

An iPad is shown from a front-facing perspective, centered against a white background. The screen is black and displays white text. The text is centered and consists of two lines: 'Afstudeerscriptie' on the top line and 'Zorg op de iPad' on the bottom line. The bottom line is in a larger, bolder font. The iPad's silver bezel and the circular home button at the bottom center are visible.

Afstudeerscriptie **Zorg op de iPad**

Afstudeerder: Bart Rabeling
Studentnummer: 2125740

AFSTUDEERVERSLAG VOOR FONTYS HOGESCHOOL ICT

GEGEVENS STUDENT

Naam:	B.T. Rabeling
Studentnummer:	2125740
Afstudeerrichting:	HBO ICT Software Engineering (Veltijd)
Afstudeerperiode	30-01-2012 t/m 30-06-2012 (100 werkdagen)

GEGEVENS BEDRIJF

Naam bedrijf:	Info Support
Afdeling:	Businessunit zorg
Plaats:	Veenendaal
Bedrijfsbegeleider:	Ivo Diepstraten

GEGEVENS DOCENTBEGELEIDER

Naam:	Marc Krielen
-------	--------------

GEGEVENS VERSLAG

Titel afstudeerverslag:	Afstudeerscriptie: Zorg op de iPad
Datum uitgifte afstudeerverslag:	07-06-2012

Getekend voor gezien door bedrijfsbegeleider:

Datum:

Handtekening bedrijfsbegeleider:

VOORWOORD

Dit verslag is geschreven in het kader van mijn afstudeeropdracht voor de opleiding HBO ICT Software Engineering aan de Hogeschool Fontys in Eindhoven. Het afstuderen vond plaats van 1 februari 2012 t/m 30 juni 2012 binnen het bedrijf Info Support in Veenendaal.

Het doel van het afstuderen is bewijzen dat ik voldoende kennis en ervaring bezit om mijn diploma voor de opleiding HBO ICT Software Engineering te mogen ontvangen. Verder heb ik als persoonlijke doelstellingen om meer technische kennis op te doen en mijn ondernemende vaardigheden verder te ontwikkelen.

Binnen Info Support heb ik zelfstandig een proof of concept gebouwd van een web-applicatie, geoptimaliseerd voor mobiele platformen. Deze web-applicatie is een revisie van de binnen Info Support ontwikkelde applicatie ZorgInfo.

Ik zou graag mijn opdrachtgever Ivo Diepstraten, mijn technisch begeleider Marco Pil, mijn procesbegeleider Pascalle Hijn, mijn onofficiële begeleider Rik Meijer en alle mede-afstudeerders willen bedanken voor het mede mogelijk maken van een leerzaam en plezierig afstuderen.

INHOUD

Samenvatting	4
Summary.....	5
Verklarende Woordenlijst.....	6
1. Inleiding	7
2. Het bedrijf.....	8
3. De Opdracht.....	9
Aanleiding	9
Probleemstelling.....	9
Doelstelling	9
Resultaat	9
4. Inceptiefase.....	10
4.1 Het plan van aanpak	10
4.2 Het onderzoek.....	11
4.3 Training: Developing web applications with ASP.NET MVC.....	13
4.4 Training: Developing web applications using jQuery and HTML 5	13
4.5 Het Functioneel en het technisch ontwerp	14
5. Elaboratiefase	15
5.1 Verbeterpunt: communicatie.....	16
5.2 De ontwikkelstraat Endeavour	17
5.3 De algemene gebruikersinterface	18
5.4 SOS: Save our software-project.....	19
5.5 Twitter Bootstrap	20
5.6 EmbeddedJS	21
6. Constructiefase.....	22
7. Transitiefase.....	23
8. Conclusies en Aanbevelingen	24
Conclusies.....	24
Aanbevelingen.....	25
Evaluatie	26
Literatuurlijst.....	28
Bijlage A: Plan van Aanpak	
Bijlage B: Onderzoeksdocument	
Bijlage C: Functioneel Ontwerp	
Bijlage D: Technisch Ontwerp	

SAMENVATTING

Het afstuderen heeft plaatsvonden binnen het bedrijf Info Support, een IT-dienstverlener, oorspronkelijk uit Veenendaal, die bedrijven ondersteunt bij het behalen van hun doelstelling met behulp van solide en innovatieve oplossingen.

Binnen de Algemene Wet Bijzondere Ziektekosten (AWBZ) worden persoonsgebonden hulpmiddelen vergoed. Omdat deze hulpmiddelen vaak persoongebonden zijn, en dus niet herbruikbaar zijn voor andere individuen, dient hier verantwoording voor afgedragen te worden. Om de aanvraag, met daarin de betreffende verantwoording, en eventuele goedkeuring hiervoor te managen heeft Info Support de ZorgInfo applicatie ontwikkeld.

Op dit moment moeten zorgverleners eerst notities maken bij het onderzoeken van een cliënt, om deze vervolgens op een later tijdstip in ZorgInfo in te voeren in de vorm van een aanvraag. Dit vergt veel tijd. Daarom is de afstudeeropdracht om een mobiele variant, ten eerste gebouwd voor de iPad, van de ZorgInfo applicatie te bouwen. Deze moet in ieder geval de functionaliteiten van het maken en versturen van een aanvraag op zich kunnen nemen. Het eindproduct is een proof of concept om aan te tonen hoe uitbreidbaar ZorgInfo is en welke aspecten er komen kijken bij het maken van een applicatie voor mobiele platformen.

Als ontwikkelingsmethodiek heb ik gekozen voor de RUP-methodiek met elementen van de SCRUM-methodiek, zoals de product backlog. Dit omdat deze methodiek de standaard is binnen Info Support en het ten hoogste aangeraden wordt.

Voor er begonnen werd aan het bouwen van de applicatie werd onderzoek gedaan naar de keuze om een native iPad applicatie óf een web-applicatie te bouwen voor de front-end. Uiteindelijk werd geconcludeerd dat een web-applicatie de beste keuze zou zijn voor dit project. De hoofdredenen voor deze keuze waren dat op deze manier updates direct doorgevoerd konden worden en de applicatie direct op meerdere platformen zou werken.

Het project voorliep voorspoedig. Tijdens de loop van het project waren er slechts enkele complicaties die uiteindelijk eindigden in grote vooruitgang binnen mijn persoonlijke ontwikkeling. De belangrijkste vooruitgang in ontwikkeling was mijn verbetering in communicatie tegenover de betrokkenen binnen het project. Dit gebeurde na aanleiding van een opmerking van mijn opdrachtgever dat betrokken partijen niet voldoende op de hoogte gehouden werden.

Als conclusie, betreffende de vragen van de opdrachtgever, kan gesteld worden dat de uitbreidbaarheid van ZorgInfo redelijk beperkt is. De reden hiervoor is voornamelijk dat componenten van ZorgInfo te diverse functionaliteiten op zich nemen. Om deze reden is mijn aanbeveling dan ook om componenten binnen software maar één taak of verantwoordelijkheid samen met een duidelijke interface voor andere componenten te geven.

SUMMARY

The graduation internship took place within the company Info Support. Originally from Veenendaal, Info Support is an IT-services company which supports other companies in achieving their business objectives through solid and innovative solutions.

Individual medical aids are being reimbursed by means of the Dutch law “Algemene Wet Bijzondere Ziektekosten” (AWBZ). Since these medical aids are often personally customized, and therefore not reusable for other individuals, these need to be accounted for. To manage the inquiries, containing the concerning justification, and the possible approval, Info Support developed the “ZorgInfo” application.

