datum-picker	X	-
Progress-indicator	X	X

Figuur 5 checklist JavaFX & Swing

Tijdens het onderzoek van deelvraag 7 is er gekozen tussen RMI, SOAP webservices en REST webservices. RMI is een techniek speciaal voor Java applicaties, hierdoor is RMI taal afhankelijk, het kan alleen maar in combinatie met Java gebruikt worden. Webservices zijn taal onafhankelijk wat het mogelijk maakt om twee applicaties die in een andere taal geschreven zijn met elkaar te laten communiceren. Webservices zijn dus beter geschikt als het gaat om Cross-platform (Englander, 2002). Met webservices is het zowel mogelijk om asynchrone communicatie als synchrone communicatie te hebben, met RMI is alleen synchrone communicatie mogelijk. Uit onderzoek blijkt dat RMI sneller is (Techannotation, 2012). Dit komt doordat het serialiseren en deserialiseren sneller gaat met de binaire data representaties van RMI dan met XML wat in SOAP webservices wordt gebruikt (Harold, 2004). Bij REST webservices kan er geserialiseerd worden naar XML of JSON. Tussen

REST en SOAP zitten niet veel verschillen behalve dat SOAP beter bekend is binnen de organisatie.

Onderdeel	SOAP Webservices	REST Webservices	RMI
Taal onafhankelijk	X	X	-
Synchrone communicatie	X	X	X
Asynchrone communicatie	X	X	-
Bekend binnen GoodTimeGifts	X	-	-

Figuur 6 SOAP & REST & RMI

Toen duidelijk was welke technologieën er het beste gebruik konden worden is het onderzoek voor deelvraag 8 gestart. Er is toen eerst een gedeelte van het ontwerp gemaakt. Hierna is een testapplicatie gemaakt op basis van het ontwerp. Met deze testapplicatie zijn toen de metingen uitgevoerd. Tijdens dit onderzoek zijn er twee testen uitgevoerd: test 1: Het laden van 100 orders in de ordertabel en test 2: Het zoeken in alle orders.

Test 1: Het laden van 100 orders in de ordertabel.

Stappen:

1. De client applicatie start de aanvraag.
2. De tijdsmeting wordt gestart.
3. De client applicatie roept de Webservice methode van de server applicatie aan.
4. De server applicatie haalt 100 Random OrderOverview records uit de tabel.
5. De server applicatie maakt 100 OrderOverview objecten aan en zet deze in een ArrayList.
6. De server applicatie plaats de ArrayList in een ArrayListContainer object.
7. De server applicatie retourneert het ArrayListContainer object aan de client applicatie.
8. De client applicatie vult de GUI tabel.
9. De tijdsmeting wordt gestopt.

In figuur 7 zijn de metingen te zien in milliseconden

Meting 1	239
Meting 2	142
Meting 3	135
Meting 4	132
Meting 5	154
Meting 6	116
Meting 7	126
Meting 8	108
Meting 9	131
Meting 10	112
Meting 11	111
Meting 12	109
Meting 13	102
Meting 14	106
Meting 15	106

Meting 16	122
Meting 17	114
Meting 18	124
Meting 19	100
Meting 20	93
Meting 21	123
Meting 22	108
Meting 23	102
Meting 24	92
Meting 25	115
Meting 26	91
Meting 27	100
Meting 28	112
Meting 29	107
Meting 30	101

Figuur 7 Performance meting

De meting is in totaal 30 keer uitgevoerd. Met een gemiddelde tijd van 117 milliseconden. Uit de metingen is af te lezen dat de eerste aanvraag langer duurt (meting 1). Dit komt doordat er voor de eerste keer connectie met de webservice gemaakt. Bij de metingen hierna wordt de connectie gebruikt. De test is meerdere keren herhaald waarbij tussen elke meting de client applicatie werd afgesloten. Hierbij kwam een gemiddelde tijd van 275 milliseconden.

Test 2: Het zoeken in alle orders.

Stappen:

1. De client applicatie start de aanvraag.
2. De tijdsmeting wordt gestart.
3. De client applicatie roept de Webservice methode van de server applicatie aan.
4. De server applicatie zoekt in de OrderOverview tabel naar de juiste records.
5. De server applicatie maakt een aantal OrderOverview objecten aan, maximaal 100, en zet deze in een ArrayList.
6. De server applicatie plaats de ArrayList in een ArrayListContainer object.
7. De server applicatie retourneert het ArrayListContainer object aan de client applicatie.
8. De client applicatie vult de GUI tabel.
9. De tijdsmeting wordt gestopt.

Als variabelen zijn het totaal aantal records in de tabel meegenomen en het aantal records die de zoek query retourneert (figuur 8).

	Totaal aantal records						
Aantal records die voldoen aan de zoek query		2500	5000	10000	15000	20000	25000
	10	120	110	115	130	131	135
	20	125	130	132	129	142	138
	40	134	133	142	147	150	149
	80	150	153	159	152	158	163
	160	162	159	159	168	165	168
	320	156	149	168	172	142	168
	640	145	157	165	166	149	156
	1280	152	163	149	181	176	165
	2560	x	145	159	165	162	178
	5120	x	x	143	136	146	153
	10240	x	x	x	132	156	149
	20480	x	x	x	x	x	146

Figuur 8 Database performance test

Elke waarde in de tabel is een gemiddelde van tien tests in milliseconden. Uit de waarde van de tabel is af te lezen dat het aantal records die voldoen aan de zoek query alleen verschillen als dit minder als 100 is. Het totaal aantal records in de database maakt geen verschil uit.

4.4 Conclusie

Uit het onderzoek zijn de volgende conclusies gekomen.

Het huidige systeem bestaat uit vier gedeeltes: Outlook, Excel, Parcelware en Administratie. De gebruikte architectuur en werkwijze zijn afgebeeld in figuur 1 tot en met figuur 6 in bijlage II.

Er zijn 55 systeemeisen opgenomen tijdens het onderzoek, hiervan zijn er 39 functionele requirements, drie performance requirements en dertien gebruikersinterface requirements.

Verder is er tijdens het onderzoek gekeken naar vijf mogelijk te gebruiken bestaande software pakketten. Na onderzoek op basis van hoofd requirements en een gesprek met de opdrachtgever is er gekozen om niet een bestaande applicatie te gebruiken. De reden hiervan is dat vier van de vijf pakketten boven het budget lagen, alleen Embedded ERP lag binnen het budget. Er is niet gekozen voor dit pakket omdat het geïntegreerd zit met Magento waardoor er in de toekomst niet meer af te stappen is van Magento.

Voor deelvraag 6 is er gekeken welke gebruikersinterface technologie het beste aan de gestelde eisen voldoet. Nadat JavaFX en Swing met elkaar zijn vergeleken is er gekozen om JavaFX te gebruiken voor de applicatie. Dit omdat JavaFX meer toekomstgericht is en alle gewenste onderdelen bevat. Een nadeel van JavaFX is dat het een groot aantal bugs heeft, dit zal geen probleem zijn doordat slechts de standaard functionaliteit gebruikt zal worden.

Bij het kiezen voor de meest geschikte remote communicatie technologie zijn SOAP webservices, REST webservices en RMI bekeken. De keuze is gevallen op SOAP webservices omdat deze technologie beter cross-platform geschikt is dan RMI. Er is niet voor REST gekozen omdat deze minder bekend is binnen de organisatie. Uit onderzoek bleek Webservices iets langzamer te zijn dan RMI. Dit zou een probleem kunnen zijn, maar de gestelde performance eisen zijn ook met webservices goed te halen.

Om te kijken of de performance eisen haalbaar zijn met het beoogde ontwerp zijn er diverse metingen gedaan. Uit de verschillende metingen is gebleken dat de performance met het ontwerp haalbaar zijn. De gemiddelde meting kwam uit op 117 milliseconden en de eis was drie seconden.

5 Het ontwerp

De ontwerpfase is gedeeltelijk parallel aan de onderzoeksfase uitgevoerd. Toen een groot gedeelte van het ontwerp klaar was is er onderzoek gedaan of de performance eisen haalbaar waren. Om deze vraag te kunnen beantwoorden is er een testapplicatie gemaakt. Toen ook deze deelvraag beantwoord was is het ontwerp verder voltooid. De testapplicatie kon daarna grotendeels gebruikt worden voor het begin van de applicatie.

De conclusies uit het onderzoek zijn ook meegenomen in de ontwerpfase. Zo is als remote communicatie technologie voor SOAP webservices gebruikt, dit was ook de aanbeveling die voort is gekomen uit het onderzoek. De aanbeveling om JavaFX als gebruikersinterface technologie te gebruiken is ook toegepast. Verder is er met alle requirements die voort kwamen uit het onderzoek rekening gehouden.

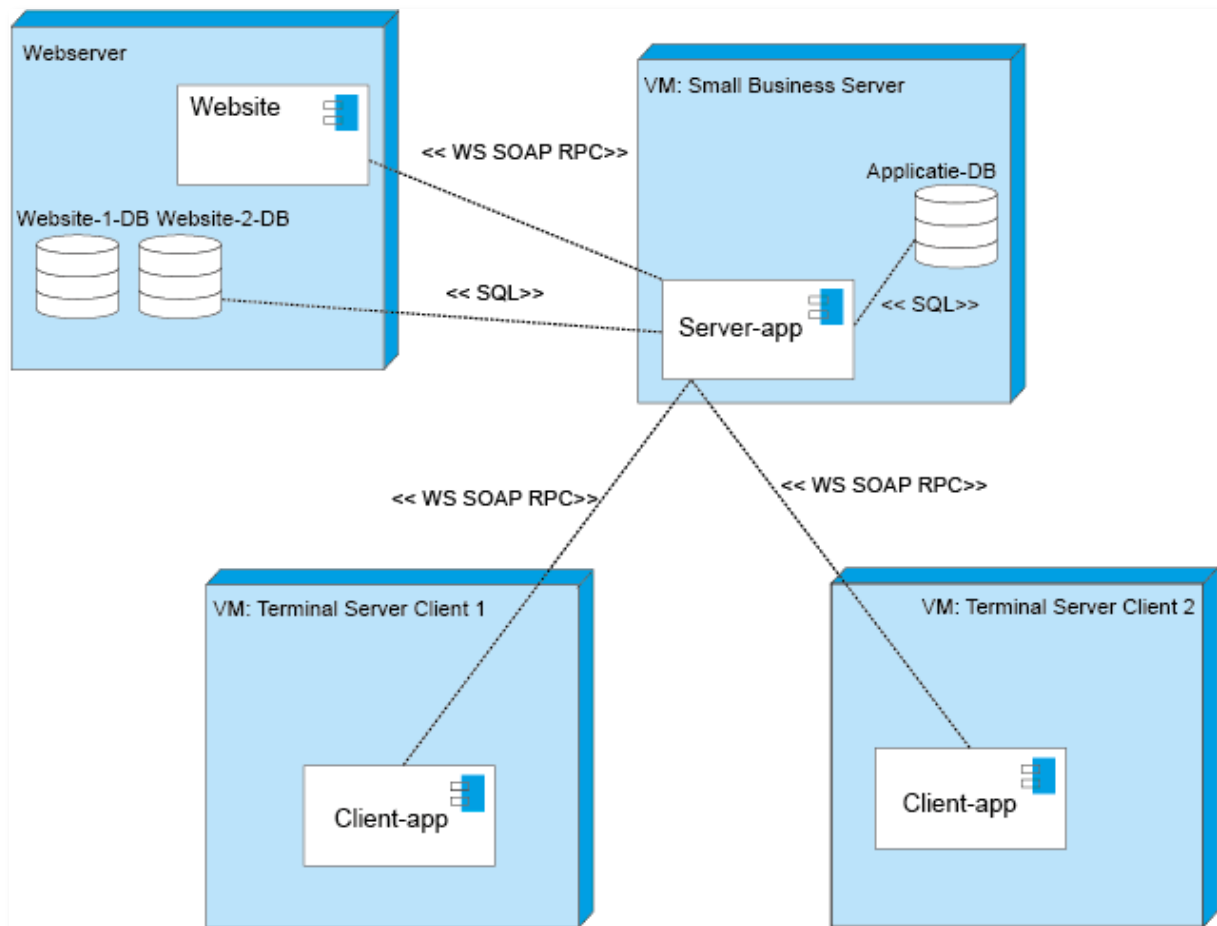
5.1 Deploymentdiagram

GoodBackOffice bestaat uit drie gedeeltes.

- Website gedeelte
- GBOServerApp
- GBOClientApp

Wanneer er via een webshop een order geplaatst wordt, voert de website een RPC uit bij de GBOServerApp. Deze gaat vervolgens de gegevens ophalen uit de databases van de websites.

Vanuit de GBOClientApp worden de gegevens uit de database van de GBOServerApp opgehaald. Dit is terug te zien in het Deploymentdiagram (figuur 9). De webserver draait bij een externe hosting provider.

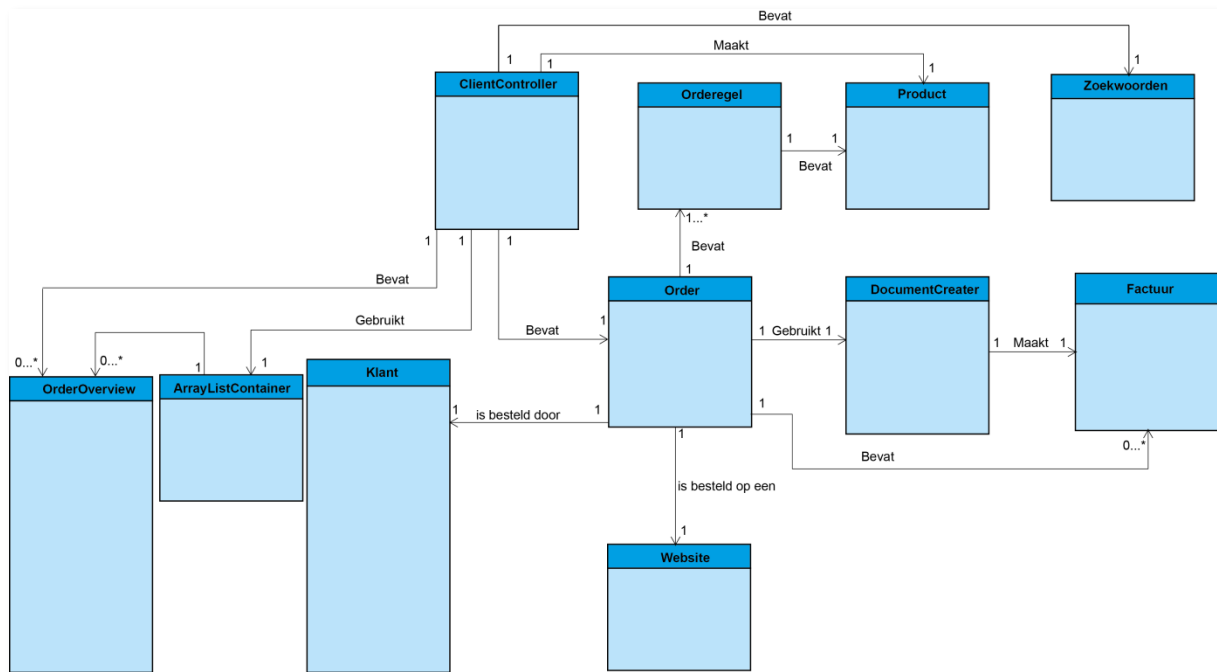


Figuur 9 Deploymentdiagram

De splitsing is gedaan ten behoeven van de veiligheid. Zo is het van buitenaf niet mogelijk om via RPC ordergegevens op te vragen uit GoodBackOffice. Ook kan de toegang tot het aanroepen van de website-GBOServerApp webservice verder geblokkeerd worden door alleen het IP-adres van de webserver toe te laten tot de webservice binnen het lokale netwerk.

5.2 Domeinmodel

GoodBackOffice is opgedeeld in elf domeinklassen. In de ArrayListContainer zit een lijst met één of meerdere OrderOverviews. De DocumentCreator kan nu alleen facturen maken later wordt dit met pakbonnen, verzendlabels en eventueel andere documenten aangevuld. De relaties tussen de verschillende klassen zijn te zien in figuur 10.

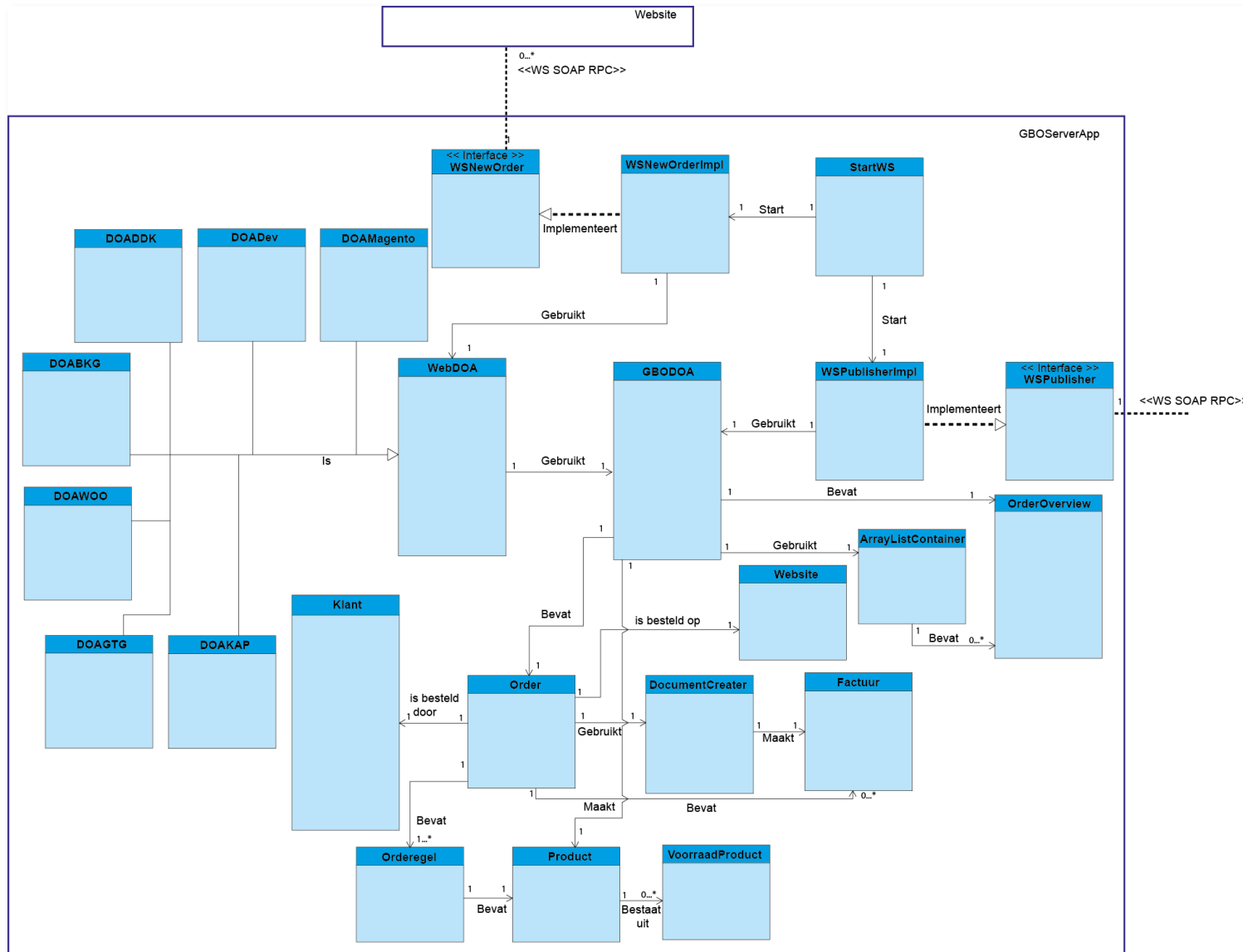


Figuur 10 Domeinmodel

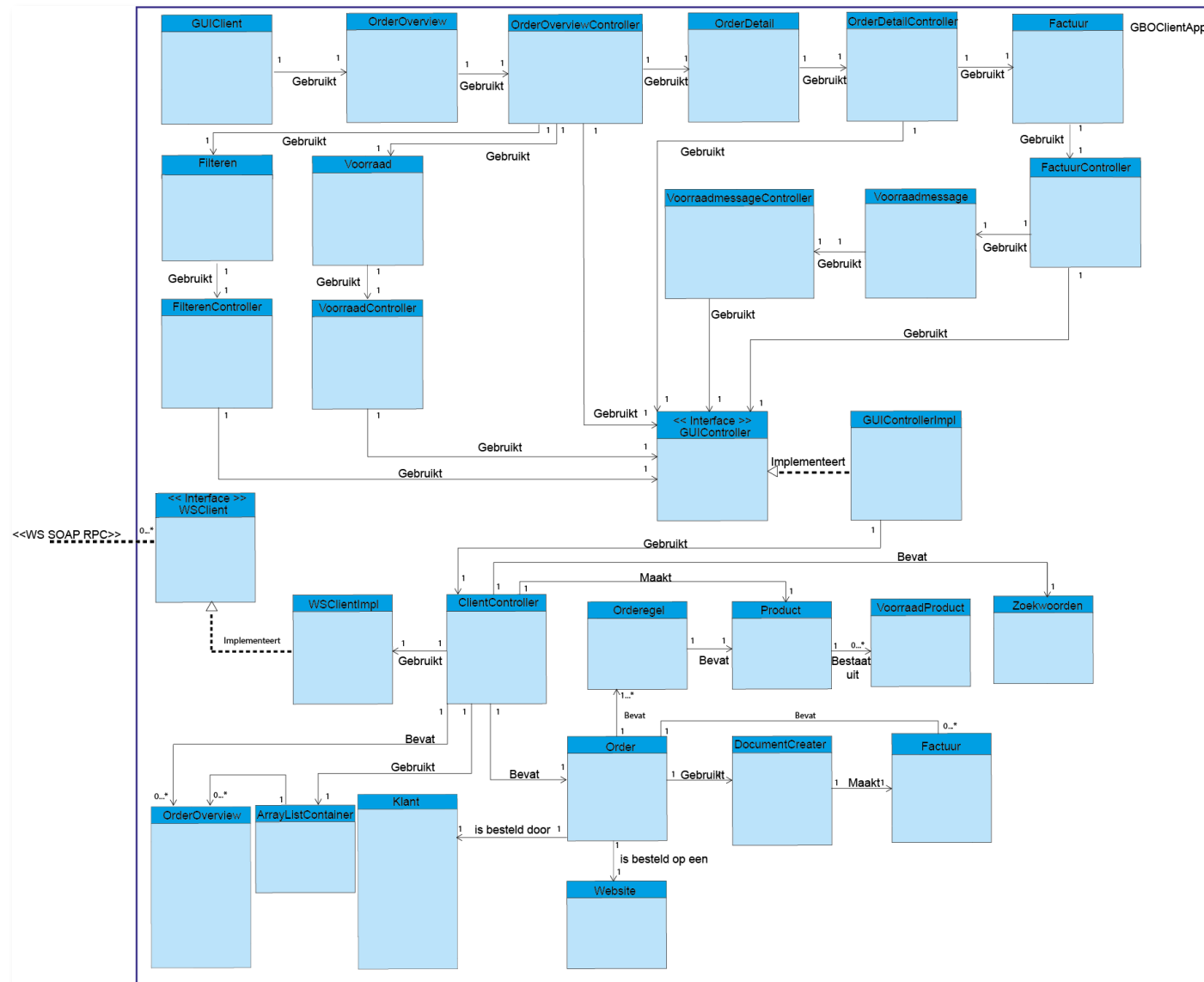
Tijdens de ontwerpfase is er gekozen om zowel OrderOverview objecten als Order, Klant en andere objecten te gebruiken. Dit is gedaan omdat het op deze manier sneller is om een overzicht van orders te laten zien. Zo hoeft er bij aan de aanvraag van een nieuw overzicht slechts 100 OrderOverview objecten aangemaakt en gevuld te worden. Terwijl dit er anders 100 keer Order, website, DocumentCreator, Factuur, OrderRegel en Producten objecten aangemaakt en gevuld dienen te worden. Het OrderOverview is speciaal gemaakt om enkel gegevens te laten zien. Hierin zijn ook verschillende variabelen die in de andere objecten los staan samengevoegd. Bijvoorbeeld FactuurAdres is één variabele in OrderOverview terwijl het in Klant uit tien variabelen bestaat.