Currently, caregivers first have to make notes during the examination of a patient. They then have to later on fill in this data, as an inquiry, within the ZorgInfo application. This takes up a lot of time. Therefore, my assignment within the graduation internship is to build a mobile version of the ZorgInfo application, which at first only needs to be able to run on an iPad. The application should at least enable the user to create and send an inquiry. The final product is to be a proof of concept, which is to demonstrate the extensibility of ZorgInfo and to see which aspects come in play while developing an application for mobile platforms.

As the software development process framework I chose to use RUP in combination with specific elements of SCRUM, such as the product backlog. The reason for this decision is that this combination of frameworks is standardized within Info Support and therefore is highly recommended.

Before building the application, research was done in order to decide if the application ought to be a native iPad application or a web-application. Eventually the research concluded that building a web-application would be the best option for this specific project. The main advantages for this decision were immediately implemented updates and immediate multi-platform support.

The course of the project went smoothly. The few complications that arose during the course of the project ultimately ending up in significant progress within my personal development. The most important advance within my personal development was the improvement in the way that I communicate with the stakeholders of the project. This was due to a remark from my client regarding stakeholders not being sufficiently informed of the progress within the project.

Regarding the initial unknowns of the client, there can be concluded that the extensibility of the ZorgInfo application is fairly limited. The reason for this is mainly that components within ZorgInfo undertake various functionalities, which are not directly related to each other. For this reason, it is my recommendation that software-components should only be given one task or responsibility while also containing a clear interface for other components to communicate with.

VERKLARENDE WOORDENLIJST

ASP.NET

Een door Microsoft geproduceerde verzameling van tools die web-ontwikkelaars ondersteunen in het creëren van dynamische websites en web-applicaties.

AWBZ

Een verzekering, waarover alle Nederlanders automatisch beschikken, die medische kosten dekt die niet onder de zorgverzekering vallen en die door bijna niemand op te brengen zijn.

EJS

Een in Javascript geschreven templating engine. Met behulp van deze framework kan de gebruikersinterface dynamisch gegenereerd worden met behulp van een apart gedefinieerde template.

Info Support

Een IT-dienstverlener die sinds 1986 bedrijven ondersteunt in het behalen van hun doelstellingen met behulp van solide en innovatieve oplossingen.

jQuery

jQuery is een Javascript library die web-ontwikkelaars ondersteunt door snelle en makkelijke uitlezing en manipulatie van het HTML-document te ondersteunen. jQuery assisteert ook door inconsistenties tussen verschillende webbrowsers recht te trekken.

(ASP.NET) MVC 3

ASP.NET MVC 3 is een framework voor het bouwen van schaalbare, op standaard gebaseerde web-applicaties door middel van gevestigde ontwerp patronen en de kracht van ASP.NET en het .NET framework.

Twitter Bootstrap

Twitter Bootstrap is een uitgebreide verzameling van tools die ondersteunt in de ontwikkeling van schaalbare en grafisch aantrekkelijke gebruikersinterfaces van web-applicaties. Twitter Bootstrap bestaat uit een combinatie van CSS en Javascript-code.

ZorgInfo

Een applicatie, ontwikkeld door Info Support, die zorgkantoren ondersteunt in hun taak om het totale aanvraag- machtiging- en declaratietraject voor persoonsgebonden hulpmiddelen te verzorgen.

1. INLEIDING

Deze afstudeerscriptie is geschreven ter afsluiting van de opleiding HBO ICT Software Engineering aan de Hogeschool Fontys in Eindhoven. De afstudeerstage is uitgevoerd in opdracht van het bedrijf Info Support in Veenendaal, een IT-dienstverlener die sinds de oprichting in 1986 bedrijven ondersteunt in het behalen van hun doelstellingen met behulp van solide en innovatieve oplossingen.

De beginsituatie is als volgt: Info Support heeft voor het verantwoorden van persoonsgebonden hulpmiddelen binnen de Algemene Wet Bijzondere Ziektekosten (AWBZ), tegenover zorgkantoren, de ZorgInfo applicatie ontwikkeld. Zorgverleners moeten momenteel eerst een cliënt onderzoeken, notities maken en vervolgens achteraf de gegevens invullen in ZorgInfo. Door deze tussenstap gaat er veel kostbare tijd verloren.

De opdracht binnen het afstuderen is om een proof of concept te bouwen van een nieuwe versie van ZorgInfo. Deze versie moet geoptimaliseerd zijn voor mobiele platformen, beginnende met de iPad, en moet in ieder geval als functionaliteit een aanvraag kunnen opmaken en versturen. Het doel van deze opdracht is naast het oplossen van het genoemde probleem, om zorgverleners veel tijd te besparen in het opmaken en versturen van aanvragen, om de uitbreidbaarheid van ZorgInfo te testen en de mogelijkheden van optimalisatie voor mobiele platformen te verkennen. Voor deze opdracht mag gebruik worden gemaakt van bestaande achterliggende frameworks van de huidige ZorgInfo applicatie.

Het afstuderen begint met onderzoek naar de oorspronkelijke applicatie. Vervolgens wordt er een onderzoek gedaan om te bepalen of het beter is om een native iPad applicatie te bouwen, of een web-applicatie. Aan de hand van het resultaat van het onderzoek zal de oplossing daarna gebouwd worden met als ontwikkelingsmethodiek de RUP-methodiek in combinatie met elementen van de SCRUM-methodiek, zoals een product backlog.

De opbouw van dit document is als volgt: In hoofdstuk 2 wordt meer informatie gegeven aangaande het bedrijf Info Support. In hoofdstuk 3 wordt de afstudeeropdracht uitgebreid beschreven. In de hoofdstuk 4 wordt de inceptiefase beschreven, binnen deze fase vindt de planning en voorbereiding, voor de opdracht, plaats samen met het onderzoek betreffende de keuze om een native- of web-applicatie te bouwen. In de hoofdstukken 5 en 6 worden de elaboratie- en constructiefase beschreven, in deze fasen vindt de daadwerkelijke bouw van het eindproduct plaats. In hoofdstuk 7 wordt de transitiefase beschreven waarin het eindproduct wordt opgeleverd. Ten slotte worden in hoofdstuk 8 de conclusies en aanbevelingen doorgenomen.