5.3 Klassediagram

In het klassediagram (figuur 11 en figuur 12) zijn naast de domeinklasse ook de overige klassen afgebeeld. Bijvoorbeeld de GUI klassen met de bijbehorende controller klassen. Verder is hierin te zien welke interface klassen via RPC met elkaar communiceren.



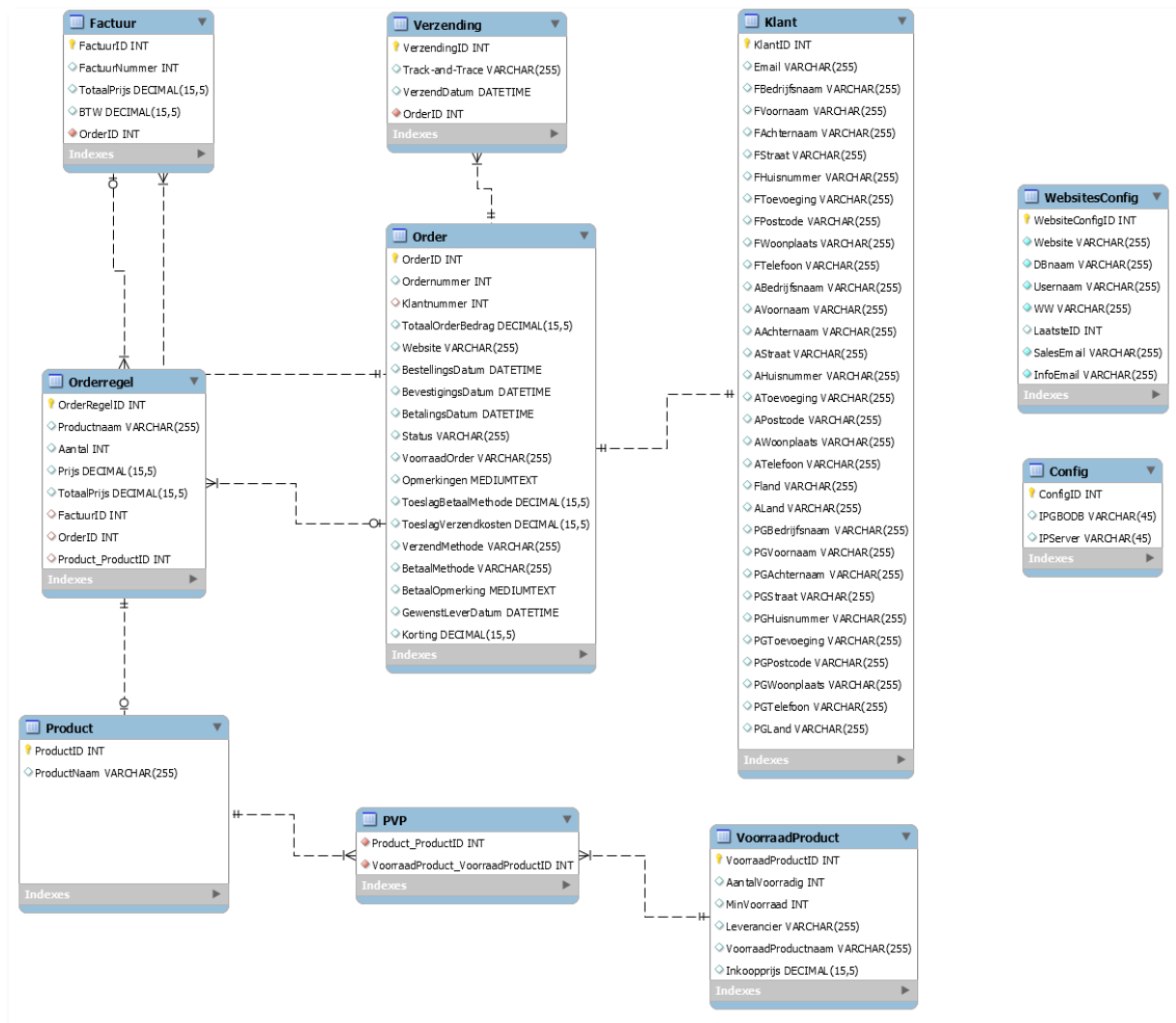
Figuur 11 Klassediagram GBOServerApp



Figuur 12 Klassediagram GBOClientApp

5.3 ERD

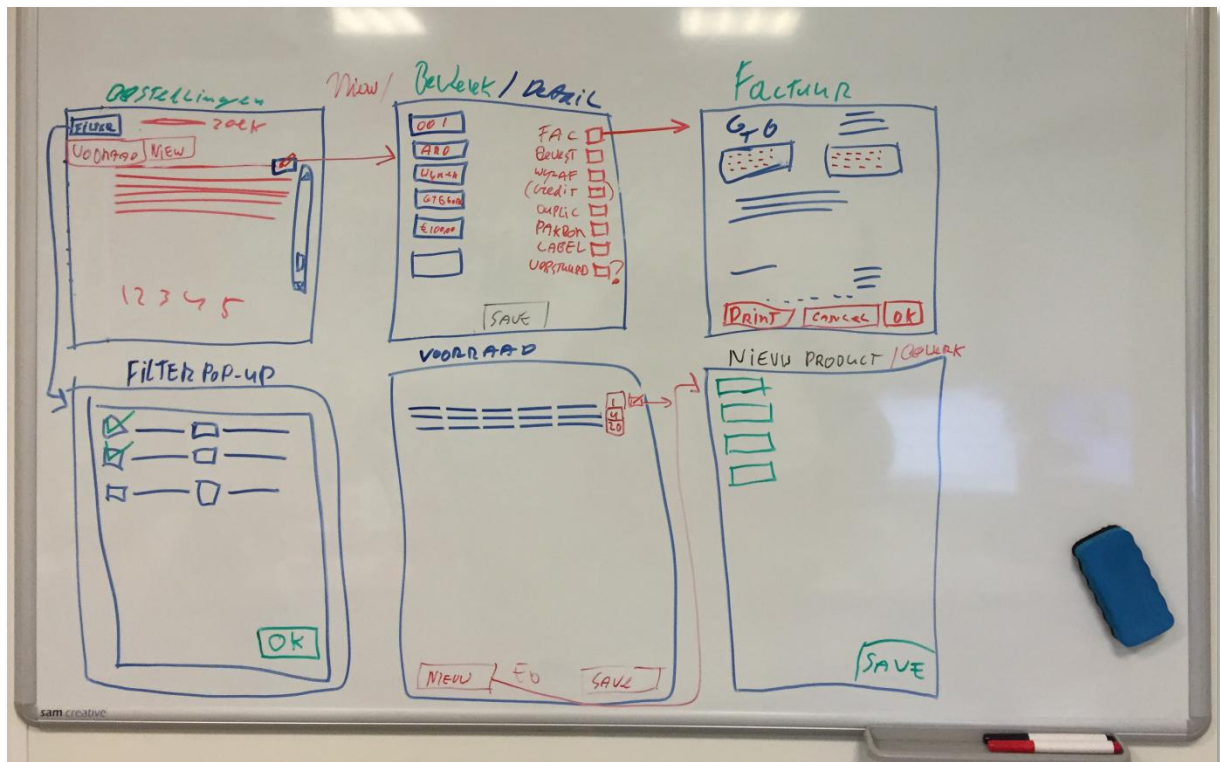
In het database ontwerp(figuur 13) is weergegeven hoe de verschillende tabellen zich tot elkaar verhouden met hun onderlinge relaties. Er zijn in totaal tien tabellen in de database.



Figuur 13 ERD

5.4 Wireframes

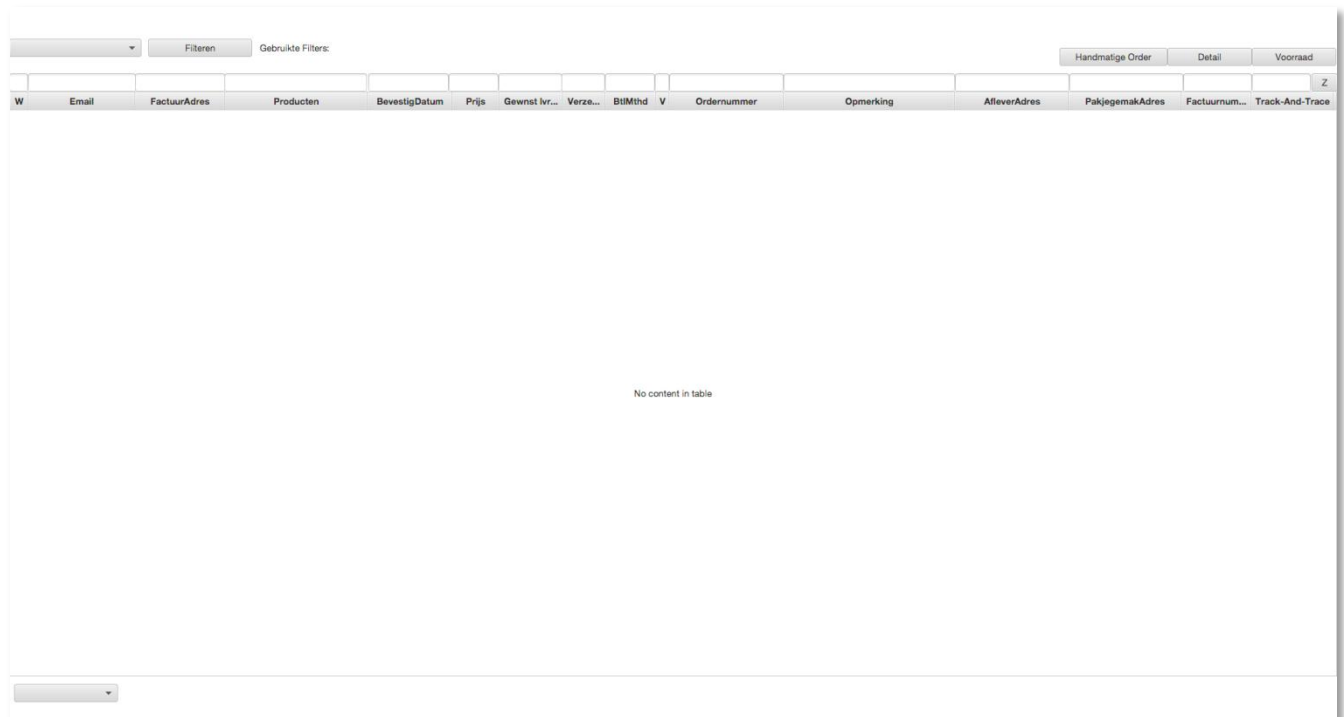
Voordat er wireframes zijn gemaakt zijn er eerst samen met de opdrachtgever enkele schetsen gemaakt op een whiteboard (figuur 14). Hierdoor was het makkelijk om dingen te herontwerpen en aan te passen. Nadat de schetsen akkoord waren is er begonnen aan de wireframes zodat er een duidelijker beeld ontstond van de werkwijze binnen de applicatie.



Figuur 14 Schets

Tijdens de ontwerpfase zijn er verschillende wireframe ontwerpen gemaakt. De ontwerpen zijn direct gemaakt met Scene Builder een draw-and-drop applicatie voor het maken van gebruikersinterfaces voor JavaFX. Hiervoor is gekozen omdat, door het direct in JavaFX te maken en niet eerst met een wireframe tool dit dubbel werk voorkomt.

Wireframe 1 is het OrderOverview scherm (figuur 15). Hierin staan van maximaal 100 orders een samenvatting met de belangrijkste informatie. In elke kolom van de tabel is apart te zoeken.



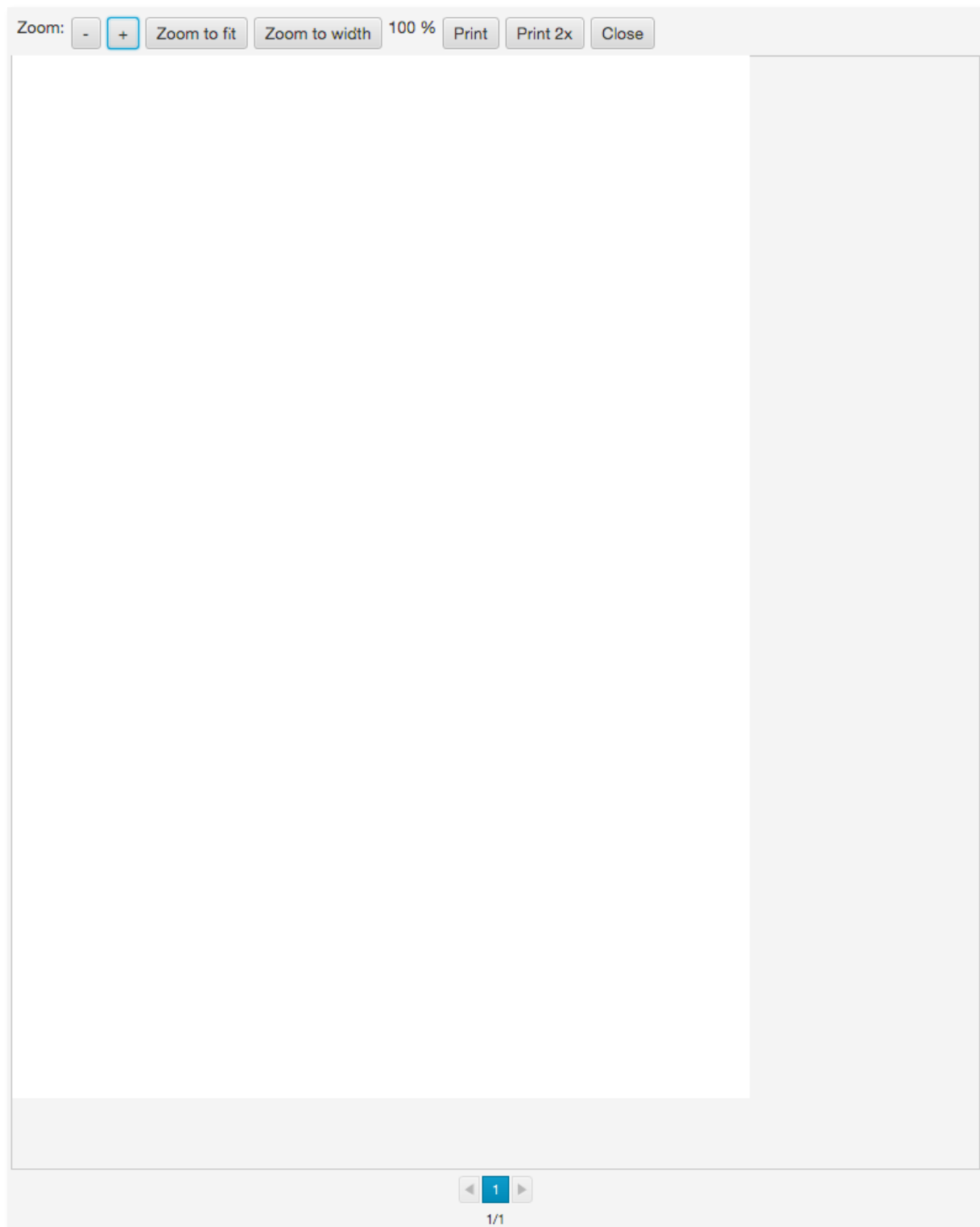
Figuur 15 Wireframe 1 OrderOverview

Er zijn verschillende sorteer- en filteropties om verschillende lijsten met OrderOverviews te krijgen. Door op de button “Detail” te klikken wordt Wireframe 2 geopend van de geselecteerde order. Bij deze actie hoort ook Sequence diagram 2.

Wireframe 2 is het OrderDetail scherm (figuur 16). Hierin zijn de details van de order te zien. Ook zijn hier een aantal acties uit te voeren zoals het bewerken van de order gegevens. Zo zijn er facturen, pakbonnen en verzendlabels te maken. Bij het maken van deze documenten is er steeds de keuze tussen het maken en bekijken van het document en het maken en direct printen van het document zonder dit te laten zien. Door te klikken op de button “Maak / Bekijk Factuur” wordt wireframe 3 geopend. Bij deze actie hoort Sequence diagram 3.

Figuur 16 Wireframe 2 OrderDetail

Wireframe 3 is het factuurscherm (figuur 17). Hierin is de gemaakte factuur te bekijken en te printen. Doordat bij GoodTimeGifts de facturen vrijwel altijd twee keer geprint worden is hier een aparte knop voor aangemaakt. Tijdens de implementatie fase is het ontwerp van dit scherm veranderd. Dit komt doordat de gebruikte library een sample bevatten waarin verschillende zoom knoppen zitten. Deze zijn overgenomen in de applicatie.



Figuur 17 Wireframe 3 Factuur

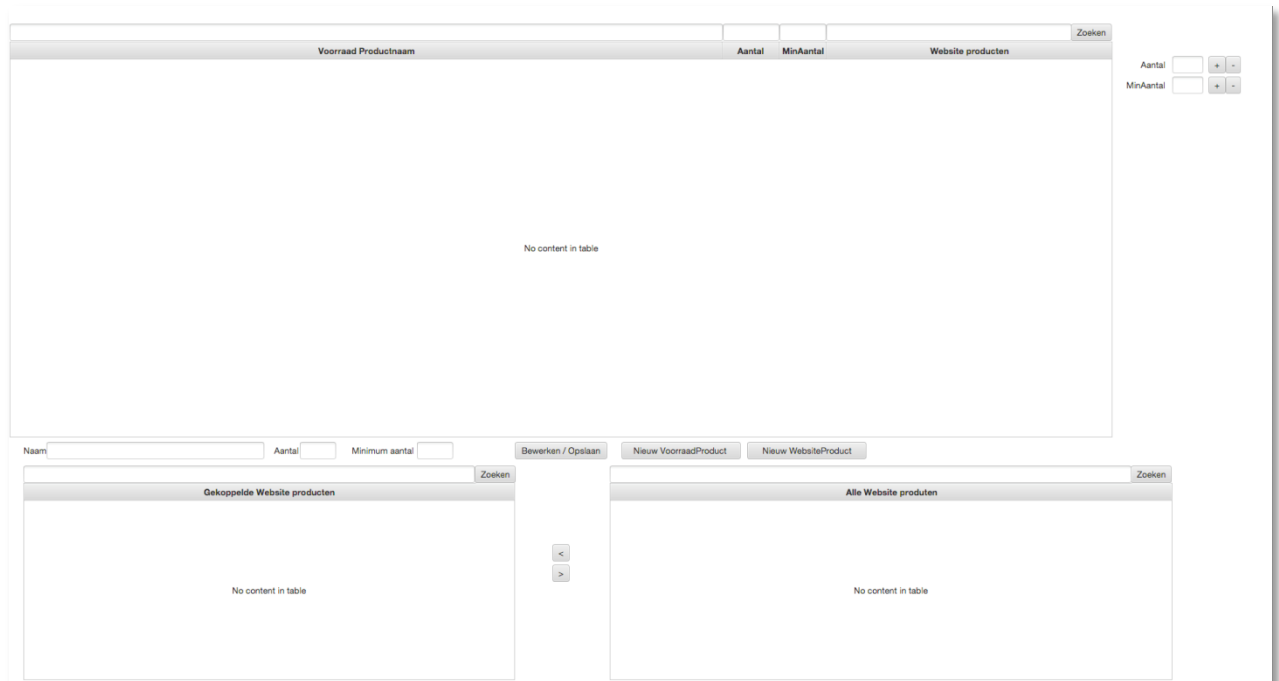
Wireframe 4 is het Filterscherm (figuur 18). Dit scherm kan opgeroepen worden door op de button “Filteren” te klikken in het OrderOverview scherm. In dit scherm staan verschillende opties voor andere overzichten te krijgen van OrderOverviews. De ingestelde opties worden ook zichtbaar in het OrderOverview scherm.

The wireframe shows a filter interface with the following elements:

- Left Column (Dropdowns):**
 - Website 1, Website 2, Website 3
 - Leverancier 1, Leverancier 2, Leverancier 3
 - Betaalmethode 1, Betaalmethode 2, Betaalmethode 3
 - Verzendmethode 1, Verzendmethode 2
 - Status 1, Status 2
- Right Column (Form Elements):**
 - Van: Text input with a clear button (X)
 - Tot: Text input with a clear button (X)
 - Soort Datum: Dropdown menu
 - Prijs van: Range slider
 - Prijs tot: Range slider
 - Soort Prijs: Dropdown menu
- Bottom Right:** OK button

Figuur 18 Wireframe 4 Filter

Wireframe 5 is het Voorraadscherm (Figuur 19). Dit scherm is op te roepen door in het OrderOverview scherm op de button “Voorraad” te klikken. In dit scherm is het mogelijk om nieuwe producten toe te voegen, en de voorraad van de producten te bewerken. Er is gekozen om een scheiding te maken tussen voorraadproducten en websiteproducten. Dit is gedaan omdat één websiteproduct uit meerdere voorraadproducten kan bestaan. Bijvoorbeeld: websiteproduct is: Set van 3 steden klokken Londen, New York en Paris. Dit product bestaat uit drie voorraad producten: Steden klok Londen, Steden klok New York en Steden klok Paris. Bij bestelling van het website product dient de voorraad van alle drie de voorraad producten afgeschreven te worden. Het is tevens ook mogelijk om één voorraadproduct te hebben met meerdere websiteproducten. Een product op een GoodTimeGifts webshop heeft een andere naam dan op bijvoorbeeld Bol.com. Ze zijn beide wel hetzelfde voorraadproduct. De koppeling tussen deze twee producten is te zien in de Gekoppelde Website producten tabel.



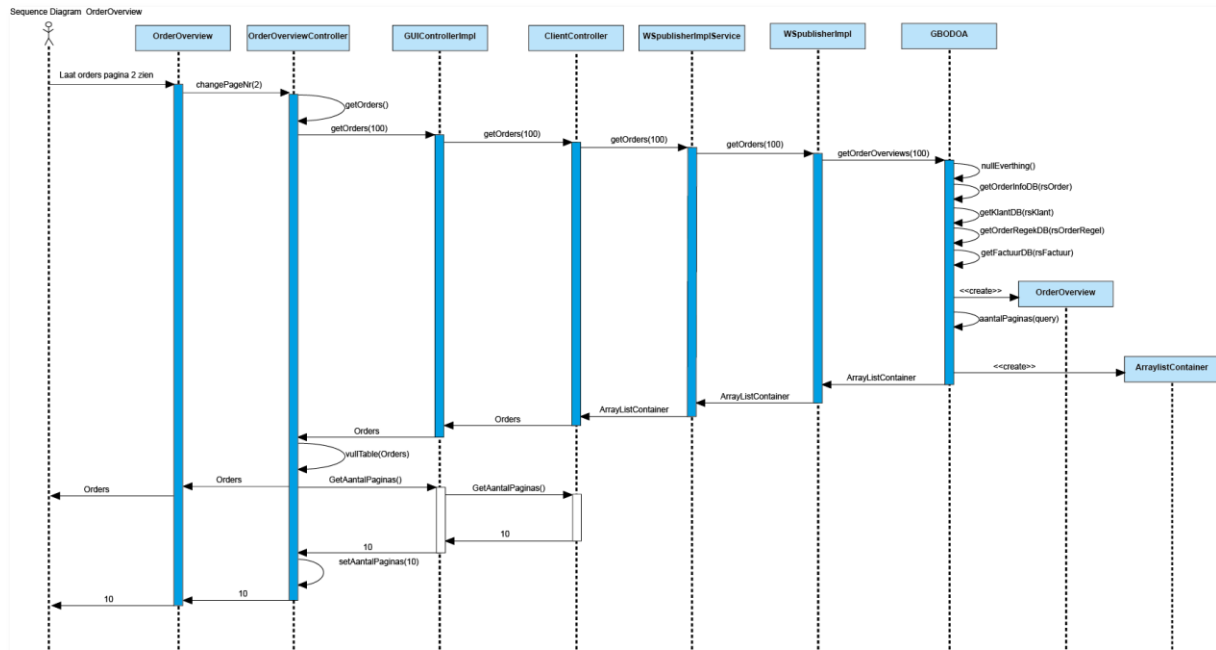
Figuur 19 Wireframe 5 Voorraad

De wireframes zijn via een acceptatietest getest, door de opdrachtgever en een eindgebruiker. De test is succesvol verlopen en wireframes zijn door beide testers geaccepteerd. Een uitgebreide testrapportage is te vinden in Bijlage III.

5.5 Sequence Diagrammen

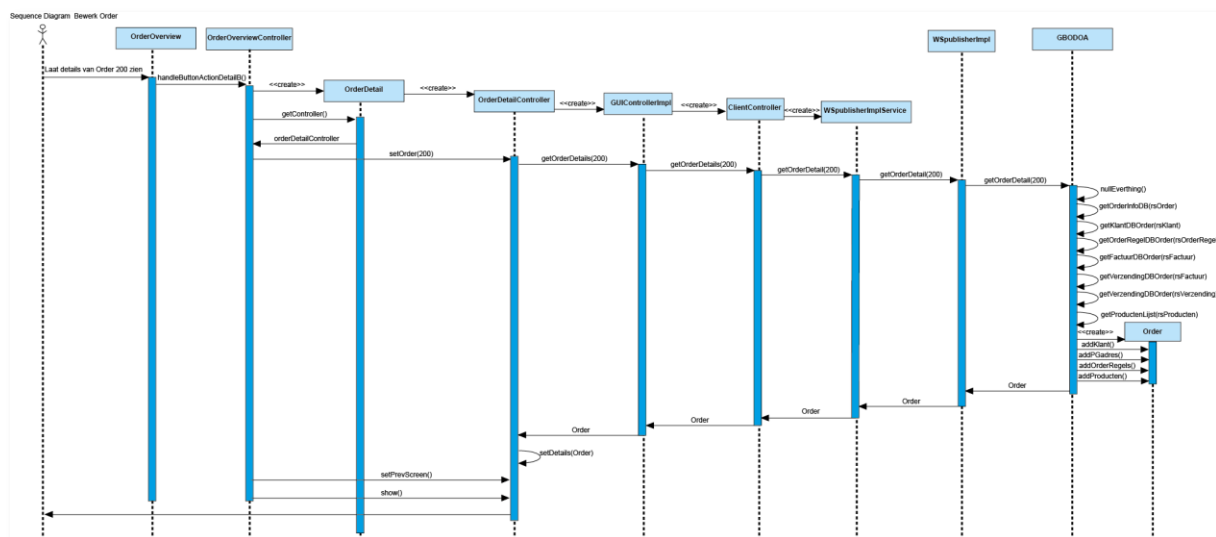
In de Sequence diagrammen is te zien welke klassen, objecten en methodes er worden gebruikt om acties uit te voeren.

In Sequence diagram 1 (figuur 20) wordt er door een gebruiker de tabel met OrderOverviews opgevraagd. Hierin is te zien dat de GBOClientApp 100 OrderOverview object opvraagt aan de GBOServerApp. De GBOServerApp stuurt dan een ArrayListContainer object terug met 100 OrderOverviews en hoeveel pagina's van 100 OrderOverview objecten er beschikbaar zijn.



Figuur 20 Sequence Diagram 1

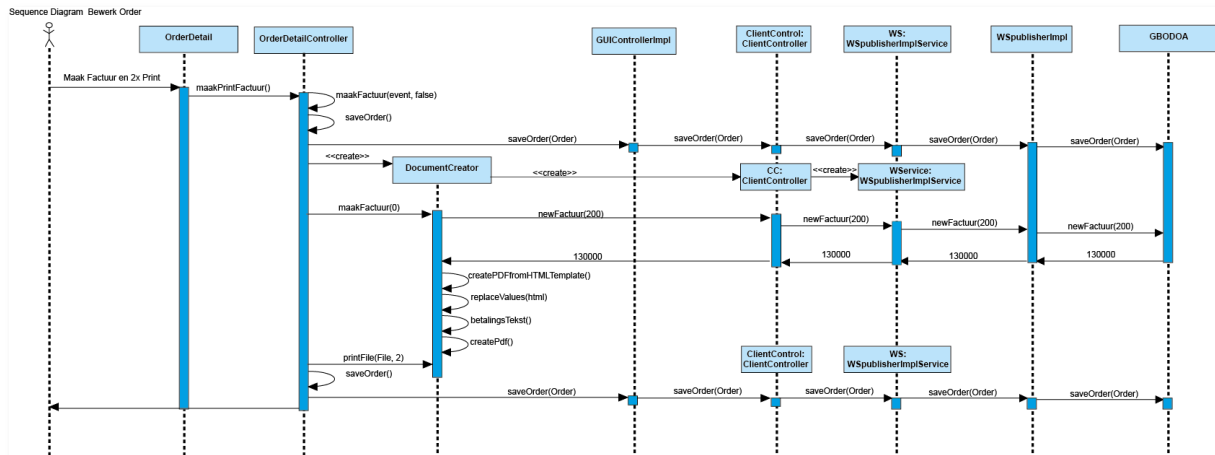
In Sequence diagram 2 (figuur 21) wordt er door de gebruiker de detailgegevens van een order met OrderID '200' opgevraagd. Hierin is te zien dat de GBOClientApp één Orderobject opvraagt aan de GBOServerApp met OrderID 200. De GBOServerApp stuurt dan het order object terug nadat eerst alle gegevens uit de database gehaald zijn. Ook worden alle gegevens van het vorige scherm opgeslagen.



Figuur 21 Sequence Diagram 2

In Sequence diagram 3 (figuur 22) wordt er door de gebruiker een factuur gemaakt en geprint. Hierin is te zien dat eerst de order wordt opgeslagen. Daarna wordt er het eerstvolgende factuurnummer uit de database opgehaald. De DocumentCreator maakt dan

een nieuwe factuur aan en print deze twee keer uit. Hierna wordt de order weer opgeslagen in de database.



Figuur 22 Sequence Diagram 3

Nadat het ontwerp voltooid was is deze door de begeleiders en de opdrachtgever goedgekeurd en is er begonnen aan de volgende fase.

6 Implementatie

Tijdens de implementatiefase is er gewerkt met een variatie op de Scrum ontwikkelmethode (Schwaber & Sutherland, 2013). Tijdens het begin van de sprint zijn door de scrummaster en de productowner de storypoints bepaald over functionaliteit die ontwikkeld wordt in die sprint. Hierin worden ook de requirements die nog niet volledig gerealiseerd zijn in de vorige sprint meegenomen. Dagelijks is er een daily-stand-up meeting, tijdens deze korte bespreking worden de reeds gemaakte de functionaliteiten en beslissingen besproken. Aan het einde van elke sprint wordt er door de ontwikkelaar een demonstratie gegeven voor de opdrachtgever en een eindgebruiker. Hierna wordt deze functionaliteit besproken en gekeken wat er geaccepteerd wordt en wat er nog aangepast dient te worden. Aan het einde van de sprint wordt de sprint en de gemaakte functionaliteit geëvalueerd. Voor het bijhouden van de backlog is er gebruik gemaakt van Trello.

De ontwikkelaar plaats de requirements die hij aan het ontwikkelen is in de kolom "In ontwikkeling". Als de requirement in de ogen van de ontwikkelaar klaar is, wordt deze naar de kolom "Klaar" verschoven. Tijdens het einde van elke sprint of bij de daily-stand-up meeting wordt de huidige functionaliteit getoond in een demonstratie. De requirements die in kolom "Klaar" staan worden dan of naar "Geaccepteerd" of "In ontwikkeling" gezet, afhankelijk of ze goedgekeurd zijn. Dit is te zien in figuur 23.

PW	Peter Wijnen moved <u>Een factuur moet geprint kunnen worden. Must - 5</u> from Klaar to Geaccepteerd Jun 30 at 9:29 AM
PW	Peter Wijnen moved <u>Van een bevestigde order moet een factuur gemaakt kunnen worden. Must - 5</u> from Klaar to Geaccepteerd Jun 30 at 9:28 AM
PW	Peter Wijnen moved <u>De applicatie moet beschikken over een bekijk factuur scherm. Must - 20</u> from Klaar to Geaccepteerd Jun 30 at 9:28 AM
AW	Ard Wijnen moved <u>Van een bevestigde order moet een factuur gemaakt kunnen worden. Must - 5</u> from In ontwikkeling to Klaar Jun 30 at 9:27 AM
AW	Ard Wijnen moved <u>Een factuur moet geprint kunnen worden. Must - 5</u> from In ontwikkeling to Klaar Jun 30 at 9:27 AM
AW	Ard Wijnen moved <u>De applicatie moet beschikken over een bekijk factuur scherm. Must - 20</u> from In ontwikkeling to Klaar Jun 30 at 9:27 AM
PW	Peter Wijnen moved <u>Een factuur moet geprint kunnen worden. Must - 5</u> from Klaar to In ontwikkeling Jun 29 at 11:36 AM
PW	Peter Wijnen moved <u>De applicatie moet beschikken over een bekijk factuur scherm. Must - 20</u> from Klaar to In ontwikkeling Jun 29 at 11:36 AM
PW	Peter Wijnen moved <u>Van een bevestigde order moet een factuur gemaakt kunnen worden. Must - 5</u> from Klaar to In ontwikkeling Jun 29 at 11:36 AM
PW	Peter Wijnen moved <u>Een factuur moet opgeslagen worden in PDF. Must - 40</u> from Klaar to Geaccepteerd Jun 29 at 11:35 AM

Figuur 23 Trello log

In het totaal zijn er in vier sprints aan de applicatie gewerkt. Van de 32 must requirements zijn er 21 geïmplementeerd en gemarkeerd als geaccepteerd. Dit is minder dan vooraf gedacht werd. De reden hiervoor was dat er in de planning niet goed rekening was gehouden met de vakanties. De kans dat niet alles geïmplementeerd kon worden was ook meegenomen in de risicoanalyse in de PID Bijlage I hoofdstuk 4.5. De oplossing die hier beschreven staat is toegepast. Door het gebruik van SCRUM was het mogelijk om de overgebleven functionaliteit door te schuiven naar een latere implementatiefase. De opdrachtgever en bedrijfsbegeleider gingen hiermee akkoord. Met de docentbegeleider is overlegd of het invloed had op het afstudeertraject, dit bleek niet zo te zijn.

6.1 Sprint 1

In sprint 1 is de volgende hoofdfunctionaliteit geïmplementeerd:

Webservice tussen GBOServerApp en Webshops.

Via de webshop kan aan de GBOServerApp doorgegeven worden dat er een nieuwe order geplaatst is. Er worden via deze RPC geen gegevens doorgegeven. Zie figuur 24. De GBOServerApp haalt dan vervolgens alle nieuwe orders op uit de database van de webshops. Zo worden ook orders die gemist zijn door bijvoorbeeld een connectie probleem meegenomen en niet alleen de geplaatste order die de RPC deed. GBOServerApp start direct na de RPC een nieuwe thread zodat de website niet blijft wachten totdat GBOServerApp klaar is. Zie figuur 25.

```
public function sendNewOrderMessage($dataArray)
{
    $client = new SoapClient("http://[redacted]:9999/ws/wsimportorder?wsdl");
    $client->ImportOrder();
}
```

Figuur 24 Webservice website → GBOServerApp

```
/**
 * Import orders van webshop.
 */
@Override
public void ImportOrder() {
    Thread one = new Thread() { // Hierdoor blijft de website niet wachten tot GBO klaar is
        public void run() {
            try {
                GBODAO GBODAO = new GBODAO();

                GBODAO.newOrder();
            } catch (SQLException ex) {
                System.out.println("Fout in SQL code" + ex);
            } catch (ClassNotFoundException ex) {
                System.out.println(ex);
            }
        }
    };

    one.start();
}
```

Figuur 25 Webservice website → GBOServerApp

Webservice tussen GBOServerApp en GBOClientApp.

GBOClientApp kan via de webservice op verschillende manieren een lijst van OrderOverview objecten opvragen afhankelijk van de ingevulde zoekwoorden en de geselecteerde status. Zie figuur 26 en figuur 27.


```

/**
 *
 * @param arg0
 * @return returns gboserverapp.ws.ArrayListContainer
 */
@WebMethod
@WebResult(partName = "return")
@Action(input = "http://ws.gboserverapp/WSPublisher/getOrdersRequest", output = "http://ws.gboserverapp/WSPublisher/getOrdersResponse")
public ArrayListContainer getOrders(
    @WebParam(name = "arg0", partName = "arg0") int arg0);

```

Figuur 26 Webservice GBOClientApp → GBOServerApp

```

/**
 * Wordt aangeroepen vanuit clientapp. returns ArrayListContainer
 * @param beginnr
 * @return
 */
@Override
public ArrayListContainer getOrders(int beginnr) {
    ArrayListContainer ALC = null;
    try {
        ALC = this.GBODAO.getOrderOverviews(beginnr);
    } catch (SQLException ex) {
        Logger.getLogger(WSPublisherImpl.class.getName()).log(Level.SEVERE, null, ex);
    }
    return ALC;
}

```

Figuur 27 Webservice GBOClientApp → GBOServerApp

Ophalen gegevens uit de webshops database en invoeren in GBO database

Er is eerst een mapping gemaakt tussen de GBO database en de develop-webshop databases zodat duidelijk was in welke tabellen de gegevens stonden. Enkele gegevens moeten omgeschreven worden naar het gewenste format. Zo wordt de straat opgesplitst in straat, huisnummer en toevoeging. Zie figuur 28.

```

while (orderAddress.next()) {
    adresType = orderAddress.getString("address_type");

    if (adresType.equals("billing")) {
        FBedrijfsnaam = orderAddress.getString("company");
        FVoornaam = orderAddress.getString("firstname");
        FAchternaam = orderAddress.getString("lastname");
        FStaat = orderAddress.getString("street");
        String arr[] = FStaat.split("\\r?\\n|\\r"); // opsplitsing tussen Straat huisnummer en toevoeging

        if (arr.length == 2) { // Straat & huisnummer
            FHuisnummer = arr[arr.length - 1];
            FStaat = arr[arr.length - 2];
        }

        if (arr.length == 3) { // Straat & huisnummer & toevoeging
            FToevoeging = arr[arr.length - 1];
            FHuisnummer = arr[arr.length - 2];
            FStaat = arr[arr.length - 3];
        }

        FPostcode = orderAddress.getString("postcode");
        FWoonPlaats = orderAddress.getString("city");
        FTelefoon = orderAddress.getString("telephone");
        FLand = orderAddress.getString("country_id");
    }
}

```

Figuur 28 Opsplitsing straat

OrderOverview scherm.

De gebruiker kan verschillende overzichten krijgen van de orders, zoals:

- Alle orders
- Nieuwe orders
- Open orders
- Afgeronde orders
- Heropende orders
- Bevestigde orders

Ook is het mogelijk om per kolom te zoeken. Zie figuur 29.

	ard@goodtimegifts	oceanie						
W	Email	FactuurAdres	Producten	BevestigDatum	Prijs	Gewnst lvr...	Verze...	BtlMthd
1	ard@goodtimegifts.nl	Ard Wijnen Oceanie 8 a 6014db ITTERVO... NL	TTD Elegance Rounded Stripe 50		69.95	2014-09-06 00:00:00	Overdag	Vooruit v..
1	ard@goodtimegifts.nl	Ard Wijnen Oceanie 8 a 6014db ITTERVO... NL	TTD Elegance Rounded Stripe 50		69.95	2014-09-06 00:00:00	Overdag	Vooruit v..
1	ard@goodtimegifts.nl	Ard Wijnen Oceanie 8 a 6014db ITTERVO... NL	Saxon Wandklok Radio Controle		39.95			Vooruit v..

Figuur 29 Zoeken per Kolom

6.2 Sprint 2

In sprint 2 is de volgende hoofdfunctionaliteit geïmplementeerd:

Order in detail bekijken en bewerken.

In het Order detail scherm is het mogelijk om een order volledig te bekijken en te bewerken.

Zo is het mogelijk om de adressen aan te passen, orderregels toe te voegen en de korting te veranderen. Bij het aanpassen van prijzen en producten wordt er een nieuwe totaalprijs berekend, de oude prijs blijft ook zichtbaar. Zie figuur 30.

Om datums in te voeren zijn er speciale datepickers gebruikt. Dit is makkelijker met invoeren en zo is er altijd hetzelfde format. Zie figuur 31.

Product	Prijs	Aantal	Totaalprijs
GTG Hands Zilver 275194	25.95	1	25.95
product 1	15.00	1	15.00

Ordernummer	100040614	Facturen No content in table	Verzendkosten:	0.00
BestelDatum	24-08-2015		Korting:	0.00
BevestigDatum	dd-mm-yyyy		Betaalmethode kosten:	0.00
BetalingsDatum	dd-mm-yyyy		BTW:	7.11
Website	1		EX BTW:	33.84
BetaalMethode	Achteraf Bedrijven		Totaal:	40.95
FactuurDatum			Totaal besteld:	25.95

Figuur 30 Totaalprijs

Ordernummer	100040614
BestelDatum	24-08-2015
BevestigDatum	
BetalingsDatum	
Website	1
BetaalMethode	Achteraf
FactuurDatum	
Gewenste VerzendDatum	
Verzendtype	Overdag
Voorradig	TODO
<input checked="" type="checkbox"/> Review	

< August >		< 2015 >	
Sun	Mon	Tue	Wed
31	26	27	28
32	2	3	4
33	9	10	11
34	16	17	18
35	23	24	25
36	30	31	1

Figuur 31 Datepickers

Opslaan bewerkte order.

Door op de button "Save" te drukken wordt de gehele order opgeslagen. Er is niet gekozen voor het alleen opslaan van de onderdelen die gewijzigd zijn. Hier is voor gekozen omdat de tijds winst van minder invoegen in de database vrijwel nihil is. Mocht dit in de toekomst wel een probleem worden is dit dan nog altijd aan te passen.

6.3 Sprint 3

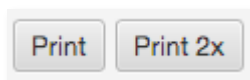
In sprint 3 is de volgende hoofdfunctionaliteit geïmplementeerd:

Maken factuur vanuit template

In overleg met de opdrachtgever is ervoor gekozen om een template van de factuur te maken in html. Hiermee is de lay-out van de factuur makkelijk aan te passen. In het template worden de correcte gegevens ingevoerd en opgeslagen als pdf.

Bekijken factuur

Als er gekozen is voor het bekijken van de factuur komt er een klein scherm waar de pdf in is afgebeeld. De gebruiker kan dan kiezen tussen 1x printen en 2x printen. Dit is erin gebouwd omdat bij GoodTimeGifts in de meeste gevallen de factuur 2x geprint moet worden. Zie figuur 32.



Figuur 32 Printen factuur

Printen factuur

De gebruiker kan er ook voor kiezen om de factuur niet te bekijken maar direct 2x uit te printen. Zie figuur 33.



Figuur 33 Bekijken of printen factuur

6.4 Sprint 4

In sprint 4 is aan dezelfde hoofdfunctionaliteit gewerkt als in sprint 3 de functionaliteit is in deze sprint voltooid.

6.5 Tests

Unittests

Tijdens de implementatiefase zijn gelijktijdig met de ontwikkeling van de functionaliteit unittests gemaakt. In totaal zijn er 24 unittests gemaakt. De unittests zijn geschreven voor stukken code waar tussen formats gewisseld wordt en waar gerekend wordt met prijzen. Zie figuur 34 en figuur 35. Een volledige testrapportage is terug te vinden in bijlage III.

```
@Test
public void OrderDetailController_calPrijsen1() {

    // Te verandere Waardes
    String totaalprijs = "100.00";
    String korting = "0.00";
    String betalingstoeslag = "0.00";
    String verzendmethodetoeslag = "0.00";

    String expectedTotaal = "100.00";
    String expectedBTW = "17.36";
    String expectedEXBTW = "82.64";
    // Eind verandere Waardes

    List<OrderRegel> or = new ArrayList();
    Klant k = new Klant("a", 1, "a", "a", "a", "a", "a", "a", "a", "a", "a", "a", "a", "a", "a", "a");
    OrderRegel orderregel = new OrderRegel("1", new BigDecimal(totaalprijs), "productnaam", new BigD
    or.add(orderregel);
    o = new Order("2015-08-19 07:48:34", "Afhalen", "null", "null", "null", null, "null", k, new BigD
    ObservableList<OrderRegel> Oregels = FXCollections.observableArrayList();
    Oregels.add(orderregel);
    ODC.setOregels(Oregels);

    ODC.calPrijsen(o);

    ODC.getBTWValue();
    ODC.getTotaalprijs();

    if (ODC.getTotaalprijs().setScale(2, RoundingMode.HALF_UP).toString().equals(expectedTotaal)) {
        assertTrue(true);
    } else {
        assertTrue(false);
    }

    if (ODC.getBTWValue().setScale(2, RoundingMode.HALF_UP).toString().equals(expectedBTW)) {
        assertTrue(true);
    } else {
        assertTrue(false);
    }
}
```

Figuur 34 Unittest prijzen



Figuur 35 Unittests

Acceptatie test

Nadat alle onderdelen geïmplementeerd zijn is er een acceptatietest uitgevoerd. Deze test is uitgevoerd om te kijken of alle onderdelen van de applicatie aan de eisen en wensen van de opdrachtgever en eindgebruiker voldoen. De test is geslaagd als de applicatie door beide testers geaccepteerd wordt. De ontwikkelaar heeft Tester 2 en Tester 3 de gehele applicatie laten zien, doormiddel van een demonstratie. Na de demonstratie konden de beide testers zelf de applicatie uitproberen, dit kon gelijktijdig zodat ook het multi user gedeelte getest werd. Er zijn tijdens de test enkele nieuwe requirements en aanpassingen naar voren gekomen die meegenomen zullen worden in de ontwikkeling. De applicatie is geaccepteerd en de test gemarkeerd als passed. Een volledige testrapportage is terug te vinden in bijlage III.

6.6 Obstakels

Tijdens het implementeren van GoodBackOffice is er tegen twee obstakels aangelopen.

Het eerste obstakel was bij het testen van de webservice. Nadat de GBOServerApp gedeployed was op de server en GBOClientApp op de terminalservers werkte de webservice in eerste instantie goed. Alleen als er ongeveer 1 minuut geen aanvragen gedaan werden viel de connectie weg en konden er geen aanvragen meer verstuurd worden. Alleen door de WSDL file te bekijken in een browser ontstond de connectie weer. Dit is getest door de

GBOServerApp elke 20 seconden de URL van de WSDL file te pingen. Wanneer dit gedaan werd bleef de connectie werken. In de Netbeans ontwikkel omgeving was dit probleem niet voorgekomen. Na verder onderzoek bleek dat de firewall op de server het probleem was. Hij sloot de webservice connectie met de terminalserver als er een tijdje geen aanvragen gedaan werden. Door de regels van de firewall aan te passen was het probleem opgelost.

Het tweede obstakel was het zoeken van een correcte plug-in voor het maken van een pdf. Na het bekijken van verschillende libraries leken PDFBox (The Apache Software Foundation, 2015) en ITEXT (iText Group NV, iText Software Corp., iText Software BVBA , 2015) beide geschikte kandidaten. Het idee was om een template te maken in pdf waarna de verschillende variabelen vervangen werden door de gegevens die op de factuur vermeld moeten worden. Het probleem hierbij was dat pdf's niet geschikt zijn voor bewerken, zo is het niet mogelijk om woorden met hoofdletters in te voegen. Er moest gezocht worden naar een ander format voor het template. Na overleg met de opdrachtgever bleek html een geschikte kandidaat te zijn, doordat er meerdere personen binnen GoodTimeGifts zijn die hiermee templates kunnen maken. De nieuwe library moest dus html om kunnen zetten in een pdf bestand. YaHP Converter (ALLCOLOR.ORG, 2015) bleek dit uitstekend te doen, voor het pdf gedeelte maakt hij gebruik van de ITEXT library. Deze library is gebruikt in GoodBackOffice.

7 Conclusie en aanbevelingen

De met de opdrachtgever afgesproken functionaliteit van GoodBackOffice voor deze afstudeeropdracht is afgerond en door de opdrachtgever geaccepteerd. De hoofdfunctionaliteit: Importeren van orders, bewerken van orders en maken van factuur zijn allemaal geïmplementeerd. De implementatiefase is verdeeld over vier sprints. GoodBackOffice is nog niet geschikt om in gebruik te nemen er zal eerst verder gebouwd moeten worden zodat de applicatie meer functionaliteit bevat.

Tijdens het onderzoek is gebleken wat de eisen waren voor het nieuwe systeem. JavaFX bleek na onderzoek de meest geschikte keuze als Gebruikersinterface technologie. SOAP webservices was de meest geschikte keuze als Remote communicatie technologie. Ook bleek dat er geen bestaande applicaties waren die voldeden aan de eisen van GoodTimeGifts. Wanneer GoodBackOffice verder wordt ontwikkeld wordt er aanbevolen dat er dan een nieuwe onderzoeksfase plaatsvindt, waarin het onderzoek wordt aangevuld met bijvoorbeeld een nieuw onderzoek naar de eisen voor functionaliteit die buiten de scope is gevallen en welke software het meest geschikt is voor het aanmaken van verzendlabels.