2. HET BEDRIJF

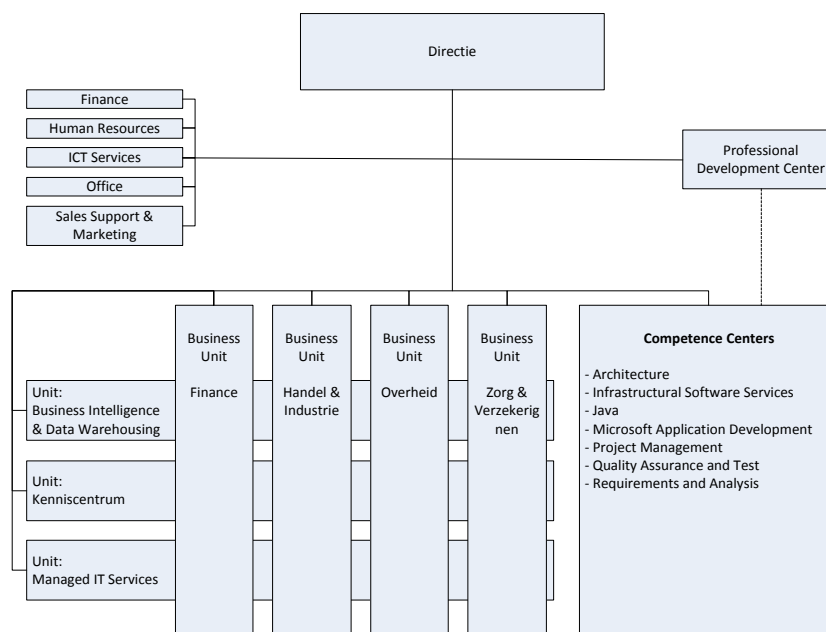
Info Support is sinds de oprichting in 1986 een IT-dienstverlener, met ruim 350 medewerkers, die zorgt voor innovatieve en solide softwareoplossingen die organisaties ondersteunen in het realiseren van hun bedrijfsdoelstellingen. De technische middelen die Info Support gekozen heeft om deze taak te volbrengen zijn Microsoft (.NET, C#, SQL Server), Oracle (Fusion Middleware, Weblogic, Suite) en Java.

Naast het leveren van allerlei softwareoplossingen geeft Info Support vanuit het kenniscentrum in Veenendaal meer dan 250 trainingen aan IT-Professionals. Naast de genoemde 250 trainingen die beschikbaar zijn kan Info Support op verzoek ook op maat gemaakte trainingen verzorgen.

De werkwijze binnen Info Support wordt gekenmerkt door een aantal kernwaarden: Soliditeit, integriteit, vakmanschap en passie. Dit zorgt voor degelijke betrouwbare producten die met vakmanschap en passie gerealiseerd worden. Daarnaast betekent dit een open, eerlijke en betrouwbare relatie met de klant.

Om met solide en innovatieve oplossingen te blijven komen zijn intensieve partnershiprelaties met belangrijke spelers in de IT-industrie noodzakelijk. Zo hebben ze binnen Info Support snel kennis van de nieuwste technologische ontwikkelingen. Deze kunnen dan weer toegepast worden in hun producten en diensten. Partners zijn onder andere: Microsoft, Oracle, Adobe, IBM en Sparx Systems.

Info Support kent een platte procesgerichte organisatie zoals in Figuur 1 als organogram is weergegeven. De werkzaamheden worden, afhankelijk van de markt waarin de werkzaamheden plaatsvinden, ingedeeld in businessunits.



Figuur 1: Organogram van de organisatie binnen Info Support

3. DE OPDRACHT

AANLEIDING

Binnen de Algemene Wet Bijzondere Ziektekosten (AWBZ) worden persoonsgebonden hulpmiddelen vergoed. Deze hulpmiddelen zijn vaak op maat gemaakt voor een individu en is daardoor ook niet bruikbaar voor anderen. Omdat de AWBZ betaald wordt met algemene middelen is het van belang dat het inzetten van deze persoonsgebonden hulpmiddelen wordt verantwoord.

Om de communicatie van verantwoording en goedkeuring tussen respectievelijk de zorgverlener en het zorgkantoor makkelijker te maken is de web-applicatie ZorgInfo ontwikkeld. Echter is deze applicatie geoptimaliseerd voor gebruik met Microsoft Internet Explorer terwijl de cliënten over het algemeen niet naast een computer geïndiceerd worden.

PROBLEEMSTELLING

Omdat zorgverleners op dit moment notities moeten maken bij het behandelen van cliënten, om deze op een later tijdstip op een computer in te voeren zijn ze een hoop kostbare tijd kwijt die voor betere doeleinden gebruikt zou kunnen worden.

DOELSTELLING

Er zijn 2 doelstellingen:

1. **Afstuderen:** Met dit project probeer ik af te studeren voor de opleiding HBO ICT & Software Engineering aan de Fontys Hogescholen te Eindhoven om zo mijn diploma te kunnen halen. Dit doe ik
2. **Info Support:** De opdrachtgever verwacht van dit project een proof of concept van een mobiele applicatie die als portal dient voor zorgverleners om een indicatie in te kunnen voeren en op te sturen. Uit deze proof of concept wil de opdrachtgever kennis vergaren over de uitbreidbaarheid van de oorspronkelijke ZorgInfo applicatie en de manier waarop een web-applicatie geoptimaliseerd kan worden voor mobiele platformen. Deze kennis wordt vergaard door middel van de te bouwen applicatie, geoptimaliseerde voor mobiele platformen, die voortbordurt op bestaande onderdelen van de huidige ZorgInfo applicatie.

RESULTAAT

1. Het feit dat ik succesvol kan afstuderen.
2. Een proof of concept van een applicatie, die geoptimaliseerd is voor mobiele platformen als de iPad, waarin hoofd-functionaliteiten van de oorspronkelijke ZorgInfo applicatie geïmplementeerd zijn.
3. Nieuwe kennis voor Info Support over de uitbreidbaarheid van de originele ZorgInfo applicatie en over hoe een web-applicatie mogelijk geoptimaliseerd kan worden voor mobiele platformen.

4. INCEPTIEFASE

Voordat er daadwerkelijk aan het eindproduct gebouwd gaat worden, worden in deze fase de voorbereidende werkzaamheden uitgevoerd. Er wordt een plan van aanpak gemaakt, om te bepalen hoe de voortgang gedurende het gehele project zal verlopen. Ook worden er een ontwikkelingsmethodiek gekozen, onderzoek gedaan en requirements opgesteld.

De officiële start van deze fase werd direct vertraagd. Dit gebeurde dankzij een onverwachte onvoldoende beoordeling vanuit school die vlak voor de officiële start van het afstuderen nog gegeven werd. De opgelegde reparatie, die ervoor zou zorgen dat ik alsnog een voldoende zou krijgen, kostte een week om uit te voeren. De reparatie was een succes, waarna ik alsnog mocht beginnen met mijn afstuderen.

De inceptiefase begon met het maken van een eerste versie van het plan van aanpak. Deze eerste versie werd door mijn docent begeleider vanuit school direct goedgekeurd. Mijn technisch begeleider en opdrachtgever vonden deze echter inhoudelijk niet specifiek genoeg. Achteraf zou blijken dat dankzij een miscommunicatie tussen mij en mijn technisch begeleider het nog 2 weken zou duren voor er een algehele goedkeuring zou zijn. De volledige beschrijving van hoe het plan van aanpak uiteindelijk tot stand is gekomen is na te lezen in de paragraaf “4.1 Het plan van aanpak”.

In week 5 begon ik met het onderzoek. Dit ging bepalen of het beter zou zijn voor het project om het eindproduct te bouwen in de vorm van een native iPad-app of een web-applicatie. Al gauw bleek dat de voorkeur voor één van de twee vormen sterk afhankelijk is van het betreffende project. Het werd de vraag van welke vorm mijn afstudeerproject de meeste baat zou hebben. Uiteindelijk wees het onderzoek uit dat er, vanuit mijn project, duidelijke redenen waren om te kiezen voor de vorm van een web-applicatie. In paragraaf “4.2 Het onderzoek” is meer te lezen over zowel de uitvoer als de resultaten van het betreffende onderzoek.

In de weken na het onderzoek heb ik een aantal trainingen gevolgd om beter voorbereid te zijn op het daadwerkelijk bouwen van het eindproduct. Over de gevolgde trainingen, en mijn ervaringen hierbij, is meer te lezen in respectievelijk de paragrafen “4.3 Training: Developing web applications with ASP.NET MVC” en “4.4 Training: Developing web applications using jQuery and HTML 5”.

Tussen de twee trainingen zaten nog ongeplande dagen. Op deze dagen heb ik de requirements en daarmee een basaal functioneel ontwerp opgesteld. Ook is in deze tijd de basis van een technisch ontwerp opgezet met daarin de algemene structuur van het te bouwen product. Over zowel het functioneel als het technisch ontwerp is meer te lezen in paragraaf “4.5 Het Functioneel en het technisch ontwerp”.

Nadat de requirements tegen het einde van week 9 goedgekeurd werden door de opdrachtgever kon de stap naar de volgende fase, elaboratie, gezet worden.

4.1 HET PLAN VAN AANPAK

Van het plan van aanpak een eerste concept versie opbouwen kostte één week. Op de maandag erna, in week 3, had ik een introductiegesprek op school met mijn docent begeleider. Na enkele formaliteiten gerelateerd aan het afstuderen werd de eerste versie direct goedgekeurd.

Mijn opdrachtgever en technisch begeleider vonden de eerste versie echter niet specifiek genoeg. Aan de hand van hun commentaar maakte ik de nodige aanpassingen en stuurde het opnieuw op voor goedkeuring of verdere feedback. Dit op en neer sturen van nieuwe versies en feedback herhaalde zich gedurende week 3 met steeds dezelfde conclusie dat het verslag niet specifiek genoeg was. Het zag er naar uit dat er zich een miscommunicatie gevormd had tussen mij en mijn begeleiders.

In week 4 werd duidelijk waar de miscommunicatie had plaatsgevonden. Ik had uit verschillende bronnen de definities van de door mij gekozen ontwikkelingsmethode, RUP, opgezocht. Waar ik achter kwam is dat deze door de industrie heen op heel verschillende wijzen geïnterpreteerd en uitgevoerd wordt. Het bleek dat mijn interpretatie behoorlijk verschilde met die van mijn technisch begeleider.

Om de zaken duidelijk op een rijtje te krijgen ben ik naar het opleidingscentrum van Info Support in Veenendaal gegaan, waar mijn technisch begeleider werkzaam is. Hier heb ik met mijn technisch begeleider de te gebruiken ontwikkelingsmethodiek besproken. De RUP-methodiek, en de combinatie van deze met elementen van de SCRUM-methodiek, werd aan mij uitgelegd. Deze combinatie van methodieken werd mij aangeraden omdat deze zeer praktisch is binnen mijn type project. Dit omdat ik met veel technische onbekendheden te maken had, en ik op deze manier deze onbekendheden ofwel risico's iteratief kon afstrepen. Daarnaast is deze methodiek gewenst en alom gebruikt binnen Info Support, en zou het gebruik ervan me er veel inzicht in geven.

Ik besloot het advies van mijn technisch begeleider op te volgen om de combinatie van RUP en SCRUM te gaan gebruiken binnen mijn project. Ik verwerkte de methodiek in mijn plan van aanpak, paste nog her en der wat aan en verstuurde het voor de laatste keer naar mijn begeleiders van zowel de Fontys als Info Support. Hierna werd unaniem besloten dat het plan van aanpak goedgekeurd was en ik door kon om aan mijn onderzoek te beginnen.

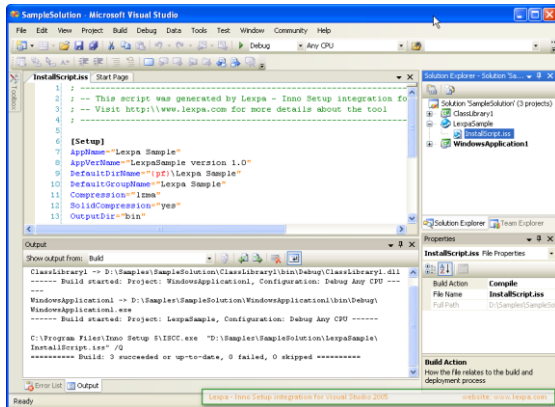
Het uiteindelijke plan van aanpak is terug te vinden in Bijlage A van dit document.

4.2 HET ONDERZOEK

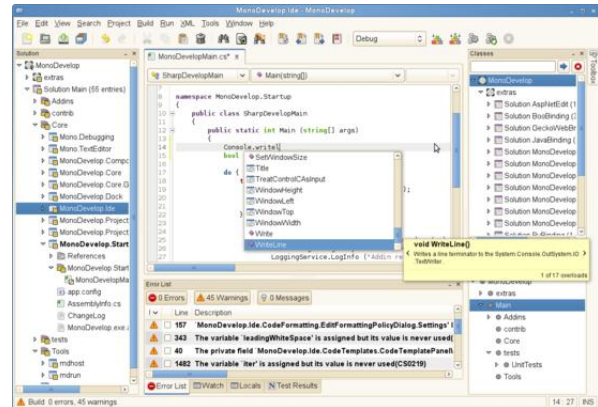
Het onderzoek begon, precies volgens planning, in week 5. Er zou door mij uitgezocht worden of het, specifiek voor het betreffende project, beter zou zijn om een native iPad-applicatie of een web-applicatie te bouwen. Deze applicatie zou binnen het uiteindelijke product verantwoordelijk zijn voor de gebruikersinterface. Tijdens het wachten op de feedback voor mijn plan van aanpak had ik al het initiatief genomen een onderzoeksplan te schrijven. Dit onderzoeksplan, welke terug te vinden is in het plan van aanpak ofwel bijlage A, is na enkele overwegingen toen verwerkt in het plan van aanpak. Dit heeft mij een duidelijke leidraad gegeven tijdens het uitvoeren van het onderzoek.

Het onderzoek begon met het zoeken naar de specifieke voordelen per vorm van applicatie. Hier merkte ik al dat beide keuzes grote voordelen hadden. Verbazingwekkend was echter wel dat de hoeveelheid voordelen die betrekking hadden tot functionaliteiten benodigd voor het project, relatief gezien, klein was. Verderop in deze paragraaf worden de voordelen die wel significant waren voor het project uitgelicht.

Na het zoeken van de voordelen per applicatievorm werd onderzoek gedaan naar de bruikbaarheid van de te gebruiken ontwikkelomgevingen. In het geval van een native app zou gebruik moeten worden gemaakt van het Mono-framework, aangezien dit de wenste methodiek is binnen Info Support. Het mono-framework is een platform om met behulp van C# applicaties te schrijven voor allerlei platformen, met daarbij de MonoDevelop compiler en X-Code om de gebruikersinterface op te bouwen. In het geval van een web-applicatie zou enkel gebruik worden gemaakt van Visual Studio als de ontwikkelingsomgeving.



Figuur 2: Visual Studio



Figuur 3: MonoDevelop

Omdat de ontwikkelaars van MonoDevelop het besluit hadden genomen deze zoveel mogelijk te laten lijken op Visual Studio waren er meer overeenkomsten dan verschillen tussen beiden. Beide omgevingen waren verder, functioneel gezien, zeer capabel en boden alle tools aan die men van een moderne ontwikkelomgeving mag verwachten.

Nadat alle gegevens verzameld waren, moesten deze geanalyseerd en vergeleken worden.