Het ontwerp is zo gemaakt dat er in de toekomst snel en gemakkelijk verder gebouwd kan worden aan de applicatie. Tijdens de ontwerpfase is ook rekening gehouden met Multi user eis. In het ontwerp zijn ook de aanbevelingen en conclusie van het onderzoek gebruikt. Het wordt dan ook aanbevolen dat bij de volgende implementatiefase het ontwerp verder wordt aangevuld.

De gemaakte planning in het begin van het project (PID Bijlage I) is volledig haalbaar geweest. Dit bleek na overleg geen probleem te zijn, de functionaliteit waarvan het implementeren niet gelukt is zal in de volgende ontwikkelingsfase meegenomen worden. Deze mogelijkheid was ook meegenomen in de risicoanalyse in hoofdstuk 4.5 Risicomanagement van de Bijlage I.

Aanbevolen wordt dat alle drie de uitgevoerde tests bij een nieuwe implementatiefase weer uitgevoerd worden, hierbij hoort ook het aanvullen van de unittests.

Evaluatie

Ik wilde er graag voor zorgen dat de opdracht duidelijk was voor alle betrokken partijen. Daarom heb ik hier veel tijd aan besteed en een aantal gesprekken gevoerd met de opdrachtgever. Het werd mij al snel duidelijk dat het niet ging lukken om alle functionaliteit te voltooien binnen de afstudeerperiode. Daarom vond ik het bepalen van de correcte scope van het project een belangrijk aspect. Achteraf gezien bleek de vooraf bepaalde scope nog iets te groot. Dit werd tijdens de ontwikkelingsperiode duidelijk. Ik vond het jammer dat het niet ging lukken zoals gepland maar gelukkig bleek dit geen probleem te zijn.

Voor het project vond ik het belangrijk om te zorgen dat alle onderdelen waarbij het nodig was een onderzoek uit te voeren van te voren bekend waren. Zodat deze niet in een andere fase onderzocht hoefde te worden. Tijdens het onderzoek heb ik huidige applicatie bekeken en in kaart gebracht en zoals ik al verwachtte ontstond er daardoor voor mij een duidelijk beeld van de wensen en eisen van de nieuwe applicatie. De volgende keer zou ik ervoor kiezen om de requirements interviews in meerdere gesprekken te verdelen, omdat het nu veel was om in een gesprek te bespreken. Ik vond het leerzaam om onderzoek te doen naar de meest geschikte technologie omdat ik op deze manier veel leerde van, en bekend werd met, de technologieën en de verschillen hiertussen.

Tijdens de ontwerpfase ben ik erachter gekomen dat er, door het maken van het ontwerp, duidelijkheid ontstaat over hoe de applicatie gebouwd dient te worden. Ik vond het lastig om tijdens het maken van de Sequence diagrammen aan alle benodigde onderdelen te denken. Hierdoor zijn deze tijdens de implementatiefase nog een paar keer aangepast.

De implementatiefase is in mijn ogen snel en goed verlopen doordat ik tijdens de implementatiefase niet tegen problemen ben aangelopen. Ik vond het handig om de daily-stand-up meeting meer te gebruiken als een demonstratie, zo werd mij ook steeds goed duidelijk wat de opdrachtgever van mijn geïmplementeerd stukken vond.

Ik heb deze afstudeerperiode veel mogen leren en mij mogen ontwikkelen, dit kwam vooral doordat ik vrijwel het hele project alleen heb uitgevoerd met uitzondering van de begeleiding. Het was voor mij de eerste keer dat ik een project voor het grootste gedeelte alleen heb gedaan. Deze ervaring was voor mij zeer leerzaam omdat ik alle aspecten van het project doorlopen heb en niet in teamverband werkte waar iedereen maar een gedeelte van de verantwoordelijkheid draagt.

De aankomende maanden zal ik verder gaan ontwikkelen aan GoodBackOffice. In deze maanden zal ik ervoor zorgen dat GoodBackOffice afgrond wordt en voldoet aan de eisen en wensen van GoodTimeGifts.

Referenties

ALLCOLOR.ORG. (2015.). *YaHP Converter (Yet another Html to Pdf converter)*.
Opgeroepen op 19, 2015 Van: <http://www.allcolor.org/YaHPConverter/>

Englander, R. (2002). *Java and SOAP*. " O'Reilly Media, Inc.". Opgeroepen op 74, 2015

Harold, E. (2004). *Java network programming*. " O'Reilly Media, Inc.". Opgeroepen op 74, 2015

iText Group NV, iText Software Corp., iText Software BVBA (2015). *ITEXT programmable PDF Software*. Opgeroepen op 19, 2015 Van: <http://itextpdf.com/>

JavaFX-Jira.kenai (2015). *JavaFX Bug tracking and project tracking*. Opgeroepen op 144, 2015 Van: [https://jira.kenai.com/issues/?jql=issuetype%20in%20\(%22Enhancement%20request%22%2C%20Epic%2C%20Feature%2C%20Improvement\)](https://jira.kenai.com/issues/?jql=issuetype%20in%20(%22Enhancement%20request%22%2C%20Epic%2C%20Feature%2C%20Improvement))

JavaFX-Jira.kenai (2015). *JavaFX Bug tracking and project tracking*. Opgeroepen op 144, 2015 Van: <https://jira.kenai.com/issues/?jql=issuetype%20%3D%20Bug>

Stafford, J. (2013). *Pros, cons of moving from Swing to JavaFX: UI tools a plus*. Opgeroepen op 144, 2015, van: <http://www.theserverside.com/feature/Pros-cons-of-moving-from-Swing-to-JavaFX-UI-tools-a-plus>

Techannotation (2012). *Rmi vs Web Service*. Opgeroepen op 74, 2015. Van: <https://techannotation.wordpress.com/2012/08/31/rmi-vs-web-service/>

The Apache Software Foundation (2015). *Apache PDFBox - A Java PDF Library*.
Opgeroepen op 19, 2015 Van: <https://pdfbox.apache.org/index.html>

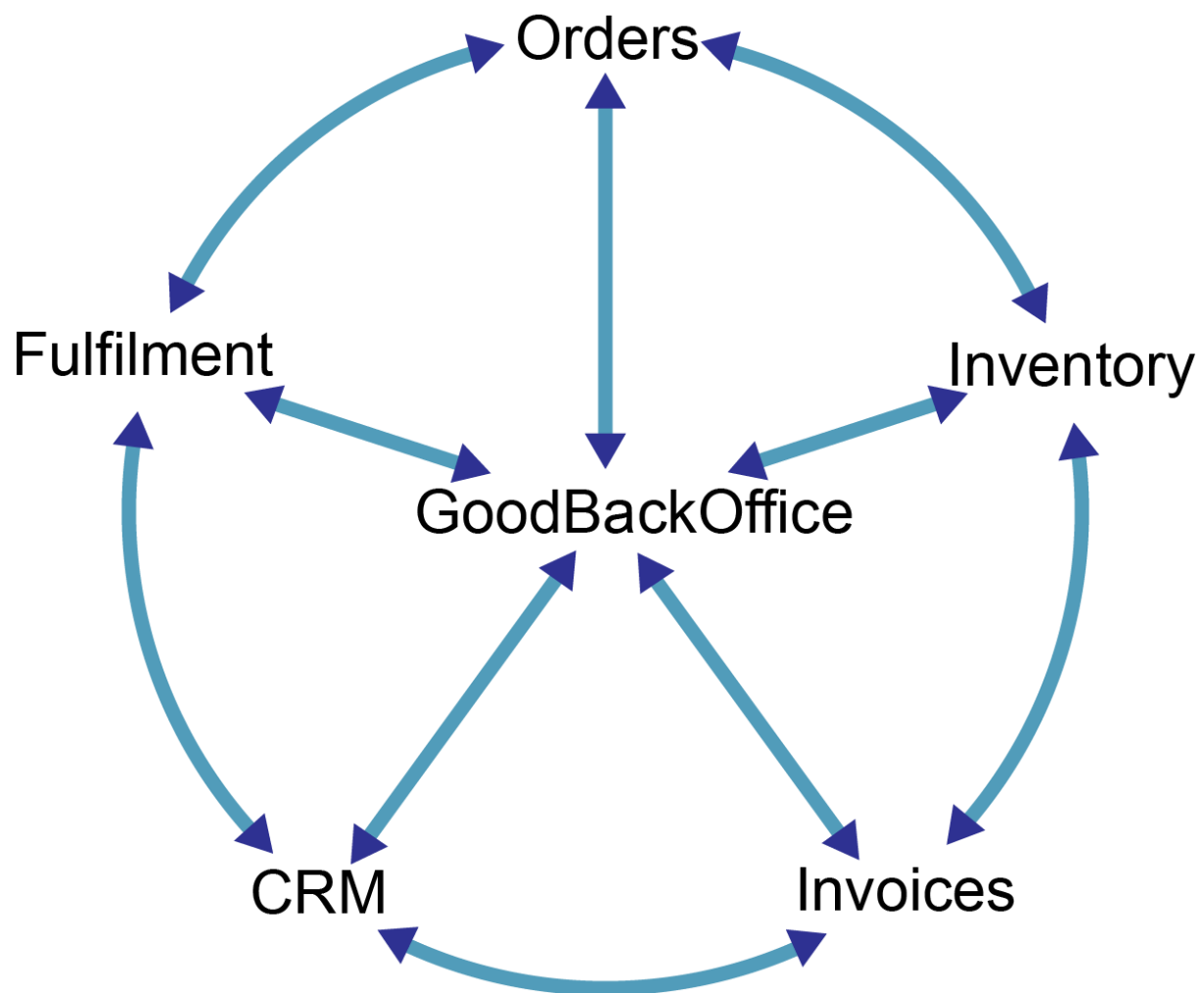
Weaver, J., Gao, W., Chin, S., Iverson, D., & Vos, J. (2014). *Pro JavaFX 8: A Definitive Guide to Building Desktop, Mobile, and Embedded Java Clients*. Apress. Opgeroepen op 74, 2015

Bijlage I

GOODBACKOFFICE

ORDER EN VOORRAAD MANAGEMENTSYSTEEM

Project Initiatie Document



Project
Datum
Auteur
Versie
Status
Bedrijf

GoodBackOffice
31-3-2015
Ard Wijnen
1.2
Goedgekeurd
GoodTimeGifts V.O.F.

Documenthistorie

Versie	Datum	Wijzigingen
0.1	11-2-2015	Geen wijzigingen
1.0	27-2-2015	Feedback verwerkt van Marc Henssen
1.1	11-3-2015	Feedback verwerkt van Marcel Meesters
1.2	18-3-2015	Feedback verwerkt van bedrijfsbezoek

Goedkeuring Versie 1.1

Bedrijfsbegeleider

Naam: Marc Henssen

Datum: 12-3-2015

Paraaf:

Docentbegeleider

Naam: Marcel Meesters

Datum: 12-3-2015

Paraaf:

Inhoudsopgave

1 Achtergrond.....	6
1.1 Aanleiding.....	6
1.2 Context Opdracht.....	6
1.3 Context Onderzoek.....	7
1.4 Huidige situatie	7
2 Projectdefinitie.....	8
2.1 Projectdoelstellingen.....	8
2.2 Project aanpak.....	8
2.2.1 Onderzoek	8
2.2.2 Ontwerp	9
2.2.3 Implementatie.....	9
2.2.4 Oplevering.....	9
2.2.5 Afronding.....	10
2.3 Producten	14
2.4 Scope van het Project.....	11
2.5 Productdecompositiestructuur.....	12
2.6 Productbeschrijving.....	13
2.6.1 Projectdocumentatie.....	13
2.6.2 Ontwerp	13
2.6.3 Applicatie	13
2.6.4 Applicatie documentatie	13
2.7 Projectbudget.....	13
2.8 Projectmanagement.....	13
2.9 Wijzigingsbudget.....	14
2.10 Benodigde Resources.....	14
2.11 Uitsluitingen	14
2.12 Afhankelijkheden.....	14
2.13 Randvoorwaarden.....	14
2.14 Aannames.....	14
3 Projectorganisatiestructuur.....	15
3.1 Opdrachtgever, Productowner	15
3.2 Ontwikkelaar, Afstudeerder, Scrummaster	15
3.3 Eindgebruiker.....	15
3.4 Bedrijfsbegeleider, Applicatie beheerder.....	16
3.5 Docentbegeleider, 1 ^{ste} Assessor	16
3.6 2 ^{de} Assessor	16
4 Projectbeheersing	17

4.1 Rapportage en Voortgangsbewaking	17
4.2 Toleranties	17
4.3 Kwaliteitsbewaking.....	17
4.4 Wijzigingsprocedure.....	17
4.5 Risicomanagement	18
4.6 Planning.....	19
5 Referenties.....	21

Managementsamenvatting

Doel van dit document

Dit document heeft meerdere doelen. Een van deze doelen is alle onderdelen van het project duidelijk in kaart te brengen, en te zorgen voor een duidelijke en realiseerbare planning. Verder zorgt dit document ook voor een duidelijk overzicht van de projectleden en hun taken. Het laatste doel van dit document is dat Fontys Hogescholen (School), Ard Wijnen (Ontwikkelaar), GoodTimeGifts (Opdrachtgever) en Marc Henssen (Bedrijfsbegeleider) op een lijn te zitten betreffende de aanpak en de op te leveren producten.

Aanleiding van het project

GoodTimeGifts is een e-commerce bedrijf met 12 webshops, het order- en voorraad systeem is inmiddels flink verouderd en heeft enkele nadelen. Zo is het niet mogelijk om met meerdere personen het systeem tegelijk te gebruiken. Er missen verschillende stukken functionaliteit zoals het maken van een creditnota. GoodTimeGifts heeft besloten om zelf een nieuw order- en voorraad management systeem te maken genaamd "GoodBackOffice" om deze problemen op te lossen. Het oude systeem is gebouwd in Microsoft Excel en Outlook met verschillende Macro's.

Globale aanpak

GoodTimeGifts heeft ervoor gekozen om een systeem te bouwen in Java. Het onderzoek, ontwerp en ontwikkeling zal gedaan worden door Ard.

Het project is opgedeeld in 5 fases: Onderzoek, ontwerp, implementatie, oplevering en afronding. Het onderzoek wordt uitgevoerd om de volgende hoofdvraag te beantwoorden: *"Wat zijn de eisen voor GoodBackOffice en op welke manier kunnen deze het beste gerealiseerd worden?"* De implementatiefase zal middels een variant op de Scrum ontwikkelmethode (Schwaber & Sutherland, 2013) uitgevoerd worden.

GoodBackOffice moet beschikken over de volgende functionaliteiten: Importeren verkoop orders, maken facturen en beheren voorraad.

Het project is opgedeeld in de volgende hoofdproducten: Projectdocumentatie, applicatie documentatie, ontwerp en applicatie.

Globale kosten en doorlooptijd

Beschikbaar Budget A	Beschikbare resources	Gewenste opleverdata	Beschikbare Doorlooptijd
600uur	0.5fte	31-10-2015	9 maanden

De planning loopt van week 7 tot en met week 44 van 2015.

1 Achtergrond

GoodTimeGifts is een e-commerce bedrijf met 12 webshops, in verschillende productcategorieën zoals klokken en waxjassen. Het bedrijf is opgericht in 2005 en is gevestigd in Ittervoort. GoodTimeGifts verkoopt hun producten via verschillende verkoopkanalen: Eigen webshops, bol.com en Amazon.de.

1.1 Aanleiding

Het order- en voorraad systeem is inmiddels flink verouderd en heeft enkele nadelen. Zo is het niet mogelijk om met meerdere personen het systeem tegelijk te gebruiken. Er missen ook verschillende stukken functionaliteit zoals het maken van een creditnota. GoodTimeGifts heeft besloten om zelf een nieuw order- en voorraad management systeem te maken om deze problemen op te lossen genaamd "GoodBackOffice". Het oude systeem is gebouwd in Microsoft Excel en Outlook met verschillende Macro's.

1.2 Context Opdracht

GoodTimeGifts heeft ervoor gekozen om het systeem in Java te bouwen. Dit zodat de gebruikers ongeacht welk systeem(Windows, Mac) de applicatie kunnen gebruiken. Verder moet het systeem Multi user zijn. Het systeem moet dus meerdere gebruikers tegelijk aankunnen en zorgen dat deze gebruikers dezelfde informatie kunnen zien. Er zal een database moeten worden gerealiseerd om de gegevens op te kunnen slaan. Er moet daarnaast gezorgd worden dat de applicatie snel werkt. Dit zal gecontroleerd worden volgens de hiervoor opgesteld performance eisen. Bij het maken van het ontwerp zal er rekening gehouden moeten worden met de performance.

Taken die in het huidige systeem gedaan worden zijn:

- Importeren verkoop orders;
- Maken facturen;
- Beheren voorraad;
- Maken creditnota's;
- Exporteren orderstatus;
- Updates orderstatus versturen naar klant;
- Order fulfilment;
- CRM;
- Inkoop.

Binnen de scope van het afstudeerproject vallen de volgende functionaliteiten:

- Importeren verkoop orders;
- Maken facturen;
- Beheren voorraad.

Buiten de scope het afstudeerproject vallen de volgende functionaliteiten:

- Maken creditnota's;
- Exporteren orderstatus;
- Updates orderstatus versturen naar klant;
- Order fulfilment;
- CRM;
- Inkoop.

Er is gekozen voor deze verdeling omdat dit de belangrijkste functionaliteit is. De functionaliteit binnen de scope zorgt voor een goede basis, wanneer dit af is zal de applicatie goed uit te breiden zijn met de functionaliteit die tijdens dit project buiten de scope valt.

1.3 Context Onderzoek

Tijdens het onderzoek zal de volgende hoofdonderzoeksvraag beantwoordt worden:

Wat zijn de eisen voor GoodBackOffice en op welke manier kunnen deze het beste gerealiseerd worden?

Dit door middel van het onderzoeken van de volgende deelvragen:

9. Hoe ziet het oude systeem eruit, zowel de architectuur als de gebruikte werkwijze?
10. Wat zijn de functionele systeemeisen voor het nieuwe systeem?
11. Wat zijn de performance eisen voor het nieuwe systeem?
12. Wat zijn de gebruikersinterface eisen voor het nieuwe systeem?
13. Welke bestaande oplossingen zijn er die aan de criteria en eisen voldoen?
14. Welke gebruikersinterface technologieën moeten er gebruikt worden om aan de eisen te voldoen?
15. Welke remote technologieën moeten er gebruikt worden om aan de eisen te voldoen?
16. Zijn de performance eisen haalbaar?

1.4 Huidige situatie

De huidige applicatie werkt momenteel met Microsoft Office Excel en Outlook en verschillende Macro's. De ordergegevens worden ingelezen via de order e-mails die ontvangen worden. Deze worden via een Macro geïmporteerd in de Bestelling Excel. Hier krijgt elke order een regel, vanuit de Bestelling is het mogelijk om de voorraad van een product te controleren, een factuur te maken en een verzendlabel uit te printen. Het is momenteel niet mogelijk om met meerdere personen de applicatie tegelijk te gebruiken. Hoe het volledige systeem precies werkt zal nog onderzocht worden tijdens het onderzoek.

2 Projectdefinitie

2.1 Projectdoelstellingen

GoodTimeGifts heeft besloten om zelf een nieuw order- en voorraad management systeem te maken genaamd "GoodBackOffice". Het doel is dat GoodBackOffice aan het einde van het project beschikt over de volgende functionaliteiten:

- Importeren verkoop orders;
- Maken facturen;
- Beheren voorraad.

Er dienen ook verschillende producten opgeleverd te worden tijdens het project. Zie hiervoor hoofdstuk 2.3 Producten, 2.5 Productdecompositiestructuur en 2.6 Productbeschrijving.

2.2 Project aanpak

Het project is opgedeeld in de volgende 5 fases:

2.2.1 Onderzoek

Tijdens het onderzoek zal de volgende onderzoeksvraag beantwoordt worden:

Wat zijn de eisen voor GoodBackOffice en op welke manier kunnen deze het beste gerealiseerd worden?

Deze vraag wordt beantwoordt doormiddel van de volgende deelvragen:

1. Hoe ziet het oude systeem eruit, zowel de architectuur als de gebruikte werkwijze?

Door interviews te houden met de opdrachtgevers en eindgebruikers over de werkwijze, te kijken naar de architectuur van het systeem, het opstellen een dataflowdiagram, een componentdiagram, een activiteitendiagram en een daarbij horende beschrijving zal deze deelvraag onderzocht worden. Er zal ook enige tijd meegekeken worden met een eindgebruiker terwijl het oude systeem gebruikt wordt. Dit zal een duidelijk beeld scheppen van het huidige systeem.

2. Wat zijn de functionele systeemeisen voor het nieuwe systeem?

Deze deelvraag zal onderzocht worden door interviews te houden met de opdrachtgevers, eindgebruikers en de applicatie beheerder over de systeemeisen en deze om te zetten in requirements.

3. Wat zijn de performance eisen voor het nieuwe systeem?

Door interviews te houden met de opdrachtgevers en eindgebruikers over de performance eisen en deze om te zetten in requirements zal deze deelvraag onderzocht worden.

4. Wat zijn de gebruikersinterface eisen voor het nieuwe systeem?

Deze deelvraag zal onderzocht worden door interviews te houden met de opdrachtgevers en eindgebruikers over de gebruikersinterface eisen en deze om te zetten in requirements.

5. Welke bestaande oplossingen zijn er die aan de criteria en eisen voldoen?

Deze deelvraag zal doormiddel van de volgende stappen onderzocht worden:

1. Maak een lijst van de op het internet gevonden beschikbare en belangrijkste backend applicaties.
2. Verklein de lijst door op hoofd requirements te controleren. De hoofd requirements zijn een deelverzameling van de requirements en worden in samenspraak met de opdrachtgever bepaald.

3. Verklein de lijst door gedetailleerd onderzoek van de applicatie specificaties versus de requirements.
4. Aanbeveling voor zelf te maken of bestaande applicatie.

Indien er in een van de bovenstaande stappen geen bestaande applicatie overblijft zal deze deelvraag afgerond zijn met als aanbeveling te kiezen voor een zelf te maken applicatie.

6. Welke gebruikersinterface technologieën moeten er gebruikt worden om aan de eisen te voldoen?

Door te kijken naar de verschillende mogelijkheden en te kiezen voor de technologie die het beste past bij de gebruikersinterface eisen zal deze deelvraag onderzocht worden.

7. Welke remote technologieën moeten er gebruikt worden om aan de eisen te voldoen?

Door te kijken naar de verschillende mogelijkheden en te kiezen voor de technologie die het beste past bij de applicatie eisen zal deze deelvraag onderzocht worden.

8. Zijn de performance eisen haalbaar?

Door te meettesten te doen naar de gekozen architectuur zal onderzocht worden of de gekozen architectuur aan deze eisen voldoet.

2.2.2 Ontwerp

Synchroon met de onderzoeksfase zal de ontwerpfase gestart worden. In deze ontwerpfase zal van het nieuwe systeem in grote lijnen ontworpen worden. Dit door middel van het opstellen van verschillende diagrammen(ERD, Domeindigram, Componentdiagram, etc.) en Wireframes.

2.2.3 Implementatie

Tijdens de implementatie fase zal de daadwerkelijke applicatie gemaakt worden. Dit gebeurt met behulp van een variatie op de Scrum ontwikkelmethode (Schwaber & Sutherland, 2013). In hoofdstuk 2.8 Projectmanagement is dit verder uitgewerkt. Een sprint begint altijd met het bepalen van de requirements die in de sprint van toepassing zijn, hierin worden ook de requirements meegenomen die nog niet volledig gerealiseerd zijn in de vorige sprint. Hierna zullen er enkele ontwerpbeslissingen plaatsvinden met betrekking tot de systeemeisen. Hierna volgt het implementatie deel. Een sprint wordt afgesloten met een presentatie van de gerealiseerde systeemeisen in de vorm van een demo. Op deze manier is er nauwe samenwerking tussen de partijen en blijft iedereen betrokken bij het project waardoor het eindresultaat ook echt iets is waar de eindgebruikers en de opdrachtgever baat bij hebben. Tijdens de implementatie fase zullen er ook verschillende tests uitgevoerd worden. De fase wordt afgesloten met een hardening sprint waarin de applicatie klaar gemaakt wordt voor oplevering

2.2.4 Oplevering

De opleveringsfase begint nadat de implementatie fase is afgesloten. Deze fase begint met een grote test waarin het gehele applicatie getest wordt. Ook vindt er een acceptatie test plaats met de opdrachtgever. Indien de opdrachtgever de applicatie accepteert zal de applicatie opgeleverd worden. De applicatie zal nog niet beheer en gebruik genomen worden. Hiervoor dienen eerst de functionaliteiten die voor dit project buiten de scope vallen ontwikkeld te worden. In deze fase zal zowel een testrapportage worden opgeleverd als een gebruikershandleiding.