De eerste analyse was die van de voordelen per keuze. Er moest gezocht worden naar voordelen die niet alleen relevant waren, maar ook relevant waren voor het afstudeerproject. De voordelen die naar voren kwamen om voor een web-applicatie te kiezen waren dat het direct op meerdere platformen zou werken en dat updates direct doorgevoerd konden worden. Voor het kiezen voor de vorm van native applicatie gold maar één voordeel: Dat visueel gezien alles veel mooier en vloeiender weergegeven kon worden. Dit was dankzij het feit dat native applicaties veel directer gebruik kunnen maken van de hardmatige kracht van de iPad.

De tweede analyse betrof de onderzochte ontwikkelomgevingen. Dankzij hun vele overeenkomsten zou je denken dat er weinig significante argumenten zouden zijn waarom er voor één bepaalde oplossing gekozen zou moeten worden. Er waren echter twee argumenten ten voordele van Visual Studio. Er was namelijk veel meer documentatie aanwezig voor Visual Studio dan voor MonoDevelop. Ook had Visual Studio veel meer gebruikers. Hierdoor is voor Visual Studio een veel grotere ondersteunende community dan voor MonoDevelop. Deze community is zeer nuttig aangezien deze veel support kan leveren aan ontwikkelaars.

De uiteindelijke conclusie van het onderzoek was dat het beter was om te kiezen voor het ontwikkelen van een web-applicatie. De aanbeveling tegenover het overkoepelende project, wat dus de afstudeeropdracht is, is om een web-applicatie te bouwen die strak en minimalistisch is vormgegeven. Dit om de gebruikerservaring zo simpel en gestroomlijnd als mogelijk, met de te

gebruiken middelen, te maken. Op deze manier zou gebruik worden gemaakt van de voordelen van een web-applicatie, en tegelijkertijd het nadeel tegenover native apps, een minder vloeiende gebruikerservaring, zoveel mogelijk wegnemen.

Indien een nog diepgaandere blik in de onderzoekplanning of het onderzoek zelf gewenst is, kan gekeken worden in respectievelijk bijlage A, waar de onderzoekplanning in het plan van aanpak verwerkt zit, en bijlage B, het onderzoek.

4.3 TRAINING: DEVELOPING WEB APPLICATIONS WITH ASP.NET MVC

In Week 8 ben ik begonnen aan de training “Developing web applications with ASP.NET MVC”. Deze werd gegeven door Simona Colapicchioni. In deze training werd les gegeven over MVC 3, een Microsoft implementatie van het MVC principe (Model View Controller). Deze maakt gebruik van hun bestaande web-ontwikkelingstools ASP.NET.

Voor ik aan deze training begon wist ik bijzonder weinig van de ASP.NET en MVC 3 technologieën. Daarnaast had ik al een hele tijd niet meer met de programmeertaal C# gewerkt. Dankzij de goede inrichting en begeleiding binnen de training had ik ondanks mijn onervarenheid alle genoemde onderwerpen binnen no-time onder de knie.

Mijn mening tegenover Microsoft en hun producten is cynisch te noemen. Vaak is de technologie er wel maar is de oplevering en implementatie in mijn ogen niet correct uitgevoerd. Ondanks deze blik was ik prettig verrast met het MVC 3 framework. Microsoft had voor deze framework open-source technologieën, zoals jQuery en Modernizr, omarmd en modulair geïntegreerd. Het totaalpakket ging meer om het organiseren van bestaande web-technologieën in plaats van het verzinnen en opdringen van nieuwe datavormen. Het was makkelijk in gebruik en gaf een duidelijke structuur. Dankzij het overzicht wat MVC 3 creëert binnen een project wordt het ontwikkelingsproces van web-applicaties gigantisch versneld.

Wat jammer was aan MVC 3 was dat ik grote delen niet direct kon gebruiken. Validatie en Ajax waren op dusdanige wijze geïmplementeerd dat er een vaste collectie van data nodig was om het correct te laten functioneren. Dit terwijl in mijn afstudeerproject zowel het type object als de validatie die erbij hoort behoorlijk kunnen verschillen. Om deze reden zou ik alsnog veel handmatig, met behulp van C#- en Javascript-code, moeten doen.

4.4 TRAINING: DEVELOPING WEB APPLICATIONS USING JQUERY AND HTML 5

In week 9 volgde ik de training “Developing web applications using jQuery and HTML 5”. De docent binnen deze training was Erno de Weerd. Het leermateriaal betrof de nieuwe HTML 5 standaard en de Javascript-framework jQuery. Het framework jQuery is gebouwd om de inconsistenties binnen verschillende browsers, aangaande de uitvoering van Javascript-code, opvang te vangen voor de ontwikkelaar. Hierdoor kan de ontwikkelaar zich op zijn project focussen zonder zich druk te hoeven maken over problemen in de Javascript-intepretaties.

Omdat ik in de te behandelen stof in de voorgaande weken al veel onderzoek had gedaan kwam ik minder nieuwe stof tegen dan het geval was in de vorige training. Tijdens deze week ben ik echter wel dieper ingegaan op zowel de programmeertaal Javascript als de library jQuery. Het is namelijk zo dat Javascript een bijzonder dynamische taal is die qua uiterlijk bijzonder veel lijkt op programmeertalen als C en Java. Echter bevat deze in werking veel uitzonderingen en eigenaardigheden waar de meeste ontwikkelaars zich niet van bewust zijn. Dankzij deze

onwetendheid wordt de moeilijkheid van Javascript onderschat, wat een oorzaak kan worden van slechte code en trage voortgang binnen ontwikkelingsprojecten.

Concluderend was de training bijzonder nuttig om verdieping te geven in stof die ik enkele weken ervoor nog amper had behandeld. De geworven kennis zou door mijn hele project heen gebruikt gaan worden gebruiken.

4.5 HET FUNCTIONEEL EN HET TECHNISCH ONTWERP

Tijdens de loop van het project was het bijzonder belangrijk dat zowel de opdrachtgever als ikzelf op de hoogte waren van de gespecificeerde eisen voor het eindproduct. Om deze reden heb ik in de inceptiefase de basis van een functioneel ontwerp opgesteld.

In het functioneel ontwerp stonden de eisen globaal gespecificeerd en de bijbehorende use cases gedeclareerd. In de volgende fasen, de elaboratie- en constructiefase, zouden de eisen verder uitgediept worden en uitgewerkte versies van de gedeclareerde use cases toegevoegd worden. De gedeclareerde use cases staan in Figuur 4 in tabelvorm weergegeven.

Use Case ID	Use Case
UC 1	Overzicht van cliënten bekijken.
UC 2	Binnen het overzicht van cliënten zoeken
UC 3	Een cliënt in detail bekijken.
UC 4	Een nieuwe cliënt aanmaken.
UC 5	De details van een cliënt aanpassen.
UC 6	Een nieuwe aanvraag maken voor een cliënt.
UC 7	Bij het maken van een aanvraag terug springen naar een vorige stap
UC 8	Een aanvraag tijdelijk opslaan voor latere hervatting.
UC 9	Een tijdelijk opgeslagen aanvraag hervatten.
UC 10	Een tijdelijk opgeslagen aanvraag verwijderen
UC 11	Inloggen in de applicatie.
UC 12	Uitloggen uit de applicatie.

Figuur 4: Overzicht van gedeclareerde use cases

Wanneer men het functioneel ontwerp definieert als de “wat” is binnen het project is het technisch ontwerp de “hoe”. Het is een hulpmiddel voor de ontwikkelaar om de architectuur van het eindproduct te definiëren. Het geeft voor de ontwikkelaar en buitenstaanders een duidelijk overzicht van de technische kant van het project.

De aanpak die ik probeer te handhaven binnen het ontwikkelen van software is om een duidelijke vanzelfsprekende opbouw binnen de applicatie te handhaven. Op deze manier is er weinig extra documentatie nodig om duidelijk te krijgen hoe de applicatie is opgebouwd. Omdat ik deze aanpak handhaaf beschrijf ik in mijn technisch ontwerp alleen overzichten van de architectuur en entiteiten van specifieke componenten.

5. ELABORATIEFASE

In de elaboratiefase wordt een architecturale basis gelegd door een werkende applicatie met beperkte mogelijkheden op te leveren. Ook wordt in deze fase een overeenstemming met de opdrachtgever geformuleerd betreffende de requirements. De doelstellingen worden behaald door de functionele eisen te specificeren, de use cases uit te werken, technische ontwerpen te maken en te bouwen aan het betreffende product. De werkzaamheden worden in deze fase in iteraties ofwel “Sprints” uitgevoerd. Om een overzicht te behouden van werkzaamheden per sprint inclusief hun voortgang heb ik gebruik gemaakt van een product-backlog, één van de tools uit in de SCRUM-methodiek.

De elaboratiefase begon in sprint één met een sprintmeeting met mijn opdrachtgever. Hierin werden de eerste use cases besproken. Dit om in groter detail inzicht te krijgen in de requirements van de opdrachtgever. Ook heb ik in dat gesprek aangegeven welke middelen, zoals databases en inloggegevens voor bestaande software, ik nodig zou hebben door de loop van de ontwikkeling. Aan het einde van de sprintmeeting kreeg ik van mijn opdrachtgever te horen dat ik professioneel gezien niet voldoende communiceerde. Ik kon hem zodanig niet voldoende op de hoogte houden betreffende de voortgang. Na deze opmerking heb ik er direct werk van gemaakt, hoe ik hierbij te werk ben gegaan is te lezen in paragraaf “5.1 Verbeterpunt: communicatie”.

Binnen Info Support worden applicaties op basis en met behulp van de ontwikkelstraat, genaamd Endeavour, gebouwd en onderhouden. Om meer inzicht te krijgen in de manier dat de bestaande code werkt moest ik meer te weten komen over hoe de Endeavour architectuur gestructureerd is. Deze kennis is mij toen overgedragen door Rik Meijer, een collega binnen Info Support. Over de kennis die ik op dat moment over Endeavour heb verkregen is meer te lezen in paragraaf “5.2 De ontwikkelstraat Endeavour”.

Na de nodige kennis vergaard te hebben, heb ik de algemene grafische gebruikersinterface voor de applicatie als geheel uitgewerkt. Binnen dit proces heb ik allerlei dingen uitgeprobeerd en uiteindelijk zelfs een jQuery-plugin gebouwd om swipe bewegingen, dus vegen over het scherm van een tablet, van zowel touch als muisbewegingen te herkennen en te verwerken. Het proces aangaande het maken van een goede gebruikersinterface wordt in groter detail beschreven in paragraaf “5.3 De algemene gebruikersinterface”.

De tweede week van de eerste sprint stond in het teken van de connectie leggen met de bestaande databases om data over cliënten en hulpmiddelen op te kunnen halen en weer te geven binnen de applicatie. Ondanks het feit dat ik de architectuur van endeavour elementair begreep bleek het echter bijzonder moeilijk om als beginner meteen gebruik te maken van het bestaande framework. Om niet onnodig veel tijd kwijt te raken heb ik toen de assistentie van Bram den Boer, een collega binnen Info Support, gevraagd. Omdat het om een oudere versie van Endeavour ging moest zelfs Bram nog even onderzoeken en uitproberen voor we uiteindelijk alles werkende zouden krijgen.

In de tweede sprint ben ik beginnen met het bouwen van het systeem waar de gebruiker een nieuwe aanvraag zou kunnen opstellen en versturen. In de eerste week heb ik gepoogd een opbouw te verzinnen om de bestaande entiteiten te kunnen gebruiken om de vragen, met behulp van een nieuwe template, te kunnen weergeven. Tegen het einde van de week was ik echter overweldigd door de complexiteit van zowel het nieuwe als het oude en heb ik uiteindelijk een

noodoproep naar mijn begeleiders binnen Info Support gestuurd. In deze oproep maakte ik duidelijk dat mijn afstuderen in gevaar kon zijn en er maatregelen nodig waren. Ik had afspraken ingepland met zowel mijn opdrachtgever als technisch begeleider om de huidige stand van zaken te bespreken.

De week erna ben ik om de tafel gaan zitten met mijn beide begeleiders. Ik had van tevoren al een plan van maatregelen opgesteld die de requirements lichtelijke versoepelden en mij in staat stelden mijn eigen data-format te gebruiken. Op deze manier hoefde ik niet met de complexe structuur van het bestaande framework te werken en kon ik me focussen op het de complexiteit van het nieuwe systeem.

Na een kort overleg met mijn begeleiders gingen beiden akkoord met het plan. Ook werd mijn project als proof of concept bestempeld. Hiermee werd de intentie van de opdrachtgever voor het afstudeerproject duidelijk gesteld en konden daarmee de prioriteiten beter afgesteld worden. De rest van de week kon ik toen besteden aan het beredeneren en opzetten van de nodige entiteiten voor het maken van een aanvraag. Om meer te weten te komen over hoe de problemen in sprint 2 ontstaan en aangepakt zijn kan er gekeken wordt in paragraaf “5.4 SOS”.

De derde sprint, en daarmee ook de laatste binnen de elaboratiefase, was er een van verandering. Omdat ik in de afgelopen weken met allerlei nieuwe technieken in aanraking was gekomen en er toch een aantal problemen zijn geweest, werd het project een beetje rommelig naar mijn mening. De gebruikersinterface begon zijn consistentie te verliezen en werd op sommige plekken met behulp van grote hoeveelheden code gegenereerd. Om deze redenen heb ik 2 externe open source frameworks geïntegreerd binnen het project.

Om de gebruikersinterface op te schonen heb ik “Twitter Bootstrap” gebruikt. Een framework bestaande uit CSS en Javascript. Hiermee kan men in een korte tijd een web-applicatie visueel aantrekkelijk en voor verschillende apparaten schaalbaar maken. Om volledig gebruik te maken van de schaalbaarheidsopties binnen Twitter Bootstrap heb ik er toen voor gekozen de gebruikersinterface volledig te vernieuwen met behulp van Bootstrap.

Om de code op te schonen heb ik de Javascript library “EJS” ofwel “EmbeddedJS” geïntroduceerd, een templating engine om dynamisch een gebruikersinterface te genereren op basis van meegegeven waarden. Over beide frameworks is meer te lezen in respectievelijk de paragrafen “5.5 Twitter Bootstrap” en “5.6 EmbeddedJS”.

In de tweede helft van de sprint kon ik dankzij mijn nieuwe opbouw in zeer korte tijd de volledige weergave van vragenlijsten construeren. De hoeveelheid onderzoeken en ontwerpen in mijn dagelijkse bezigheden werd verruild voor het effectief bouwen van het eindproduct. Met deze ontdekking kon ik concluderen dat ik in de overgang van sprint 3 naar 4 de constructiefase in ben gegaan. Wellicht met grote toeval dat deze overgang exact volgens planning plaatsvond.