2.2.5 Afronding

Tijdens deze fase zal de scriptie afgerond worden en ingeleverd worden. De eindpresentatie zal ook in deze fase gegeven worden. De fase wordt afgesloten met een reflectie over het gehele project.

2.3 Producten

Het project is opgedeeld in de volgende hoofdproducten en tussenproducten:

- Projectdocumentatie
 - Scriptie
 - PID
 - Onderzoekverslag
- Applicatie documentatie
 - Test rapportage
 - Gebruikershandleiding
- Ontwerp
- Applicatie
 - Factuur Management
 - Order Management
 - Voorraad Management

2.4 Scope van het Project

Binnen de scope van het afstudeerproject vallen de volgende functionaliteiten:

- Importeren verkoop orders;
- Maken facturen;
- Beheren voorraad.

Buiten de scope het afstudeerproject vallen de volgende functionaliteiten:

- Maken creditnota's;
- Exporteren orderstatus;
- Updates orderstatus versturen naar klant;
- Order fulfilment;
- CRM;
- Inkoop.

GoodBackOffice moet na het voltooien van dit project uit te bereiden zijn deze functionaliteit. Hier dient dus in het ontwerp rekening mee gehouden te worden.

De organisatorische context van het project blijft binnen GoodTimeGifts, Linuxcompany en Fontys Hogescholen en betreft de volgende personen:

GoodTimeGifts – Peter Wijnen – Opdrachtgever – Eindgebruiker – Productowner

GoodTimeGifts – Helene Wijnen – Eindgebruiker

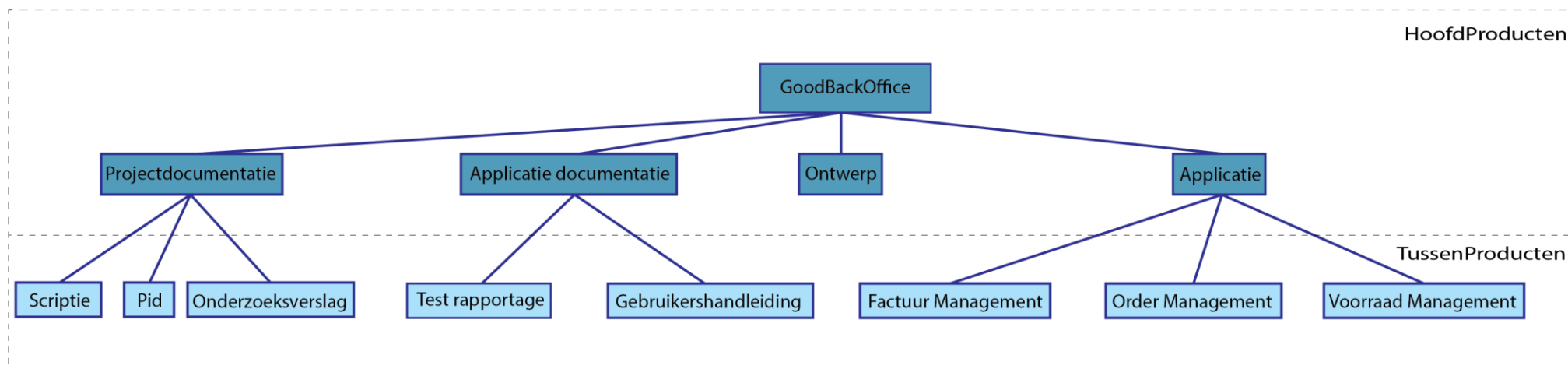
GoodTimeGifts – Ard Wijnen – Ontwikkelaar – Afstudeerder – Scrummaster

Linuxcompany – Marc Henssen – Bedrijfsbegeleider – Applicatie beheerder

Fontys Hogescholen – Marcel Meesters – Docentbegeleider – 1^{ste} Assessor

Fontys Hogescholen – Jelle Oosterkamp – 2^{de} Assessor

2.5 Productdecompositiestructuur



Figuur 1 Productdecompositiestructuur

2.6 Productbeschrijving

De resultaten van alle producten zullen met de opdrachtgever, bedrijfsbegeleider en de docentbegeleider besproken worden. De feedback die hieruit volgt zal meegenomen worden in de eindversie van de producten. Er zal ook bij eventuele twijfel over de producten ondersteuning gevraagd worden aan een of meerdere van de begeleiders.

2.6.1 Projectdocumentatie

Projectdocumentatie bestaat uit een scriptie, een PID en een onderzoeksverslag. Als eerste zal de PID gemaakt worden, het onderzoeksverslag wordt hierna gemaakt, de scriptie wordt als laatste gemaakt. In het onderzoeksverslag worden de resultaten, conclusies en aanbevelingen van het onderzoek verwerkt. In de scriptie worden de PID en het onderzoeksverslag als bijlage ingevoegd. Met de scriptie zal begonnen wanneer het onderzoeksverslag is afgerond. De scriptie is pas klaar als alle andere onderdelen volledig afgerond zijn.

2.6.2 Ontwerp

Vanuit het onderzoek en het onderzoeksverslag zal een ontwerp volgen. Hierin worden de systeemeisen en de aanbevelingen omgezet in een ontwerp. Het doel hiervan is het bepalen van de architectuur, de verschillende componenten en interfaces hierbinnen. Het ontwerp bevat onder andere een ERD, Domeindigram, Component diagram en een Swimlane activiteiten diagram. Wireframes zullen er ook gemaakt worden zodat er een duidelijk beeld ontstaat hoe het de applicatie geïmplementeerd dient te worden en hoe deze er uiteindelijk uitkomt te zien. De Wireframes worden ook getest via usertests. In het ontwerp zal ook rekening gehouden worden met de gestelde performance eisen, hier wordt ook onderzoek naar gedaan worden.

2.6.3 Applicatie

Applicatie kan gebouwd worden nadat het ontwerp klaar is. Dit is het geïmplementeerde eindresultaat van het systeem. Hierin moeten straks ook alle systeemeisen terug te vinden zijn in de vorm van functionaliteit, usability en performance. Ook dienen de verschillende onderdelen getest te worden middels unittests en usertests.

2.6.4 Applicatie documentatie

Dit bestaat uit de Test rapportage en de Gebruikershandleiding. Deze documenten zullen in de opleveringsfase worden ontwikkeld.

2.7 Projectbudget

Het afstudeertraject duurt 8 maanden. Er wordt in de periode minimaal 600uur aan het project besteedt. De beschikbare tijd is opgedeeld in een aantal verschillende fases, waarin aan specifieke onderdelen van het project wordt gewerkt.

2.8 Projectmanagement

Tijdens implementatie fase van het project wordt de Scrum ontwikkelmethode (Schwaber & Sutherland, 2013) gebruikt om agile te kunnen ontwikkelen met sprints van 2 weken. De ontwikkelaar vervult ook nog de rol van scrummaster. De opdrachtgever vervult hierin ook de rol als productowner. Hij zal meedoen met de pokersessie voor het bepalen van de story points en aan de daily standup meeting. In de backlog worden de requirements geprioriteerd

met de MoSCoW methode (Brennan, 2009). Een sprint begint altijd met het bepalen van de requirements die in de sprint van toepassing zijn, Hierin worden ook de requirements meegenomen die nog niet volledig gerealiseerd zijn in de vorige sprint. Daarna zullen er enkele ontwerpbeslissingen plaatsvinden met betrekking tot de systeemeisen. Hierna volgt het implementatie deel. Een sprint wordt afgesloten met een presentatie van de gerealiseerde systeemeisen in de vorm van een demo en een retrospectives waarin reflectie gegeven word op het verloop van de sprint. Tijdens de retrospectives zijn de productowner en het project lid aanwezig.

2.9 Wijzigingsbudget

Indien er wijzigingen voordoen die de planning doen veranderen, zullen er aanpassingen gemaakt moeten worden om ervoor te zorgen dat het project binnen de gestelde periode afgerond kan worden. Dit zal gebeuren in overleg tussen de ontwikkelaar, de opdrachtgever en de begeleiders. Door het gebruik van Scrum ontwikkelmethode (Schwaber & Sutherland, 2013) is het mogelijk gedeeltes van de op te leveren functionaliteit door te verschuiven naar een volgend project.

2.10 Benodigde Resources

De afstudeerder werkt alleen aan het project, maar doet wel verslag aan de begeleiders over de voortgang van het project. De resources die gebruikt worden zijn:

- Computer met ontwikkelomgeving met Netbeans 8.0.1 voor het implementeren van de software.
- Microsoft Office 2010 voor de projectdocumentatie.
- Adobe Illustrator voor het ontwikkelen van de Diagrammen en de Wireframes.
- Combinatie van test en acceptatie omgeving in vorm server met database en terminal server omgeving voor de applicatie.

2.11 Uitsluitingen

De applicatie moet na het voltooien van dit project uit te bereiden zijn met de volgende functionaliteit: Maken creditnota's, Exporteren orderstatus, Updates orderstatus versturen naar klant, Order fulfilment en CRM. Deze functionaliteit valt voor dit project buiten de scope.

2.12 Afhankelijkheden

Er moet een server aanwezig zijn met een database, zodat er een plek is waar de gegevens opgeslagen kunnen worden. De ontwikkelaar heeft tijdens de onderzoeksfase toegang tot het huidige systeem zodat hij hier een duidelijk beeld van kan krijgen.

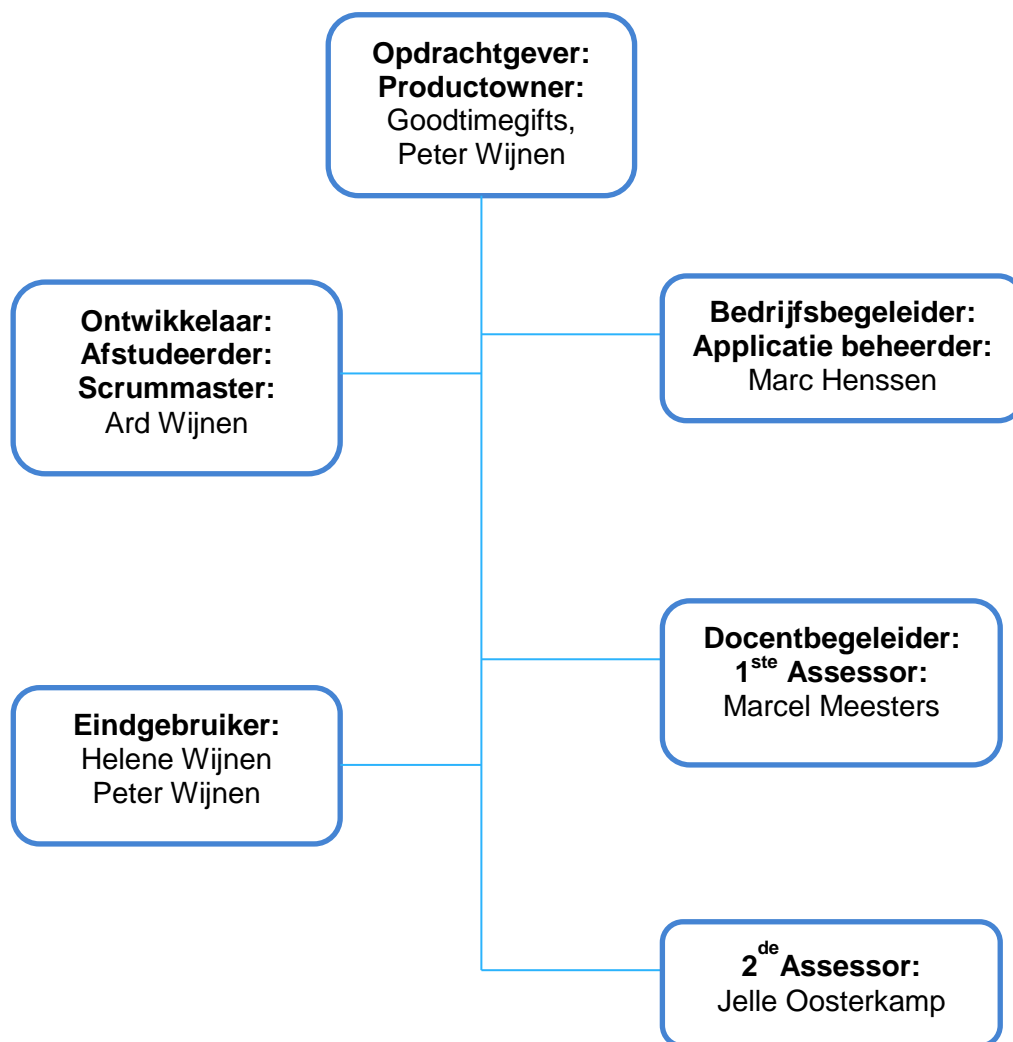
2.13 Randvoorwaarden

Er moet medewerking zijn vanuit zowel de eindgebruikers als de opdrachtgever om tot een goed eindproduct te komen.

2.14 Aannames

De docentbegeleider heeft tijd om de scriptie minimaal 2 maal voor de uiteindelijke opleverdatum na te kijken. De bedrijfsbegeleider heeft wekelijks de tijd om de voortgang van het project door te nemen en feedback te geven op de gemaakte voortgang. De eindgebruikers en de opdrachtgever zijn bereid om tweewekelijks na de start van de implementatie de demo te volgen en hierover feedback te geven.

3 Projectorganisatiestructuur



Figuur 2 Projectorganisatiestructuur

3.1 Opdrachtgever, Productowner

De opdrachtgever heeft het project bedacht en gecreëerd. Hij overziet het project en zorgt ervoor dat het eindresultaat voldoet aan de eisen en overeenkomt met wat origineel was gepland. Wijzigingen aan het project zullen worden overlegd met de opdrachtgever, om er voor te zorgen dat hij in overeenstemming is met het plan en het eindresultaat. Hij is verantwoordelijk voor de verankering van het project in de organisatie. In de rol als productowner zal hij meedoen aan de dagelijkse stand-up meetings, mee bepalen welke requirements in de sprints meegenomen worden en aan de retrospectives.

3.2 Ontwikkelaar, Afstudeerder, Scrummaster

De afstudeerder is verantwoordelijk voor het uitvoeren van het gehele project. Hij is verantwoordelijk voor het inplannen en regelen van afspraken met alle eindgebruikers zowel voor het testen als voor het bespreken en interviewen van de wensen, eisen en huidige werkwijzen. Heeft wekelijks contact met de bedrijfsbegeleider over de voortgang van het project en tweewekelijks met de docentbegeleider. In de rol als scrummaster moet hij ervoor zorgen dat het scrumproces correct verloopt. Hij heeft dan ook de leidende rol tijdens de retrospectives.

3.3 Eindgebruiker

De eindgebruikers zijn de personen die de applicatie in productie gaan nemen wanneer deze voltooid is. Zij moeten ervoor zorgen de afstudeerder zo goed mogelijk te informeren over de wensen. Tevens moeten zij aanwezig zijn bij de tweewekelijkse demo en deze voorzien van feedback.

3.4 Bedrijfsbegeleider, Applicatie beheerder

De bedrijfsbegeleider is het eerste aanspreekpunt voor vragen en feedback over het project vanuit de afstudeerder. Hij adviseert de docentbegeleider over de beoordelingen van de afstudeerproducten. De bedrijfsbegeleider is ook aanwezig bij de bedrijfsbezoeken van de docentbegeleider. De functie van de applicatie beheerder zal pas ingaan als de gehele applicatie voltooid is inclusief de functionaliteit die voor nu buiten de scope valt. Wel zal er tijdens het onderzoek met de applicatie beheerder gesproken worden over de wensen betreffende de configuratie.

3.5 Docentbegeleider, 1^{ste} Assessor

De docentbegeleider is het tweede aanspreek punt voor vragen en feedback over het project vanuit de afstudeerder. Hij geeft ook goedkeuringen over de PID. Tevens beoordeelt hij ook samen met de 2^{de} assessor de afstudeerproducten. De docentbegeleider begeleidt ook de afstudeerder tijdens het schrijven van de scriptie.

3.6 2^{de} Assessor

De 2^{de} assessor beoordeelt samen met de 1^{ste} assessor de afstudeerproducten.

4 Projectbeheersing

4.1 Rapportage en Voortgangsbewaking

Tijdens het project worden de volgende documenten eerst door de docentbegeleider van feedback voorzien voordat de definitieve versie wordt ingeleverd voor beoordeling:

- PID;
- Onderzoeksverslag;
- Scriptie.

Er is een wekelijkse bespreking met de bedrijfsbegeleider over de voortgang van het project. Tevens is er een tweewekelijkse bespreking met de docentbegeleider betreffende de voortgang.

Er zal ook een urenverantwoording bijgehouden worden, zodat er een duidelijk overzicht is van de gemaakte uren tijdens dit project.

4.2 Toleranties

Gedurende het project zal er een tolerantie zijn van 10% per fase van het tijdsbudget. Voor elke fase die het geplande budget overschrijdt, wordt het budget van de volgende fase ingekort. Uitloop in de onderzoeks- of ontwerpfase heeft als gevolg dat er sprints en functionaliteit wegvallen in de implementatiefase. De Implementatie fase kan niet uitlopen.

4.3 Kwaliteitsbewaking

Het opstellen van de systeemeisen gebeurt doormiddel van interviews van de eindgebruikers en de opdrachtgever en analyse van het huidige systeem. Deze worden met behulp van MoSCoW methode (Brennan, 2009) geprioriteerd door middel van overleg tussen de ontwikkelaar en de opdrachtgever. Het project is voltooid wanneer alle Must en Should systeemeisen zijn geïmplementeerd. De feedback die tijdens de demo aan het einde van een sprint gegeven wordt zal meegenomen worden in de lijst met systeemeisen. Tevens worden de gemaakt functionaliteit gecontroleerd via Unittest.

De projectdocumentatie documenten die ingeleverd moeten worden zullen eerst voorzien worden van feedback door de begeleiders.

Er zal ook een gebruikershandleiding gemaakt worden zodat de eindgebruikers snel het systeem zullen leren en naslagwerk hebben indien dit nodig is.

4.4 Wijzigingsprocedure

Tijdens de implementatiefase wordt er gewerkt met de Scrum ontwikkelmethode (Schwaber & Sutherland, 2013), aan het einde van elke sprint zullen de gegeven feedback meegenomen worden in de backlog.

Voor wijzigingen in de systeemeisen zal eerst gekeken worden of dit effect heeft op het ontwerp. Van hieruit zal besloten worden of er een herontwerp nodig is of niet.

4.5 Risicomanagement

In de onderstaande tabel zijn de risico's te vinden, per risico zijn cijfers toegekend.

Voor Kans wordt een cijfer gegeven tussen de 1 en de 5. 1 is hierin de kleinste kans en 5 een zeer grote kans. Voor Impact wordt een cijfer gegeven tussen de 1 en de 5. 1 is hierin de kleinste impact en 5 een zeer grote impact. Een vermenigvuldiging van Kans en Impact levert de Ernst op tussen de 1 en 25. Tevens wordt voor elk risico een oplossing gegeven die gebruikt kan worden mocht het risico zich voor doen.

Risico	Beschrijving	Kans	Impact	Ernst	Oplossing
Tijdsnood	Het project kan niet volledig afgerond worden in de beschikbare tijd.	3	2	6	Er is tijd is geroosterd voor eventuele uitloop. Daarnaast is er mogelijkheid om te schuiven met de implementatie doordat deze doormiddel van de Scrum ontwikkelmethode (Schwaber & Sutherland, 2013) ontwikkeld wordt. Mocht het zo zijn dat er delen niet gedaan kunnen worden dan zullen deze in de volgende ontwikkelsessies meegenomen worden. Bij een grote uitloop zal er met de begeleiders overlegd moeten worden wat het effect is met betrekking tot het afstuderen.
Onvolledige systeemeisen	Tijdens het opstellen worden er systeemeisen vergeten.	4	1	4	Onderzoeken of er een herontwerp nodig. Zo ja, het systeem opnieuw ontwerpen. Zo nee, Systeemeisen meenemen in de implementatie fase.
Veranderende systeemeisen	Tijdens de implementatie fase veranderen de wensen van de opdrachtgever	3	1	3	Onderzoeken of er een herontwerp nodig. Zo ja, het systeem opnieuw ontwerpen. Zo nee, systeemeisen meenemen in de implementatie fase.
Performance	Tijdens de implementatie fase ontdekt men dat de performance van het systeem niet goed genoeg is.	2	2	4	Onderzoeken of er een herontwerp nodig. Zo ja, het systeem opnieuw ontwerpen. Zo nee, het probleem oplossen in de implementatie fase. Tijdens de onderzoeksfase zal hier al onderzoek naar gedaan worden.
Ontwerpfouten	Tijdens de implementatie fase wordt duidelijk dat er fouten zitten in het ontwerp.	2	5	10	Herontwerpen van het ontwerp.

Tabel 1 Risicomanagement

4.6 Planning

In de planning zijn de 5 fases van het project terug te vinden. De 5 fases beginnen nadat de PID is voltooid. Het schrijven van de scriptie begint wanneer de onderzoeksfase is afgerond. De eerste versie van de scriptie moet afgerond zijn in week 40.

Onderzoek

De onderzoeksfase loopt van week 10 tot en met week 20.

Ontwerp

De ontwerpfase loopt van week 17 tot en met week 20.

Implementatie

De implementatie fase loopt van week 21 tot en met week 38.

Oplevering

De opleveringsfase loopt van week 39 tot en met week 40.

Nadat deze fases zijn afgerond zijn er nog 3 weken gepland voor het aanpassen van de scriptie en houden van de afstudeerzitting.