5.1 VERBETERPUNT: COMMUNICATIE

In mijn eerste sprintmeeting kreeg ik van mijn opdrachtgever te horen dat mijn communicatie, met hem en de andere betrokkenen, niet voldoende was om iedereen op de hoogte te houden. De problemen lagen in de hoeveelheid voortgangsberichten en de duidelijkheid daarbinnen.

Wat betreft de kwantiteit van voortgangsberichten had ik vóór de inceptiefase een afspraak gemaakt om elke twee weken mijn voortgang door te communiceren. Al gauw bleek echter dat in

een dergelijk project deze mate van communicatie niet voldoende was. Dit was te danken aan de dynamiek van het project en de toenmalige onduidelijkheden die continue zorgden voor grote wijzigingen in de loop van het project.

Ook over de kwaliteit van de voortgangsberichten viel te twisten. Er werd netjes een logboek bijgehouden, maar dit gaf alleen een basaal idee van wat er zich allemaal afspeelde. Er was al een half afgewerkte product-backlog, maar deze was nog niet gebruikt voor communicatie met de opdrachtgever. In retrospectief heb ik toendertijd de product-backlog zwaar onderschat in zijn vermogen om verwachtingen tussen de opdrachtgever en opdrachtnemer te synchroniseren.

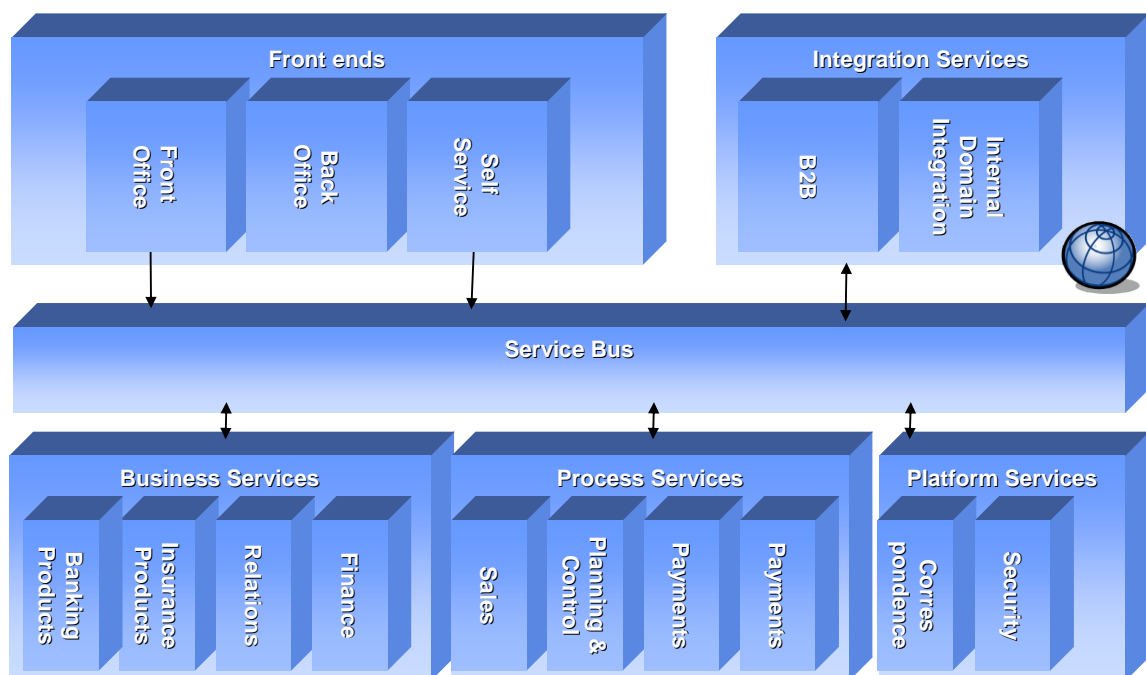
Direct nadat de feedback was gegeven werd besproken wat een mogelijke oplossing zou kunnen zijn. Uiteindelijk stelde ik voor minstens één keer per week een voortgangsbespreking in te plannen, en daarnaast eventueel informeel kleinigheden te bespreken. Daarnaast zou ik het product-backlog vanaf dat moment accuraat gaan bijhouden en voorleggen bij voortgangsbesprekingen. De opdrachtgever ging akkoord.

Toen ik enkele weken later informeerde naar de vooruitgang betreffende de communicatie bleek er grote tevredenheid te zijn. Niet alleen omdat er nu veel meer en effectiever gecommuniceerd werd, maar ook omdat er in dergelijk korte tijd een dusdanige verbetering had plaatsgevonden.

5.2 DE ONTWIKKELSTRAAT ENDEAVOUR

De definitie van een ontwikkelstraat is “Het totaal van tools, methoden, processen en mensen met als doel het effectief en productief ontwikkelen van software”. Binnen Info Support is gekozen om niet een bestaande ontwikkelstraat te gebruiken, maar zelf een nieuwe te ontwikkelen, genaamd Endeavour.

Er zijn veel aspecten die in Endeavour behandeld worden, maar binnen mijn afstudeeropdracht ben ik eigenlijk alleen in aanraking gekomen met de architectuur die gebruikt wordt voor enterprise applicaties. Hierover zal ik dan ook een bondige uitleg geven.



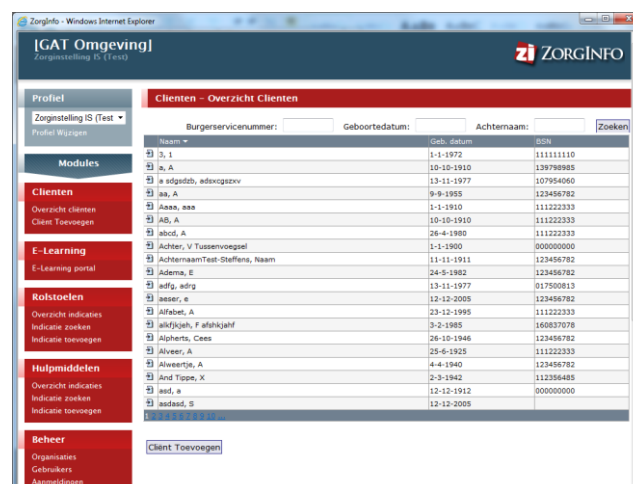
Figuur 5: Een overzicht van de Endeavour enterprise architectuur

Binnen de Endeavour enterprise architectuur, welke gebaseerd is op een Service-Oriented Architecture (SOA), is de zogenaamde “Service Bus” de centrale kracht die alle andere componenten met elkaar verbindt. De front end vraagt in deze architectuur een bepaalde service aan bij de Service Bus om aan te kunnen spreken, bijvoorbeeld om toegang te krijgen tot een onderliggende database. De Service Bus probeert vervolgens een instantie van een service te vinden om te retourneren. Als er nog geen instantie functioneel is zal de Service Bus proberen een instantie op te starten. Nadat de betreffende service is geretourneerd aan de front end kan deze een referentie opslaan om directe toegang tot de service te behouden zonder tussenkomst van de Service Bus.