Reflectie

De reflectiefase loopt in week 44

Groep	Onderdelen	Weeknummers																																										
		7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44					
Project Documentatie																																												
	PID																																											
	Scriptie																																											
Onderzoek																																												
	Onderzoeksverslag																																											
	Onderzoek																																											
Ontwerp																																												
Implementatie																																												
	Sprint 1																																											
	Sprint 2																																											
	Sprint 3																																											
	Sprint 4																																											
	Sprint 5																																											
	Sprint 6																																											
	Sprint 7																																											
	Hardening																																											
Oplevering																																												
Afstuderen																																												
	Scriptie aanpassen																																											
	Afstudeerzitting																																											
Reflectie																																												

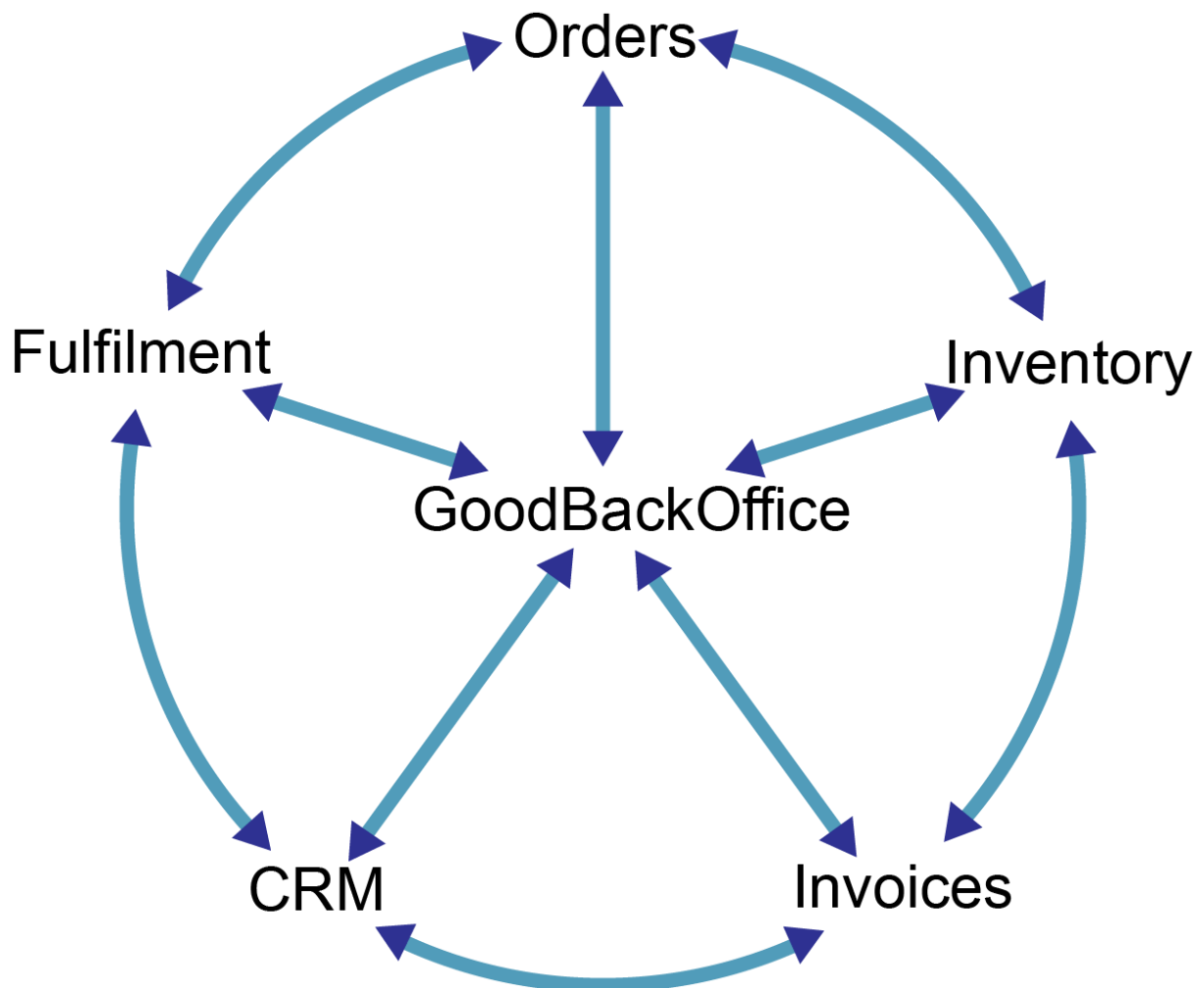
Tabel 2 Planning

5 Referenties

Brennan, K. (2009). *A Guide to the Business Analysis Body of Knowledge*. International Institute

of Business Analysis. Opgeroepen op 18 2, 2015

Sutherland, J., & Schwaber, K. (2011). The scrum guide. *The Definitive Guide to Scrum: The Rules of the Game*. Opgeroepen op 18 2, 2015, van The home of Scrum:
<http://www.scrumguides.org/docs/scrumguide/v1/Scrum-Guide-US.pdf>

Bijlage II**GOODBACKOFFICE****ORDER EN VOORRAAD MANAGEMENTSYSTEEM****Onderzoeksverslag**

Project
Datum
Auteur
Versie
Status
Bedrijf

GoodBackOffice
24-4-2015
Ard Wijnen
1.0
GoodTimeGifts V.O.F.

Documenthistorie

Versie	Datum	Wijzigingen
0.1	22-4-2015	Geen wijzigingen
1.0	24-4-2015	Feedback verwerkt van Marc Henssen
1.1	26-4-2015	Feedback verwerkt van Marcel Meesters

Inhoudsopgave

1 Inleiding.....	7
2 Deelvragen.....	12
3 Onderzoeksmethoden	13
3.1 Geplande onderzoeksmethode	7
3.1.1 Hoe ziet het oude systeem eruit, zowel de architectuur als de gebruikte werkwijze?	7
3.1.2 Wat zijn de functionele systeemeisen voor het nieuwe systeem?	7
3.1.3 Wat zijn de performance eisen voor het nieuwe systeem?	7
3.1.4 Wat zijn de gebruikersinterface eisen voor het nieuwe systeem?	7
3.1.5 Welke bestaande oplossingen zijn er die aan de criteria en eisen voldoen?	7
3.1.6. Welke gebruikersinterface technologieën moeten er gebruikt worden om aan de eisen te voldoen?	7
3.1.7 Welke remote technologieën moeten er gebruikt worden om aan de eisen te voldoen?	8
3.1.8 Zijn de performance eisen haalbaar?	8
3.2 Uitvoering	8
3.2.1 Hoe ziet het oude systeem eruit, zowel de architectuur als de gebruikte werkwijze?	8
3.2.2 Wat zijn de functionele systeemeisen voor het nieuwe systeem?	9
3.2.3 Wat zijn de performance eisen voor het nieuwe systeem?	9
3.2.4 Wat zijn de gebruikersinterface eisen voor het nieuwe systeem?	9
3.2.5 Welke bestaande oplossingen zijn er die aan de criteria en eisen voldoen?	10
3.2.6 Welke gebruikersinterface technologieën moeten er gebruikt worden om aan de eisen te voldoen?	10
3.2.7 Welke remote technologieën moeten er gebruikt worden om aan de eisen te voldoen?	10
3.2.8 Zijn de performance eisen haalbaar?	11
4 Resultaten	15
4.1 Hoe ziet het oude systeem eruit, zowel de architectuur als de gebruikte werkwijze? ..	12
4.2 Wat zijn de functionele systeemeisen voor het nieuwe systeem?	16
4.3 Wat zijn de performance eisen voor het nieuwe systeem?	16
4.4 Wat zijn de gebruikersinterface eisen voor het nieuwe systeem?	16
4.5 Welke bestaande oplossingen zijn er die aan de criteria en eisen voldoen?	17
4.6 Welke gebruikersinterface technologieën moeten er gebruikt worden om aan de eisen te voldoen?	19
4.7 Welke remote technologieën moeten er gebruikt worden om aan de eisen te voldoen?	19
4.8 Zijn de performance eisen haalbaar?	20
5 Conclusie	17
5.1 Hoe ziet het oude systeem eruit, zowel de architectuur als de gebruikte werkwijze? ..	23
5.2 Wat zijn de functionele systeemeisen voor het nieuwe systeem?	23

5.3 Wat zijn de performance eisen voor het nieuwe systeem?	23
5.4 Wat zijn de gebruikersinterface eisen voor het nieuwe systeem?	23
5.5 Welke bestaande oplossingen zijn er die aan de criteria en eisen voldoen?	23
5.6 Welke gebruikersinterface technologieën moeten er gebruikt worden om aan de eisen te voldoen?	23
5.7 Welke remote technologieën moeten er gebruikt worden om aan de eisen te voldoen?	23
5.8 Zijn de performance eisen haalbaar?	23
6 Referenties	24
Bijlage I	25

1 Inleiding

Dit onderzoeksverslag is een onderdeel van het afstudeerproject GoodBackOffice. Voor uitgebreide informatie en uitleg over het project zie het Project Initiatie Document.

Dit onderzoek is uitgevoerd voor GoodTimeGifts. Het doel van het onderzoek is om een duidelijk beeld te vormen wat de eisen moeten worden van GoodBackOffice en hoe deze het beste gerealiseerd kunnen worden. Dit komt terug in de hoofdvraag van het onderzoek.

Om een goed beeld te krijgen van de wensen en behoeften van de opdrachtgever zijn er een aantal interviews gehouden die een groot deel van het onderzoek vormen. Verder is er voor het onderzoek gebruik gemaakt van deskresearch en metingen.

Tijdens het onderzoek werd er gezocht naar het antwoord op de volgende hoofdvraag:

Wat zijn de eisen voor GoodBackOffice en op welke manier kunnen deze het beste gerealiseerd worden?

Dit door middel van het onderzoeken van de volgende deelvragen:

1. Hoe ziet het oude systeem eruit, zowel de architectuur als de gebruikte werkwijze?
2. Wat zijn de functionele systeemeisen voor het nieuwe systeem?
3. Wat zijn de performance eisen voor het nieuwe systeem?
4. Wat zijn de gebruikersinterface eisen voor het nieuwe systeem?
5. Welke bestaande oplossingen zijn er die aan de criteria en eisen voldoen?
6. Welke gebruikersinterface technologieën moeten er gebruikt worden om aan de eisen te voldoen?
7. Welke remote technologieën moeten er gebruikt worden om aan de eisen te voldoen?
8. Zijn de performance eisen haalbaar?

In hoofdstuk 5 zijn de conclusies te vinden en de antwoorden op de bovenstaande deelvragen.

2 Deelvragen

Tijdens het onderzoek werd de volgende onderzoeksvraag beantwoord:

Wat zijn de eisen voor GoodBackOffice en op welke manier kunnen deze het beste gerealiseerd worden?

Deze vraag werd beantwoord doormiddel van de volgende deelvragen:

- 1. Hoe ziet het oude systeem eruit, zowel de architectuur als de gebruikte werkwijze?**
Dit werd onderzocht zodat er een duidelijk beeld ontstaat van de werkwijze binnen het oude systeem en welke onderdelen juist wel en niet in GoodBackOffice moeten komen.
- 2. Wat zijn de functionele systeemeisen voor het nieuwe systeem?**
Dit werd onderzocht zodat er duidelijk werd wat er precies geïmplementeerd moet worden in GoodBackOffice.
- 3. Wat zijn de performance eisen voor het nieuwe systeem?**
Dit werd onderzocht zodat duidelijk werd wat de snelheidseisen zijn van GoodBackOffice.
- 4. Wat zijn de gebruikersinterface eisen voor het nieuwe systeem?**
Dit werd onderzocht zodat duidelijk werd over welke pagina's en overzichten GoodBackOffice moet beschikken.
- 5. Welke bestaande oplossingen zijn er die aan de criteria en eisen voldoen?**
Deze deelvraag wordt onderzocht om te kijken of er geen bestaande applicatie zijn die beter geschikt zijn dan een zelf te maken applicatie voor dit project.
- 6. Welke gebruikersinterface technologieën moeten er gebruikt worden om aan de eisen te voldoen?**
Hierdoor wordt er gekozen voor de beste gebruikersinterface technologie voor dit project.
- 7. Welke remote technologieën moeten er gebruikt worden om aan de eisen te voldoen?**
Hierdoor wordt er gekozen voor de meeste geschikte remote technologie voor dit project.
- 8. Zijn de performance eisen haalbaar?**
Door performance meettesten te doen van het ontwerp werd duidelijk of deze zal voldoen aan de gestelde eisen.

3 Onderzoeksmethoden

3.1 Geplande onderzoeksmethode

3.1.1 Hoe ziet het oude systeem eruit, zowel de architectuur als de gebruikte werkwijze?

Door interviews te houden met de opdrachtgevers en eindgebruikers over de werkwijze, te kijken naar de architectuur van het systeem, het opstellen van een dataflowdiagram, een componentdiagram, een activiteitendiagram en een daarbij horende beschrijving zal deze deelvraag onderzocht worden. Er zal ook enige tijd meegekeken worden met een eindgebruiker terwijl het oude systeem gebruikt wordt. Dit zal een duidelijk beeld scheppen van het huidige systeem.

3.1.2 Wat zijn de functionele systeemeisen voor het nieuwe systeem?

Deze deelvraag zal onderzocht worden door interviews te houden met de opdrachtgevers, eindgebruikers en de applicatiebeheerder over de systeemeisen en deze om te zetten in requirements.

3.1.3 Wat zijn de performance eisen voor het nieuwe systeem?

Door interviews te houden met de opdrachtgevers en eindgebruikers over de performance eisen en deze om te zetten in requirements zal deze deelvraag onderzocht worden.

3.1.4 Wat zijn de gebruikersinterface eisen voor het nieuwe systeem?

Deze deelvraag zal onderzocht worden door interviews te houden met de opdrachtgevers en eindgebruikers over de gebruikersinterface eisen en deze om te zetten in requirements.

3.1.5 Welke bestaande oplossingen zijn er die aan de criteria en eisen voldoen?

Deze deelvraag zal doormiddel van de volgende stappen onderzocht worden:

1. Maak een lijst van de op het internet gevonden beschikbare en belangrijkste backend applicaties.
2. Verklein de lijst door op hoofd requirements te controleren. De hoofd requirements zijn een deelverzameling van de requirements en worden in samenspraak met de opdrachtgever bepaald.
3. Verklein de lijst door gedetailleerd onderzoek van de applicatie specificaties versus de requirements.
4. Aanbeveling voor zelf te maken of bestaande applicatie.

Indien er in een van de bovenstaande stappen geen bestaande applicatie overblijft, zal deze deelvraag afgerond zijn met als aanbeveling te kiezen voor een zelf te maken applicatie.

3.1.6 Welke gebruikersinterface technologieën moeten er gebruikt worden om aan de eisen te voldoen?

Door te kijken naar de verschillende mogelijkheden en te kiezen voor de technologie die het beste past bij de gebruikersinterface eisen zal deze deelvraag onderzocht worden.

3.1.7 Welke remote technologieën moeten er gebruikt worden om aan de eisen te voldoen?

Door te kijken naar de verschillende mogelijkheden en te kiezen voor de technologie die het beste past bij de applicatie eisen zal deze deelvraag onderzocht worden.

3.1.8 Zijn de performance eisen haalbaar?

Door de meettesten te doen naar de gekozen architectuur zal onderzocht worden of de gekozen architectuur aan deze eisen voldoet.

3.2 Uitvoering

3.2.1 Hoe ziet het oude systeem eruit, zowel de architectuur als de gebruikte werkwijze?

Deze deelvraag is onderzocht door middel van interviews met de eindgebruikers en opdrachtgever. Ook is er meegekeken hoe de applicatie gebruikt wordt. Tijdens het interview zijn er open vragen gesteld zodat de geïnterviewde ook extra informatie konden vertellen en hierop doorgevraagd kon worden.

Tijdens het interview zijn de volgende vragen als leidraad gebruikt voor het gesprek:

-Kun je in grote lijnen vertellen wat het systeem momenteel doet?

Ordermanagement:

- Kun je kort het proces beschrijven?
- Welke overzichten zijn er beschikbaar?
- Welke schermen zijn er beschikbaar?
- Waar liggen de beslissingsmomenten?
- Door wie worden de beslissingsmomenten genomen? PC of gebruiker?
- Zijn er dingen in het systeem waar je van af wilt?
- Zijn er dingen die sneller of beter kunnen in jouw ogen?

Factuurmanagement:

- Kun je kort het proces beschrijven?
- Welke overzichten zijn er beschikbaar?
- Welke schermen zijn er beschikbaar?
- Waar liggen de beslissingsmomenten?
- Door wie worden de beslissingsmomenten genomen? PC of gebruiker?
- Zijn er dingen in het systeem waar je van af wilt?
- Zijn er dingen die sneller of beter kunnen in jouw ogen?

Voorraadmanagement:

- Kun je kort het proces beschrijven?
- Welke overzichten zijn er beschikbaar?
- Welke schermen zijn er beschikbaar?
- Waar liggen de beslissingsmomenten?
- Door wie worden de beslissingsmomenten genomen? PC of gebruiker?
- Zijn er dingen in het systeem waar je van af wilt?
- Zijn er dingen die sneller of beter kunnen in jouw ogen?

3.2.2 Wat zijn de functionele systeemeisen voor het nieuwe systeem?

Deze deelvraag is onderzocht door middel van interviews met de eindgebruikers en opdrachtgever. Tijdens het interview zijn er open vragen gesteld zodat de geïnterviewde ook extra informatie konden vertellen en hierop doorgevraagd kon worden.

Tijdens het interview zijn de volgende vragen als leidraad gebruikt voor het gesprek:

Kun je in grote lijnen vertellen wat het systeem moet doen?

Ordermanagement:

- Vanuit hoeveel webshops moeten orders geïmporteerd worden?
- Vanuit hoeveel verschillende webshop systemen moeten orders geïmporteerd worden?
- Hoeveel verschillende e-mail templates met order informatie zijn er?
- Hoeveel verschillende databases zijn er?
- Liggen er ergens keuze momenten die door een persoon bekeken moeten worden?

Factuurmanagement:

- Welke gegevens moeten op de factuur staan?
- Hoeveel procent van de facturen worden per post en per mail verstuurd?
- Is van elke factuur een pdf noodzakelijk?
- Zijn er terugkomende facturen?
- Moet een factuur nog eerst gecheckt worden voordat hij geprint wordt?

Voorraadmanagement:

- Wanneer moet er bij een bestelling de voorraad worden aangepast?
- Zijn er meerdere producten die één voorraadproduct hebben?
- Zijn er nog andere functionele eisen?

Bij het gesprek met de applicatiebeheerder zijn de volgende vragen als leidraad gebruikt:

- Welke gegevens moeten er via de configuratie aangepast kunnen worden?
- Waar moeten de configuratie gegevens aangepast worden?

3.2.3 Wat zijn de performance eisen voor het nieuwe systeem?

Deze deelvraag is onderzocht door middel van interviews met de eindgebruikers en opdrachtgever. Tijdens het interview is er open vraag gesteld zodat de geïnterviewde ook extra informatie konden vertellen en hierop doorgevraagd kon worden.

Tijdens het interview is de volgende vraag als leidraad gebruikt voor het gesprek:

- Voor welke onderdelen zijn er performance eisen en wat is hierbij de gestelde performance eis?

3.2.4 Wat zijn de gebruikersinterface eisen voor het nieuwe systeem?

Deze deelvraag is onderzocht door middel van interviews met de eindgebruikers en opdrachtgever. Tijdens het interview zijn er open vragen gesteld zodat de geïnterviewde ook extra informatie konden vertellen en hierop doorgevraagd kon worden.

Tijdens het interview zijn de volgende vragen als leidraad gebruikt voor het gesprek:

- Welke overzichten moeten er komen?
- Welke schermen moeten er komen?
- Welke gegevens moeten er in welk overzicht komen?
- Welke interface elementen zijn er nodig?

3.2.5 Welke bestaande oplossingen zijn er die aan de criteria en eisen voldoen?

Er is een lijst bestaande oplossingen gemaakt door in de Google Search Engine de volgende zoekwoorden in te vullen:

- Backoffice Ecommerce;
- Magento backoffice ;
- Backoffice Webshop;
- Backoffice Webshop Magento;
- Magento erp;
- Magento erp systeem;
- Erp Webshop;
- Erp software ecommerce.

De resultaten zijn in een checklijst met hoofd requirements ingevuld. Voor elk pakket is gekeken of ze aan deze hoofd requirements voldoen. De volgende hoofd requirements zijn gebruikt in de checklijst:

- Het pakket beschikt over een order management gedeelte;
- Het pakket beschikt over een factuur management gedeelte;
- Het pakket beschikt over een voorraad beheer gedeelte;
- Het pakket beschikt over een bestaande koppeling met Magento;
- Het pakket beschikt over een bestaande koppeling met Joomla Virtuemart;
- Het pakket werkt cross-platform;
- Het pakket is multi-user-friendly;
- Het pakket beschikt over een bestaande Parcelware integratie;
- Het pakket moet updates versturen naar klant;
- Het pakket moet status updates versturen naar Magento;
- Het pakket moet status updates versturen naar Joomla Virtuemart;
- Prijsindicatie zonder aanpassingen (eenmalig / per jaar).

3.2.6 Welke gebruikersinterface technologieën moeten er gebruikt worden om aan de eisen te voldoen?

Deze deelvraag is onderzocht door te kijken welke gebruikersinterface technologie het beste voldoet aan de gestelde eisen. Hierin zijn de eisen uit 3.2.4 meegenomen in een checklist. Ook is er een pros-and-cons lijst opgesteld waarin de volgende onderdelen zijn meegenomen:

- Roadmap van het product;
- Aantal bugs;
- Ontwikkelingsnelheid;
- Totale functionaliteit;
- Support.

3.2.7 Welke remote technologieën moeten er gebruikt worden om aan de eisen te voldoen?

Deze deelvraag is onderzocht door te kijken welke remote technologie het beste voldoet aan de gestelde eisen. Dit doormiddel van een checklist. In de checklist staan de functionele eisen die betrekking hebben op de remote communicatie.

- Ontwikkelingsnelheid;
- Totale functionaliteit;

- Cross platform usability.

3.2.8 Zijn de performance eisen haalbaar?

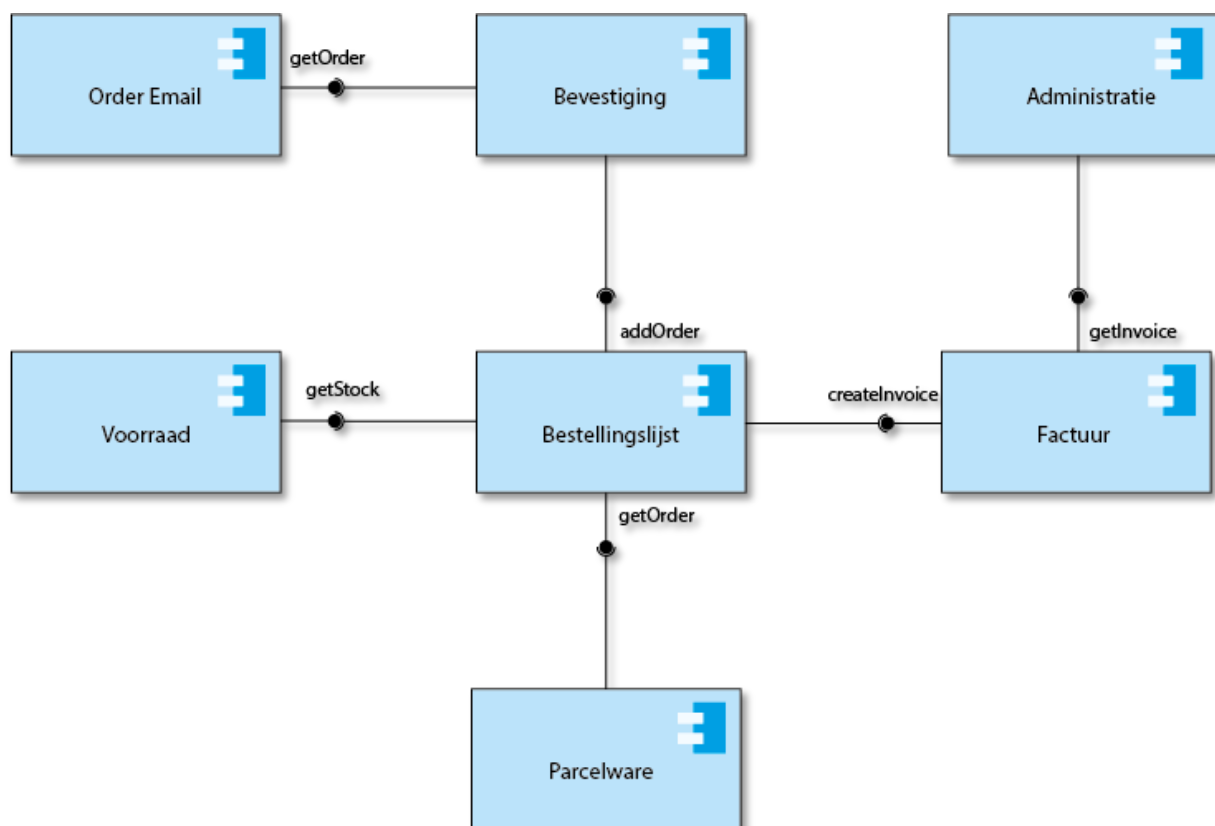
Dit onderzoek werd uitgevoerd op basis van de in 3.2.3 gestelde performance eisen. Er is een deel ontwerp gemaakt vanuit het volledige ontwerp van de applicatie. Hierin is ook de gekozen remote communicatie technologie van 3.2.6 meegenomen. Met behulp van dit deel ontwerp is een testapplicatie geïmplementeerd, die beschikt over de functionaliteiten waarvan de performance eisen beschreven zijn in 3.2.3. Bij het uitvoeren van deze functionaliteiten zijn er tijdsmetingen uitgevoerd. Deze tijdsmetingen zijn vergeleken met de gestelde performance eisen in 3.2.3.

4 Resultaten

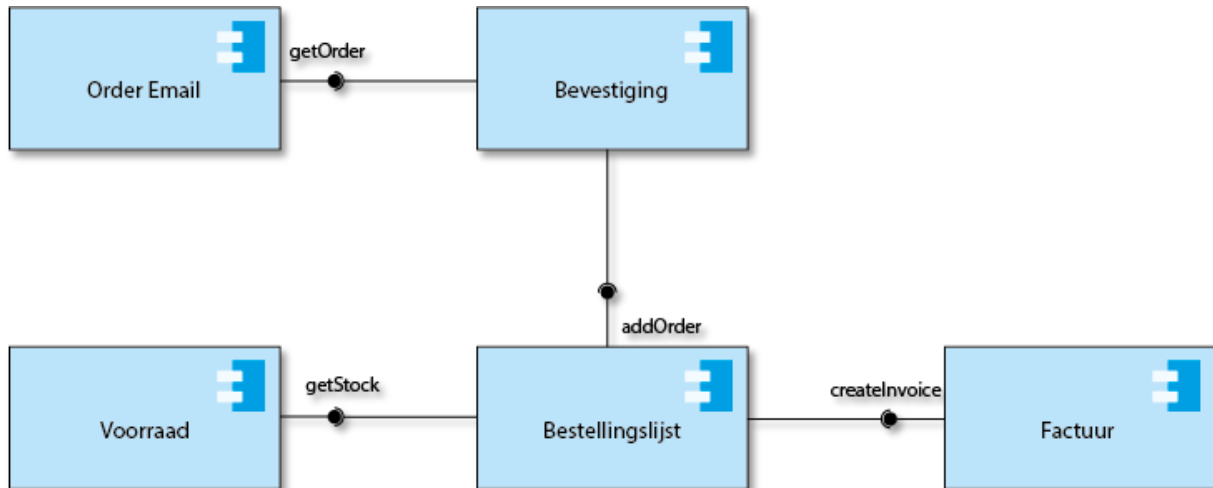
4.1 Hoe ziet het oude systeem eruit, zowel de architectuur als de gebruikte werkwijze?

Door het houden van het interview en het bekijken van het huidige systeem is er een duidelijk beeld ontstaan van het complete huidige systeem.

Het huidige systeem bestaat uit vier gedeeltes: Outlook, Excel, Parcelware en Administratie. Het Outlook gedeelte bestaat uit twee componenten Order Email en Bevestiging. Het Excel gedeelte bestaat uit drie componenten: Voorraad, Bestellinglijst en Factuur. Het Parcelware en Administratie gedeelte bestaan allebei uit één component. Deze componenten en hun interfaces zijn weergegeven in figuur 1 en figuur 2. In figuur 2 zijn alleen de componenten afgebeeld die binnen de scope vallen.



Figuur 1 Complete Componenten diagram

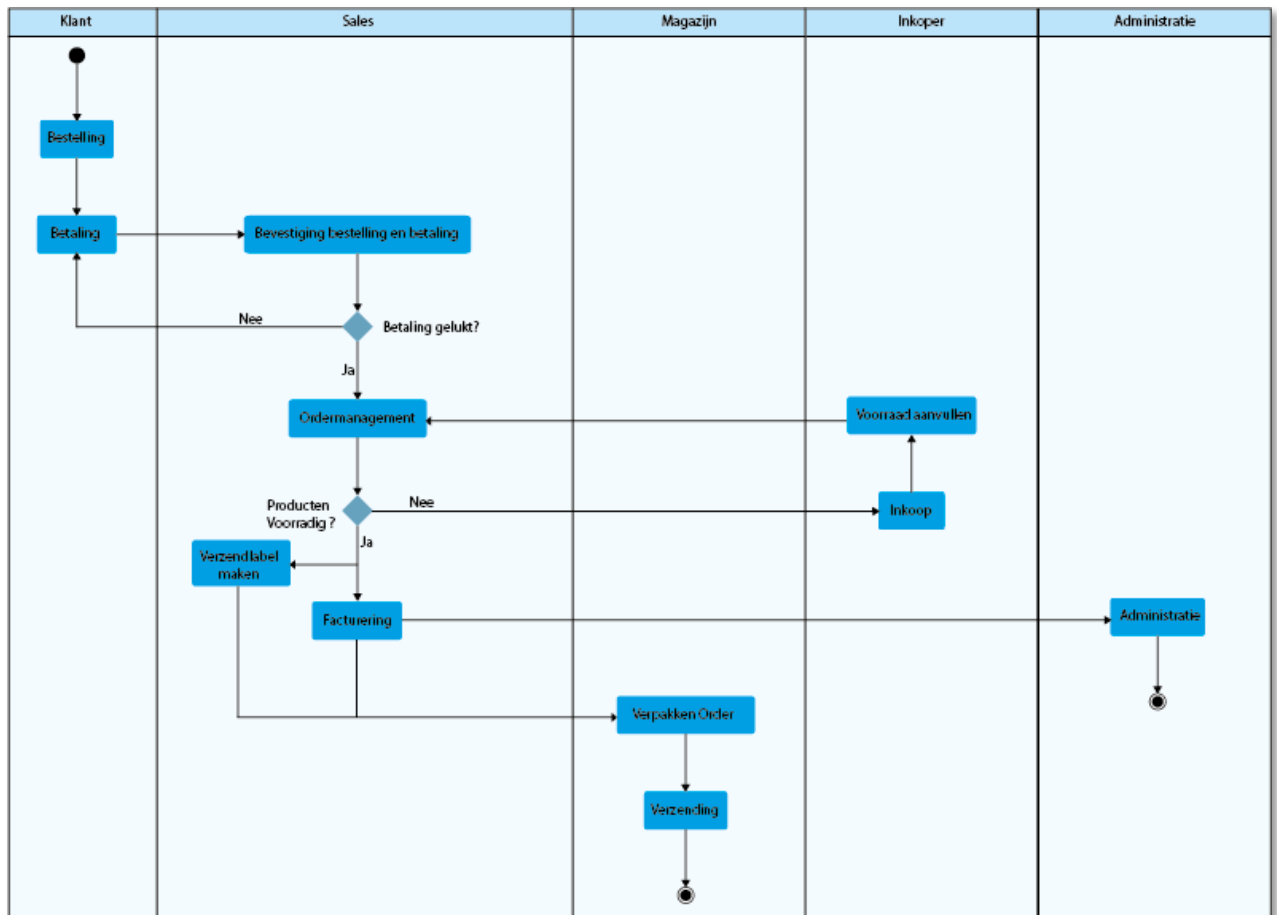


Figuur 2 Scope Componenten diagram

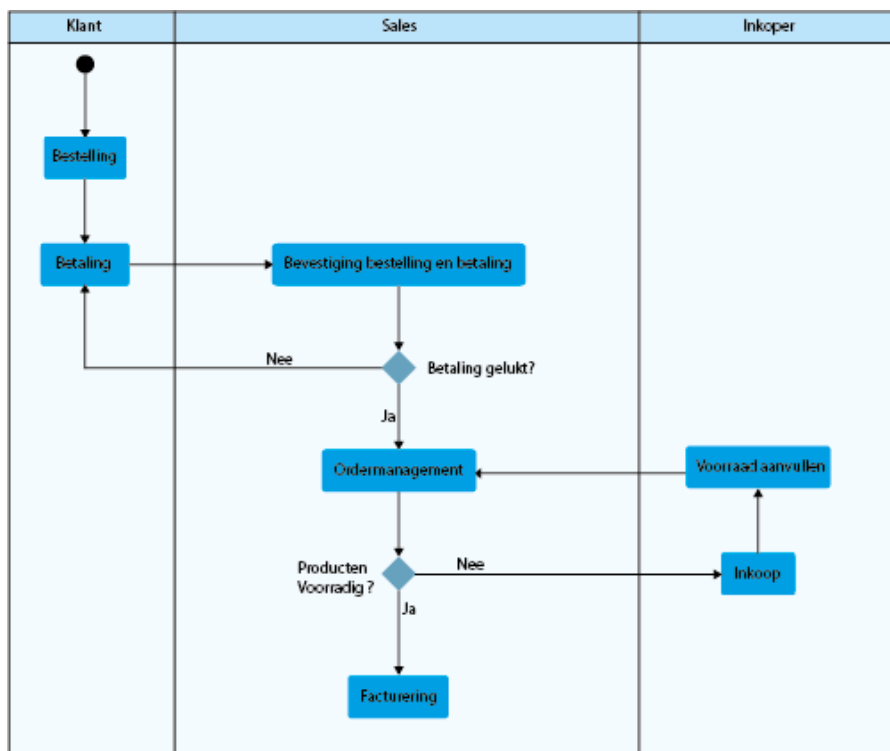
In het Outlook gedeelte komen de bestellingen die de klant op de website geplaatst heeft via e-mail binnen. Hier wordt een macro gestart die uit de email gegevens haalt en dit doorstuurt naar het Excel bestand. In het Excel bestand worden alle gegevens van de order en de daarbij behoorde klant gegevens bijgehouden. Per order wordt door middel van kleuren bijgehouden wat de status is. Vanuit een order wordt de voorraad gecontroleerd van de producten. Hierna wordt van de order een factuur gemaakt. Deze factuur zal later door de administratie verwerkt worden in de boekhouding. Vanuit de bestellinglijst wordt er een verzendlabel aangemaakt in Parcelware. Dit verzendlabel gaat samen met een factuur naar het magazijn. Hier worden de bestelde producten met factuur en verzendlabel verpakt. Vanuit de bestellinglijst wordt een e-mail naar de klant gestuurd met daarin de verzend gegevens. Zodat de klant weet wanneer hij het pakket kan verwachten.

Als een product niet voorradig is wordt er nog geen factuur gemaakt. De inkoper gaat ervoor zorgen dat het product weer voorradig is. De inkoper zorgt er ook voor dat de voorraadstatus op de website correct blijft in verhouding met de fysieke voorraad.

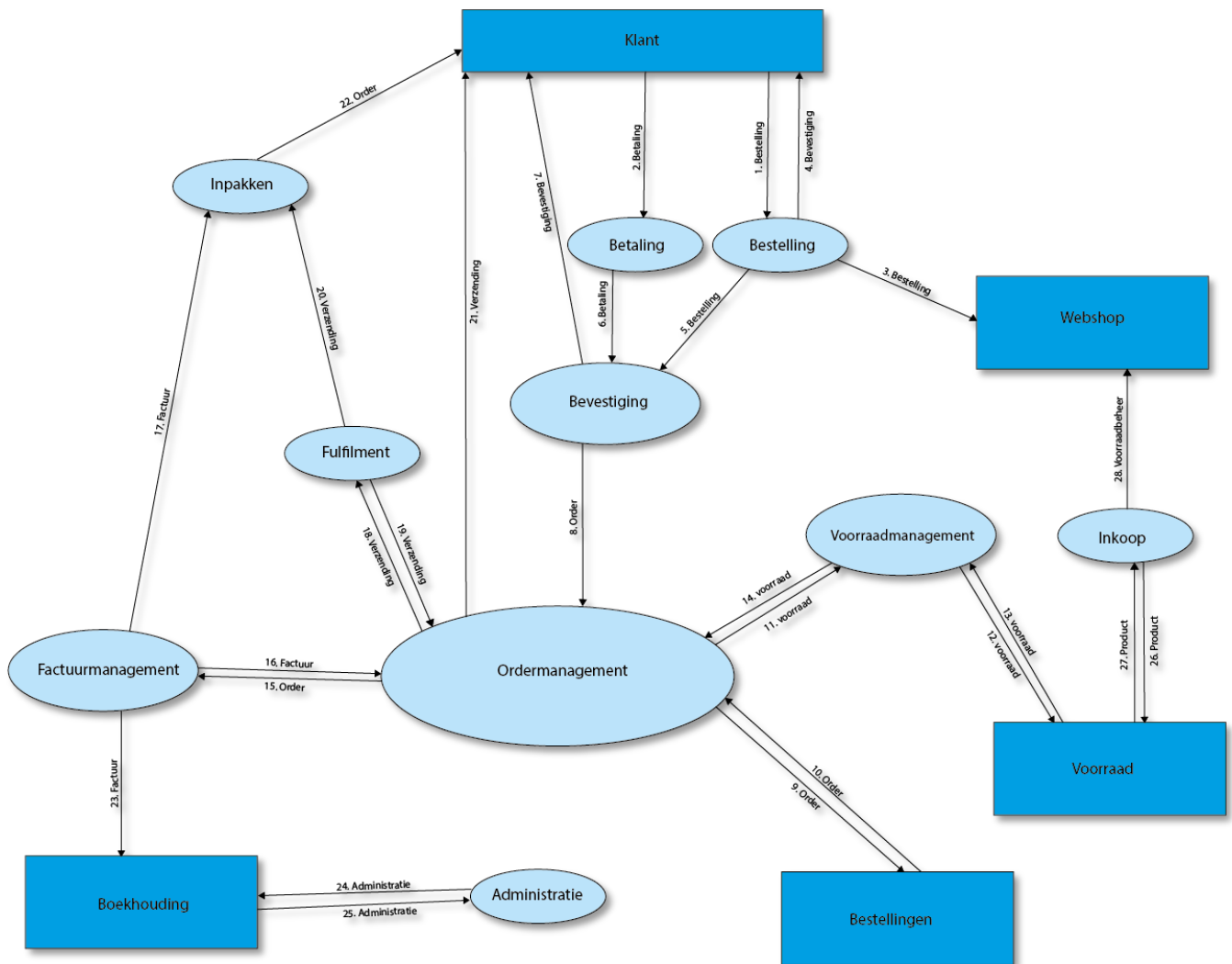
Ter illustratie zijn vier diagrammen toegevoegd. Twee Activiteitendiagrammen en twee Dataflow diagrammen. In de Activiteitendiagrammen zijn de handelingen die elke gebruiker uitvoert te zien samen met de beslissingsmomenten. Figuur 3 is het volledige Activiteitendiagram, figuur 4 bevat alleen de onderdelen die voor dit project binnen de scope vallen. In de Dataflow diagrammen zijn de gegevensstromen, de processen en de opslag zichtbaar. Figuur 5 is de complete weergave, figuur 6 is de weergave van alles wat binnen de scope valt.



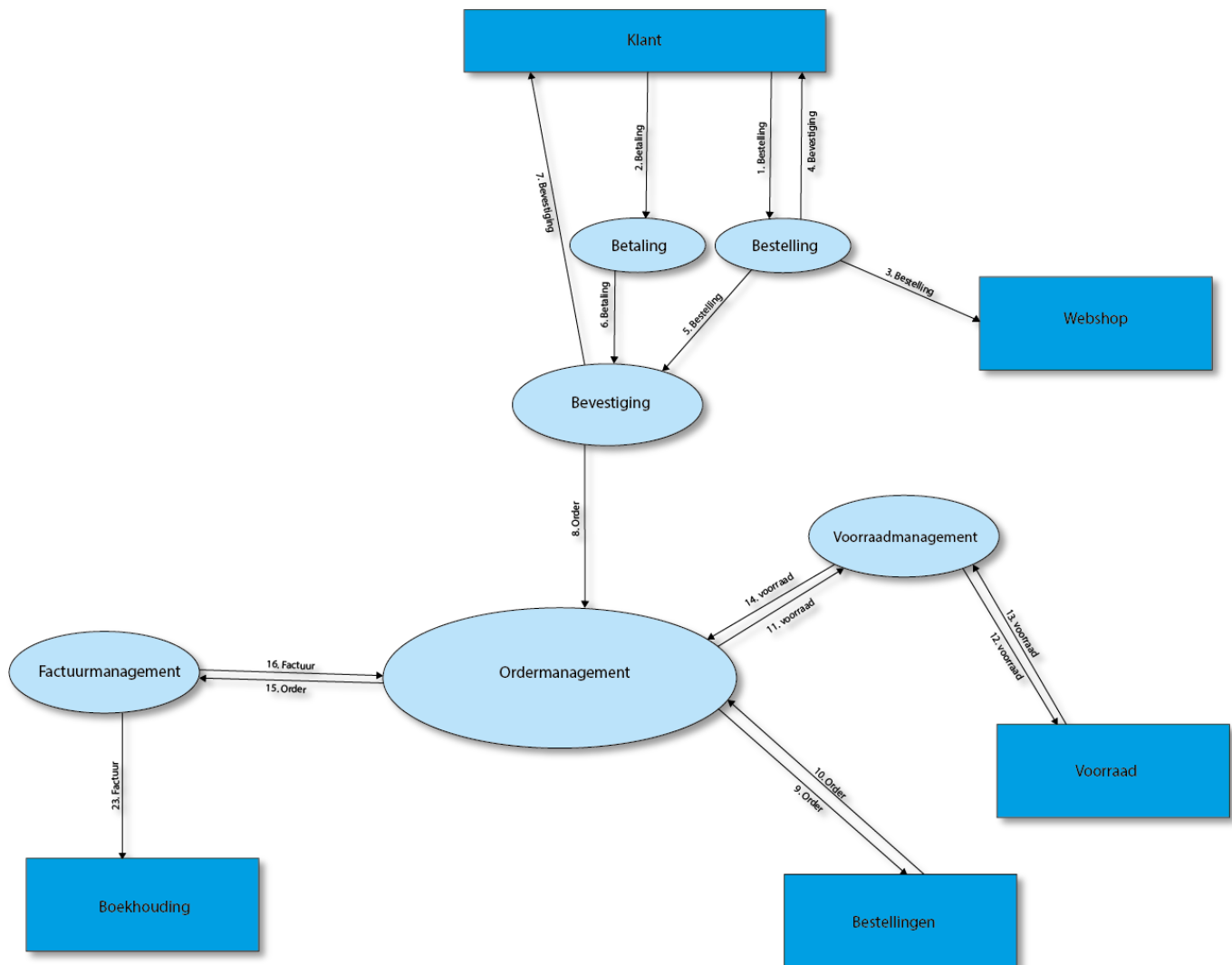
Figuur 3 Complete Activiteitendiagram



Figuur 4 Scope Activiteitendiagram



Figuur 5 Complete Dataflow diagram



Figuur 6 Scope Dataflow diagram

Verdere informatie is uitgewerkt in bijlage I op bladzijde 25.

4.2 Wat zijn de functionele systeemeisen voor het nieuwe systeem?

In combinatie met 4.1 zijn er 39 aantal functionele requirements opgenomen in de requirements lijst. Er zijn 24 requirements gemarkeerd met de prioriteit Must have's en 15 met Should have's (Brennan, 2009). Deze lijst in terug te vinden in Bijlage I.

4.3 Wat zijn de performance eisen voor het nieuwe systeem?

Door het houden van het interview zijn er drie performance requirements opgenomen in de requirements lijst. Deze lijst in terug te vinden in bijlage I. Alle performance requirements zijn gemarkeerd met de prioriteit Must have's (Brennan, 2009).

4.4 Wat zijn de gebruikersinterface eisen voor het nieuwe systeem?

Er zijn 13 gebruikersinterface requirements opgenomen in de requirements lijst als resultaat van dit onderzoek. Hiervan zijn er acht gemarkeerd als Must have's en vijf als Should have's (Brennan, 2009). De Requirements lijst is terug te vinden in bijlage I.

4.5 Welke bestaande oplossingen zijn er die aan de criteria en eisen voldoen?

Vanuit de ingevoerde zoekopdrachten zijn er vijf mogelijk te gebruiken pakketten geselecteerd.

Unit 4 (Unit4 Multivers , 2015)

Unit 4 Multivers is een standaard ERP-software pakket en beschikt onder ander over een order, voorraad en factuur gedeelte. Tevens is er een koppeling met Magento beschikbaar. Er zijn geen koppelingen met Joomla Virtuemart en Parcelware.

Scherpa (Sherpa, 2015)

Scherpa is een backoffice software pakket volledig gericht op webshops. Het beschikt onder andere over een order, voorraad en factuur gedeelte. Er is ook een koppeling met Magento en Parcelware beschikbaar. Er is geen koppeling met Joomla Virtuemart beschikbaar. Het softwarepakket Scherpa is niet cross-platform geschikt omdat de Clientside software op een Windows computer geïnstalleerd moet worden.

Microsoft Dynamics NAV (Microsoft , 2015)

Microsoft Dynamics NAV is een ERP software pakket ontwikkelt door Microsoft. Het beschikt over een Server en Clientside. Het is mogelijk om onder andere orders, voorraad en facturen te beheren. Voor de koppeling met Magento is er een koppeling beschikbaar van een 3^{de} partij. Er zijn geen koppelingen beschikbaar voor Joomla Virtuemart en Parcelware.

Embedded ERP (Embedded ERP, 2015)

Embedded ERP is een software pakket wat geïntegreerd wordt in Magento. Hierdoor is een extra koppeling met Magento niet nodig. Embedded ERP bevat onder andere de mogelijkheid om orders, voorraad en facturen te beheren. Er is geen koppeling met Joomla Virtuemart en Parcelware beschikbaar.

Suite commerce (Netsuite, 2015)

Suite commerce is een complete ERP Systeem van Netsuite. Er is een ordermanagement onderdeel dat zorgt voor de afhandeling van orders, facturen en voorraad. Er is een koppeling voor Magento beschikbaar maar geen koppelingen met Parcelware en Joomla Virtuemart.

In tabel 1 zijn de vijf pakketten en de twaalf hoofdcriteria te zien.

Pakketten / Hoofd Criteria (Max. Punten)	Unit 4	Scherpa	Microsoft Dynamics NAV	Embedded Erp	Suite commerce
13. Het pakket beschikt over een order management gedeelte.(50)	50	50	50	50	50
14. Het pakket beschikt over een factuur management gedeelte.(50)	50	50	50	50	50
15. Het pakket beschikt over een voorraad beheer gedeelte.(50)	50	50	50	50	50
16. Het pakket beschikt over een bestaande koppeling met Magento.(20)	20	20	20	20	20
17. Het pakket beschikt over een bestaande koppeling met Joomla Virtuemart.(20)	0	0	0	0	0
18. Het pakket werkt cross-platform.(40)	40	0	40	40	40
19. Het pakket is multi-user.(20)	20	20	20	20	20
20. Het pakket beschikt over een bestaande Parcelware integratie.(10)	0	10	0	0	0
21. Het pakket moet updates versturen naar klant.(10)	10	10	10	10	10
22. Het pakket moet status updates versturen naar Magento.(10)	10	10	10	10	10
23. Het pakket moet status updates versturen naar Joomla Virtuemart.(10)	0	0	0	0	0
24. Prijsindicatie zonder aanpassingen (eenmalig / per jaar).	3500 euro/jaar	7620 euro/jaar + 3100 eenmalig	1536 euro/jaar + 3500 euro eenmalig	1340 euro eenmalig	2000 euro/jaar
Prijs in punten(100)	5	0	35	100	20
Totaal punten (390)	255	220	285	350	270

Tabel 1 Bestaande applicaties

4.6 Welke gebruikersinterface technologieën moeten er gebruikt worden om aan de eisen te voldoen?

Tijdens dit onderzoek is er gekozen tussen JavaFX en Swing. De keuze is gemaakt door een pros and cons lijst op te stellen. Tevens is er een checklist ingevuld om te kijken of alle onderdelen in de technologieën zitten.

JavaFX is in vergelijking met Swing een veel nieuwere techniek (Weaver, Gao, Chin, Iverson, & Vos, 2014). In de toekomst komen er nog functionele uitbreidingen in JavaFX (JavaFX-Jira.kenai, 2015), Swing wordt niet meer uitgebreid. Doordat JavaFX de nieuwere techniek is zitten er nog veel bugs in. Wel staan veel van deze bugs in de planning om vóór de Java SE 9 release opgelost te worden (JavaFX-Jira.kenai, 2015). JavaFX beschikt over meer functionaliteit dan Swing. Een voordeel van JavaFX is de beschikking van Touch integratie. Swing is momenteel nog de meest gebruikte technologie. Voor beide technologieën is er nog volledige support. Voor Swing zal de support eerder stoppen dan voor JavaFX. De ontwikkelsnelheid van JavaFX zal iets lager liggen dan die voor Swing door de steilere leercurve (Stafford, J. (2013)).

JavaFX	Swing
Pro	Pro
Langere support termijn	Momenteel meest gebruikt
Uitgebreidere functionaliteit	Hogere ontwikkelsnelheid
Uitbreidingen in de toekomst	
Con	Con
Meer bugs	Korte support termijn
Lagere ontwikkelsnelheid	Minder functionaliteit
	Wordt niet meer uitgebreid

Tabel 2 pros and cons lijst JavaFX & Swing

Onderdeel	JavaFX	Swing
Buttons	X	X
Tabel met slider	X	X
Tickboxes	X	X
Pagination	X	-
Text-edit-single-line-field	X	X
Multiple-line-textfield	X	X
Single-line-textfield	X	X
Text-edit-multiple-line-field	X	X
Dropdown-selector	X	X
Radio-buttons	X	X
Datum-picker	X	-
Progress-indicator	X	X

Tabel 3 checklist JavaFX & Swing

4.7 Welke remote technologieën moeten er gebruikt worden om aan de eisen te voldoen?

Tijdens dit onderzoek is er gekozen tussen RMI SOAP webservices en REST webservices. RMI is een techniek speciaal voor Java applicaties, hierdoor is RMI taal afhankelijk, het kan alleen maar in combinatie met Java gebruikt worden. Webservices zijn taal onafhankelijk wat

het mogelijk maakt om twee applicaties die in een andere taal geschreven zijn met elkaar te laten communiceren. Webservices zijn dus beter geschikt als het gaat om Cross-platform (Englander, 2002). Met webservices is het zowel mogelijk om asynchrone communicatie als synchrone communicatie te hebben, met RMI is alleen synchrone communicatie mogelijk. Uit onderzoek blijkt dat RMI sneller is (Techannotation, 2012). Dit komt doordat het serialiseren en deserialiseren sneller gaat met de binaire data representaties van RMI dan met XML wat in SOAP Webservices wordt gebruikt (Harold, 2004). Bij REST webservices kan er geserialiseerd worden naar XML of JSON. Tussen REST en SOAP zitten niet veel verschillen behalve dat SOAP beter bekend is binnen de organisatie.

Onderdeel	SOAP Webservices	REST Webservices	RMI
Taal onafhankelijk	X	X	-
Synchrone communicatie	X	X	X
Asynchrone communicatie	X	X	-
Bekend binnen GoodTimeGifts	X	-	-

Tabel 4 Webservices & RMI

4.8 Zijn de performance eisen haalbaar?

Tijdens dit onderzoek zijn er twee testen uitgevoerd: test 1: Het laden van 100 orders in de ordertabel en test 2: Het zoeken in alle orders.

Test 1: Het laden van 100 orders in de ordertabel.

Stappen:

1. De client applicatie start de aanvraag
2. De tijdsmeting wordt gestart.
3. De client applicatie roept de Webservice methode van de server applicatie aan.
4. De server applicatie haalt 100 Random OrderOverview records uit de tabel.
5. De server applicatie maakt 100 OrderOverview objecten aan en zet deze in een ArrayList.
6. De server applicatie plaats de ArrayList in een ArrayListContainer object.
7. De server applicatie retourneert het ArrayListContainer object aan de client applicatie.
8. De client applicatie vult de GUI tabel.
9. De tijdsmeting wordt gestopt.

In tabel 5 zijn de metingen te zien in milliseconden

Meting 1	239
Meting 2	142
Meting 3	135
Meting 4	132
Meting 5	154
Meting 6	116
Meting 7	126
Meting 8	108
Meting 9	131
Meting 10	112

Meting 11	111
Meting 12	109
Meting 13	102
Meting 14	106
Meting 15	106
Meting 16	122
Meting 17	114
Meting 18	124
Meting 19	100
Meting 20	93
Meting 21	123
Meting 22	108
Meting 23	102
Meting 24	92
Meting 25	115
Meting 26	91
Meting 27	100
Meting 28	112
Meting 29	107
Meting 30	101

Tabel 5 Performance meting

De meting is in totaal 30 keer uitgevoerd. Met een gemiddelde tijd van 117 milliseconden. Uit de metingen is af te lezen dat de eerste aanvraag langer duurt (meting 1). Dit komt doordat er voor de eerste keer connectie met de Webservice gemaakt. Bij de metingen hierna wordt de connectie gebruikt. De test is meerdere keren herhaalt waarbij tussen elke meting de client applicatie werd afgesloten. Hierbij kwam een gemiddelde tijd van 275 milliseconden.

Test 2: Het zoeken in alle orders.

Stappen:

1. De client applicatie start de aanvraag
2. De tijdsmeting wordt gestart.
3. De client applicatie roept de Webservice methode van de server applicatie aan.
4. De server applicatie zoekt in de OrderOverview tabel naar de juiste records.
5. De server applicatie maakt een aantal OrderOverview objecten aan, maximaal 100, en zet deze in een ArrayList.
6. De server applicatie plaats de ArrayList in een ArrayListContainer object.
7. De server applicatie retourneert het ArrayListContainer object aan de client applicatie.
8. De client applicatie vult de GUI tabel.
9. De tijdsmeting wordt gestopt.

Als variabelen zijn het totaal aantal records in de tabel meegenomen en het aantal records die de zoek query retourneert.

	Totaal aantal records						
Aantal records die voldoen aan de zoek query		2500	5000	10000	15000	20000	25000
	10	120	110	115	130	131	135
	20	125	130	132	129	142	138
	40	134	133	142	147	150	149
	80	150	153	159	152	158	163
	160	162	159	159	168	165	168
	320	156	149	168	172	142	168
	640	145	157	165	166	149	156
	1280	152	163	149	181	176	165
	2560	x	145	159	165	162	178
	5120	x	x	143	136	146	153
	10240	x	x	x	132	156	149
	20480	x	x	x	x	x	146

Tabel 6 Database performance test

Elke waarde in de tabel is een gemiddelde van tien tests in milliseconden. Uit de waarde van de tabel is af te lezen dat het aantal records die voldoen aan de zoek query alleen verschillen als dit minder als 100 is. Het totaal aantal records in de database maakt geen verschil uit.

5 Conclusie

5.1 Hoe ziet het oude systeem eruit, zowel de architectuur als de gebruikte werkwijze?

Het huidige systeem bestaat uit vier gedeeltes: Outlook, Excel, Parcelware en Administratie. De gebruikte architectuur en werkwijze zijn afgebeeld in figuur 1 tot en met figuur 6.

5.2 Wat zijn de functionele systeemeisen voor het nieuwe systeem?

Er zijn 39 functionele requirements opgenomen. Deze lijst is terug te vinden in bijlage I.

5.3 Wat zijn de performance eisen voor het nieuwe systeem?

Er zijn drie performance requirements opgenomen. Deze lijst is terug te vinden in bijlage I.

5.4 Wat zijn de gebruikersinterface eisen voor het nieuwe systeem?

Er zijn dertien gebruikersinterface requirements opgenomen. Deze lijst is terug te vinden in bijlage I.

5.5 Welke bestaande oplossingen zijn er die aan de criteria en eisen voldoen?

Er is tijdens het onderzoek gekeken naar vijf mogelijk te gebruiken pakketten. Na onderzoek op basis van hoofd requirements en een gesprek met de opdrachtgever is er gekozen om niet een bestaande applicatie te gebruiken. De reden hier van is dat vier van de vijf pakketten boven het budget lagen, alleen Embedded ERP lag binnen het budget. Er is niet gekozen voor dit pakket omdat het geïntegreerd zit met Magento waardoor er in de toekomst niet meer af te stappen is van Magento.

5.6 Welke gebruikersinterface technologieën moeten er gebruikt worden om aan de eisen te voldoen?

Nadat JavaFX en Swing met elkaar zijn vergeleken. Is er gekozen om JavaFX te gebruiken voor de applicatie. Dit omdat JavaFX meer toekomst gericht is en alle gewenste onderdelen bevat. Een nadeel van JavaFX is dat het een groot aantal bugs heeft, dit zal geen probleem zijn doordat slechts de standaard functionaliteit gebruikt zal worden.

5.7 Welke remote technologieën moeten er gebruikt worden om aan de eisen te voldoen?

De keuze is gevallen op webservices SOAP omdat deze technologie beter cross-platform geschikt is dan RMI. Er is niet voor REST gekozen omdat deze minder bekend is binnen de organisatie. Uit onderzoek bleek webservices iets langzamer te zijn dan RMI.

5.8 Zijn de performance eisen haalbaar?

Uit de verschillende metingen is gebleken dat de performance met het ontwerp haalbaar zijn. De gemiddelde meting kwam uit op 117 milliseconden en de eis was drie seconden.

6 Referenties

Brennan, K. (2009). *A Guide to the Business Analysis Body of Knowledge. International Institute of Business Analysis*. Opgeroepen op 7 4, 2015

Embedded ERP 2015. *The ultimate ERP solution for Magento*. Opgeroepen op 8 4, 2015, van: <http://www.embeddederp.com/en/>

Englander, R. (2002). *Java and SOAP*. " O'Reilly Media, Inc.". Opgeroepen op 7 4, 2015

Harold, E. (2004). *Java network programming*. " O'Reilly Media, Inc.". Opgeroepen op 7 4, 2015

JavaFX-Jira.kenai (2015). *JavaFX Bug tracking and project tracking*. Opgeroepen op 14 4, 2015 Van: [https://javafx-jira.kenai.com/issues/?jql=issuetype%20in%20\(%22Enhancement%20request%22%2C%20Epic%2C%20Feature%2C%20Improvement\)](https://javafx-jira.kenai.com/issues/?jql=issuetype%20in%20(%22Enhancement%20request%22%2C%20Epic%2C%20Feature%2C%20Improvement))

JavaFX-Jira.kenai (2015). *JavaFX Bug tracking and project tracking*. Opgeroepen op 14 4, 2015 Van: <https://javafx-jira.kenai.com/issues/?jql=issuetype%20%3D%20Bug>

Microsoft (2015). *Microsoft Dynamics NAV*. Opgeroepen op 8 4, 2015, van: <http://www.microsoft.com/nl-nl/dynamics/erp-nav-overview.aspx>

Sherpa (2015). *Zorgeloos bestellingen verwerken met Sherpa Software*. Opgeroepen op 8 4, 2015, van: <http://www.sherpaan.nl/>

Stafford, J. (2013). *Pros, cons of moving from Swing to JavaFX: UI tools a plus*. Opgeroepen op 14 4, 2015, van: <http://www.theserverside.com/feature/Pros-cons-of-moving-from-Swing-to-JavaFX-UI-tools-a-plus>

Netsuite (2015). *Customer Commerce, Omnichannel — NetSuite SuiteCommerce*. Opgeroepen op 8 4, 2015, van: <http://www.netsuite.com/portal/products/netsuite/suitecommerce.shtml>

Unit4 Multivers (2015). *Unit4 Multivers: Boekhoudsoftware voor het MKB*. Opgeroepen op 8 4, 2015, van: <http://www.unit4multivers.nl/>

Techannotation (2012). *Rmi vs Web Service*. Opgeroepen op 7 4, 2015. Van: <https://techannotation.wordpress.com/2012/08/31/rmi-vs-web-service/>

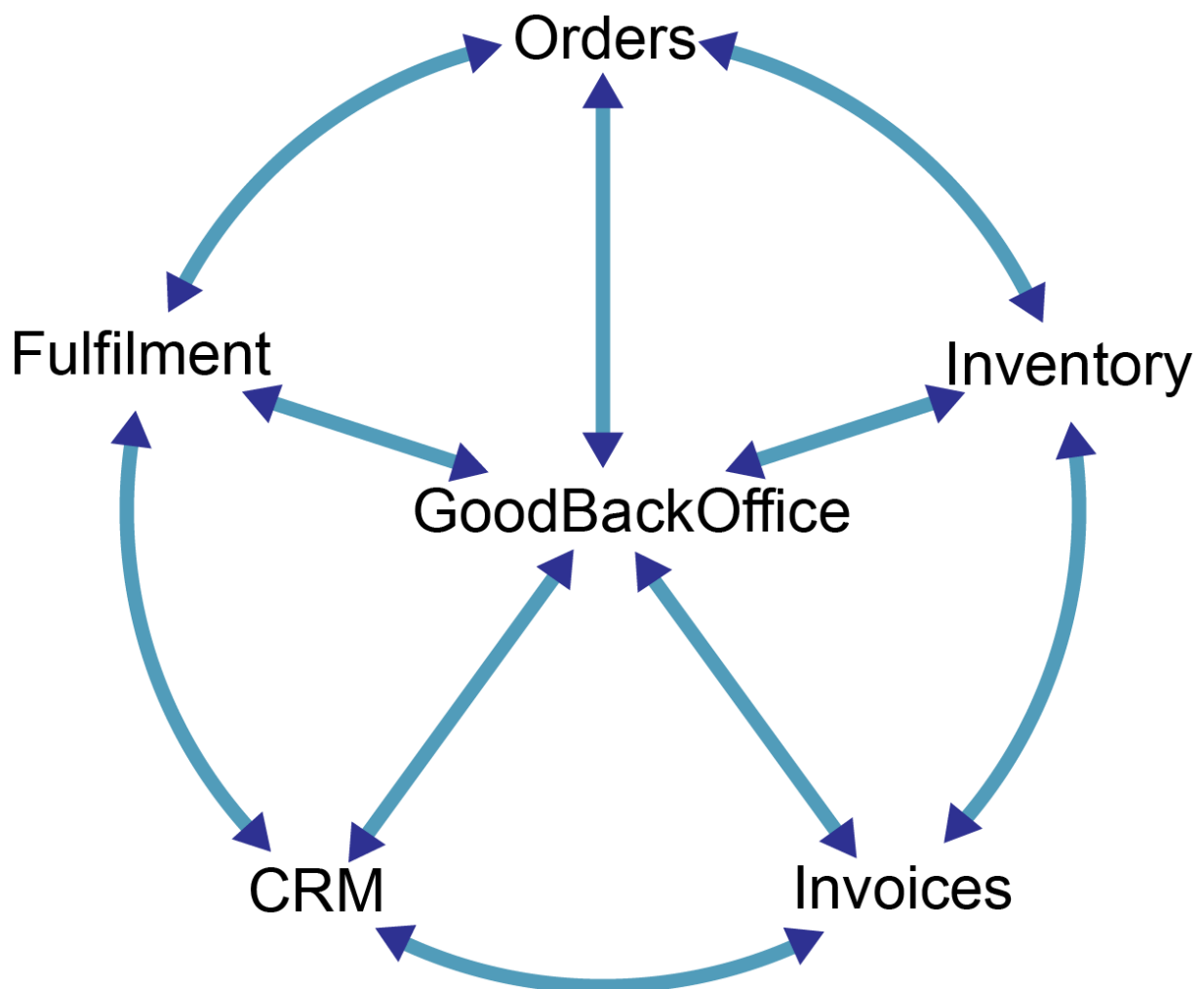
Weaver, J., Gao, W., Chin, S., Iverson, D., & Vos, J. (2014). *Pro JavaFX 8: A Definitive Guide to Building Desktop, Mobile, and Embedded Java Clients*. Apress. Opgeroepen op 7 4, 2015

Bijlage I

Requirements Lijst.

NR	Groepering	Requirements	MoSCoW
func-1	algemeen	Het systeem moet gebouwd worden in Java, zodat het ongeacht het operating system werkt.	Would
func-2	algemeen	Het systeem moet gebruik maken van de beschikbare MYSQL database zodat er geen database systeem aangeschaft hoeft te worden.	Would
func-3	algemeen	Gebruikers hoeven niet in te loggen in het systeem zodat men de applicatie snel kan gebruiken.	Would
func-4	algemeen	Het systeem moet gegevens loggen.	Would
func-5	multi-user	Het systeem moet voor meerdere gebruikers tegelijk toegankelijk zijn.	Should
func-6	multi-user	De gebruiker moet de gegevens in het systeem kunnen verversen zodat de gebruiker de gegevens van een andere gebruiker ziet.	Should
func-7	multi-user	Het systeem moet de gegevens na elke save verversen zodat de gebruiker niet handmatig hoeft te verversen.	Should
func-8	multi-user	Het systeem moet de gegevens die door een gebruiker gewijzigd worden locken voor andere gebruikers zodat men niet hetzelfde aan het aanpassen is.	Should
func-9	multi-user	Het systeem moet elke 5 minuten de gegevens automatisch verversen.	Should
func-10	order	Het systeem moet een overzicht kunnen tonen van de nieuwe orders.	Would
func-11	order	Het systeem moet een overzicht kunnen tonen van de open orders.	Would
func-12	order	Het systeem moet een overzicht kunnen tonen van de afgeronde orders.	Would
func-13	order	Het systeem moet een overzicht kunnen tonen van de afgewezen orders.	Would
func-14	order	Een nieuwe order moet bevestigd kunnen worden.	Would
func-15	order	Een nieuwe order moet afgewezen kunnen worden.	Would
func-16	order	Het systeem moet nieuwe orders kunnen importeren vanuit de webshops.	Would
func-17	order	De gebruiker moet een handmatige order in kunnen voeren zodat ook de bol.com en amazon.de orders in het systeem komen.	Should
func-18	order	Een order moet te dupliceren zijn.	Should
func-19	order	Een order moet te wijzigen zijn.	Would
func-20	order	In het overzicht van de open orders moet er gefilterd kunnen worden op: wel/niet bevestigd, een / geen factuur, welke website, wel / niet Productie product.	Should
func-21	order	Er moet op alle velden in de overzichten gezocht kunnen worden.	Would
func-22	order	Een afgewezen order moet alsnog geaccepteerd kunnen worden zodat bestellingen waar de betaling achteraf wel gelukt doorgevoerd kunnen worden.	Should
func-23	voorraad	Van een open order moet de voorraad gecontroleerd kunnen worden.	Would
func-24	voorraad	1 voorraadproduct kan meerdere verkoop producten hebben.	Would
func-25	voorraad	1 verkoopproduct kan geen voorraadproduct hebben.	Would
func-26	voorraad	Voorraad moet handmatig af- en bij geschreven kunnen worden.	Should
func-27	voorraad	Er moeten voorraadproducten toegevoegd kunnen worden.	Should

func-28	voorraad	Er moeten verkoopproducten toegevoegd kunnen worden.	Should
func-29	voorraad	Bij het maken van een factuur moet gecontroleerd worden of de producten voorradig zijn	Would
func-30	voorraad	Bij het maken van een factuur moeten de voorraadproducten automatisch afgeschreven worden.	Would
func-31	voorraad	Producten die na afschrijving op zijn, worden bij geschreven in een todo lijst.	Would
func-32	Factuur	Van een bevestigde order moet een factuur gemaakt kunnen worden.	Would
func-33	Factuur	Een factuur moet opgeslagen worden in PDF.	Would
func-34	Factuur	Een factuur moet opgeslagen worden in XML.	Should
func-35	Factuur	Een factuur moet geprint kunnen worden.	Would
func-36	Factuur	Van een factuur moet een pakbon gemaakt kunnen worden	Should
func-37	Factuur	Facturen kunnen niet aangepast te worden dit moet in de orderbewerking.	Would
func-38	Beheer	De volgende onderdelen moeten makkelijk geconfigureerd kunnen worden: Emailadressen verzender, IP adressen, inlog gegevens website databases.	Would
func-39	Beheer	De applicatie moet beschikken over een configuratie scherm	Should
Perf-1	Order	Bladeren tussen pagina's moet binnen 3 seconden klaar zijn.	Would
Perf-2	Zoeken	Zoeken in orders moet binnen 3 seconden klaar zijn.	Would
Perf-3	Order	Gegevens verversing van orderoverzicht moet binnen 3 seconden klaar zijn.	Would
GUI-1	Order	De orderoverzicht pagina bestaat uit meerdere pagina's met elk 100 regels.	Should
GUI-2	Zoeken	Zoeken gaat middels 1 zoekveld per kolom.	Would
GUI-3	Scherf	De applicatie moet beschikken over een overzicht orders scherm.	Would
GUI-4	Scherf	De applicatie moet beschikken over een bewerk/details order scherm.	Would
GUI-5	Scherf	De applicatie moet beschikken over een nieuwe order scherm.	Would
GUI-6	Scherf	De applicatie moet beschikken over een filter scherm.	Should
GUI-7	Scherf	De applicatie moet beschikken over een bekijk factuur scherm.	Would
GUI-8	Scherf	De applicatie moet beschikken over een overzicht voorraad scherm.	Would
GUI-9	Voorraad	Het overzicht voorraad scherm moet niet verdeeld zijn over meerdere pagina's.	Should
GUI-10	Scherf	De applicatie moet beschikken over een nieuw product scherm.	Should
GUI-11	Scherf	De applicatie moet beschikken over een bewerk product scherm.	Should
GUI-12	Scherf	Resolutie van schermen standaard op 1920 x 1030 pixels.	Would
GUI-13	Onderdelen	De applicatie moet beschikken over de volgende Gui-onderdelen: Knop, tabel met slider, tickboxes, Pagination, Text-edit-single-line-field, multiple-line-textfield, Single-line-textfield, Text-edit-multiple-line-field, dropdown-selector, radio-buttons, datum-picker, progress-indicator	Would

Bijlage III**GOODBACKOFFICE***ORDER EN VOORRAAD MANAGEMENT SYSTEEM***Testrapportage**

Project
Datum
Auteur
Versie
Status
Bedrijf

GoodBackOffice
25-8-2015
Ard Wijnen
0.1
Voltooid
GoodTimeGifts V.O.F.

Test 1: Wireframe acceptatie test

Applicatie	GoodBackOffice
Testrapportage gemaakt door	Ard Wijnen
Tester 1	Helene Wijnen
Tester 2	Peter Wijnen
Soort test	Wireframe Acceptatie test
Datum	06-05-2015
Passed / Failed	Passed
Bijbehorende documenten	Scriptie GoodBackOffice Ard Wijnen 2015

In test 1 zijn de gemaakte wireframes getest door middel van een acceptatietest. Het doel van de test is om te kijken of de gemaakte wireframes voldoen aan de eisen van de opdrachtgever en eindgebruiker. De test is geslaagd als de wireframes door beide testers geaccepteerd worden.

Werkwijze

De test is als volgt uit gevoerd: De ontwikkelaars heeft Tester 2 en Tester 3 de gemaakte wireframes uitgebreid laten zien. Hierbij werd duidelijk de werkwijze van de schermen getoond. Ook zijn de verschillen tussen het oude en nieuwe systeem naar voren gehaald. Na de demonstratie konden de testers vragen stellen en eventuele opmerkingen geven over de wireframes.

Resultaat

Beide testers hebben de wireframes geaccepteerd. De test hierdoor is gemarkeerd als passed.

Aanbeveling

Na het slagen van test 1 is de aanbeveling om door te gaan met de volgende fase.

Test 2: Unit test

Applicatie	GoodBackOffice
Testrapportage gemaakt door	Ard Wijnen
Tester 1	Ard Wijnen
Soort test	Unit test
Datum	17-8-2015
Passed / Failed	Passed
Bijbehorende documenten	Scriptie GoodBackOffice Ard Wijnen 2015

In test 2 zijn in totaal 24 unittest uitgevoerd. Het doel van deze test is om te controleren of er geen fouten in de code zitten. De test is geslaagd wanneer alle unittest geslaagd zijn.

Werkwijze

Tijdens de ontwikkeling van de applicatie zijn er door de ontwikkelaars verschillende unittests geschreven. Deze testen een losstaand gedeelte van de code. De unittests zijn geschreven voor de stukken code waar er gerekend wordt of waar er tussen formats gewisseld wordt. De testen zijn uitgevoerd door Tester 1.

Resultaat

Alle 24 unittests zijn geslaagd en gemarkeerd als passed.

Aanbeveling

Aanbevolen wordt om in de toekomst deze 24 unittest te blijven uitvoeren en als er verder ontwikkeld wordt aan de applicatie ook extra unittests toe te voegen.

Test 3: Applicatie acceptatie test

Applicatie	GoodBackOffice
Testrapportage gemaakt door	Ard Wijnen
Tester 1	Helene Wijnen
Tester 2	Peter Wijnen
Soort test	Applicatie Acceptatie test
Datum	25-8-2015
Passed / Failed	Passed
Bijbehorende documenten	Scriptie GoodBackOffice Ard Wijnen 2015

Tijdens test 3 is de volledige applicatie een acceptatietest doorgaan. Het doel van deze test is om te kijken of alle onderdelen van de applicatie aan de eisen en wensen van de opdrachtgever en eindgebruiker voldoen. De test is geslaagd wanneer de applicatie door beide testers geaccepteerd wordt.

Werkwijze

De ontwikkelaar heeft Tester 2 en Tester 3 de gehele applicatie laten zien, door middel van een demonstratie. Na de demonstratie konden de beide testers zelf de applicatie uitproberen, dit kon gelijktijdig zodat ook het multi user gedeelte getest werd.

Resultaat

Er zijn tijdens de test enkele nieuwe requirements en aanpassingen naar voren gekomen die meegenomen zullen worden in de ontwikkeling. De applicatie is geaccepteerd en de test gemarkeerd als passed.

Aanbeveling

Aanbevolen wordt om de aanpassingen direct mee te nemen in de volgende sprint. Het is verstandig om na een paar sprints weer een volledige acceptatie test uit te voeren.