5.3 DE ALGEMENE GEBRUIKERSINTERFACE

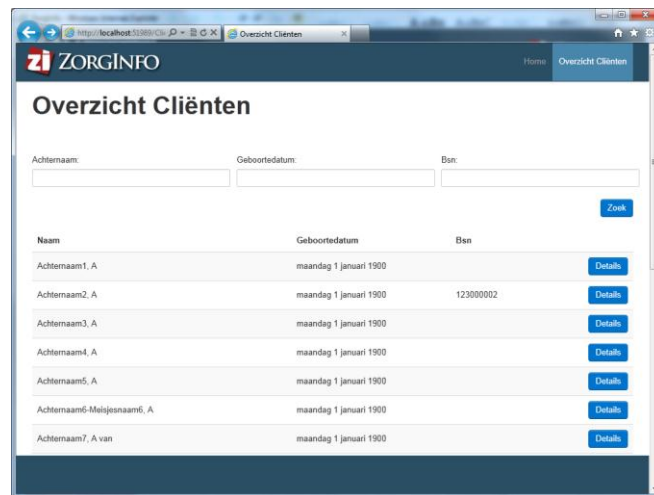
Bij het onderzoeken van grafische gebruikersinterfaces voor de iPad was ik gefascineerd door het gebruik van swipe bewegingen. Dankzij het feit dat één van de requirements is dat de applicatie ook op andere platformen moet kunnen werken, kon ik echter geen gebruik maken van de iOS-specifieke swipe-libraries. Om deze reden besloot ik een jQuery plugin te schrijven die het makkelijk maakt voor de ontwikkelaar om verschillende swipe-bewegingen te registreren en erop te reageren.

In het eerste prototype van de algemene gebruikersinterface werd het hoofdmenu naast de inhoud van de pagina gepositioneerd. Al snel bleek echter dat er op deze manier niet altijd voldoende ruimte beschikbaar was voor de inhoud van de pagina. Tablets en telefoons kunnen namelijk in portrait- en landscape-modus fungeren. In portrait-modus wordt het apparaat dusdanig gepositioneerd dat het scherm hoger is dan breed en omgekeerd in landscape-modus. In landscape-modus werkte de grafische gebruikersinterface volledig naar behoren. In portrait-modus was echter niet voldoende ruimte over om de inhoud netjes op het scherm weer te geven.



Figuur 6: De gebruikersinterface van de originele ZorgInfo applicatie

Een schermweergave die je binnen veel Apple producten ziet is dat de volledige scherm inhoud aan de kant schuift en hiermee eerder verborgen informatie onthult. Dit creëert de perceptie dat er twee lagen binnen je applicatie zijn die over elkaar heen liggen. Geïnspireerd door deze weergave besloot het probleem van ruimte-tekort binnen mijn applicatie op te lossen met dit principe. Het hoofdmenu werd volledig verscholen achter de rest van de pagina en kon, dankzij mijn jQuery-plugin, met behulp van een swipe beweging vanaf de rand van het scherm tevoorschijn gehaald worden.



Figuur 7: De gebruikersinterface van de nieuwe, voor mobiele platformen geoptimaliseerde, ZorgInfo

Nadat alles opgebouwd en werkende was, verbaasde het me hoeveel mogelijkheden er waren bij het bouwen van web-applicaties die gebruik maken van de mogelijkheden van mobiele platformen. Met een simpele veegbeweging vanaf de schermrand schoof de hele website aan de kant om de verborgen content erachter weer te geven. Het gevoel dat de gebruiker zich in een webbrowser bevond en naar een website zat te kijken verdween direct. Het is ongelooflijk hoeveel verborgen, door ontwikkelaars ongebruikt, potentieel nog aanwezig was binnen het domein van web-ontwikkeling!

Na enkele gebruikerstests bleek dat de swipe-beweging voor veel gebruikers onduidelijk was. Onder andere omdat gebruikers van tevoren moesten weten waar het hoofdmenu verborgen zat voor ze er gebruik van konden maken. Uiteindelijk heb ik besloten de mogelijkheid van swipen intact te laten, maar hiernaast ook als alternatief een knop aan te bieden die dezelfde werking zou activeren.

5.4 SOS: SAVE OUR SOFTWARE-PROJECT

Het oorspronkelijke idee binnen het afstuderen was dat ik een nieuw type template-structuur zou introduceren, mogelijk in de vorm van een XML-bestand, waarin de vragen van een aanvraag naar voren zouden komen. De antwoorden op de vragen zouden vervolgens opgeslagen worden in het bestaande ecosysteem van databases.

Na een week brainstormen, op zoek naar een mogelijke opbouw, bleek echter dat dit toch een stuk moeilijker was dan oorspronkelijk gedacht. Ten eerste was het bouwen van een dergelijk systeem een behoorlijke uitdaging, al helemaal gezien de tijdsspan en kennis die ten tijde van een afstudeerproject aanwezig is. Ten tweede leek het bestaande framework redundant informatie op te slaan en business-logica op vreemde wijze in te delen. Een voorbeeld hiervan is dat de logica voor de validatie in de definitie van de gebruikersinterface op te slaan terwijl de definitie van de vragen zelf op een andere plek opgeslagen stond.

De combinatie van de complexiteit die betrokken is bij het bouwen van een dergelijk systeem en de onduidelijkheden in het te gebruiken framework zorgden ervoor dat ik het overzicht kwijt raakte. Ik kreeg steeds minder vat op hoe ik mijn afstudeeropdracht zou aanpakken en werd een beetje angstig dat ik toch een te moeilijke opdracht had gekozen om mee af te studeren.

Het beste plan van aanpak leek op dat moment alle betrokkenen in te lichten om te zien hoe zij er tegenaan keken. De hoop, op het moment van inlichten, was om samen met hen tot een oplossing te komen. Voordat ik de daadwerkelijke bespreking zou hebben, had ik echter al een pakket van oplossingen samengesteld.

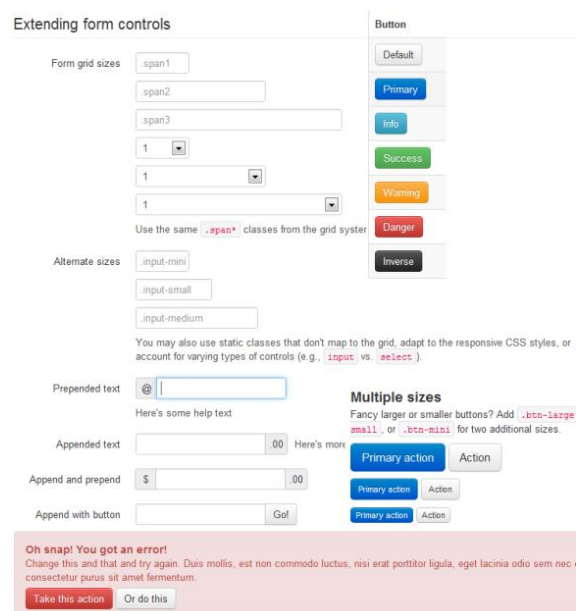
Toen de week erna de besprekingen met alle betrokkenen plaatsvonden legde ik aan hen allen mijn pakket van oplossingen voor: Een lichte aanpassing in de requirements samen met het idee om de aanvragen in een nieuwe database op te slaan in mijn eigen ontworpen format. Zo kon ik me volledig focussen op de complexiteit van het nieuwe systeem in plaats van me ook nog druk te moeten maken om compatibiliteitsproblemen.

Mijn pakket van oplossingen werd tijdens de genoemde besprekingen goedgekeurd door mijn opdrachtgever en technisch begeleider. Daarnaast werd mijn project als proof of concept bestempeld. Op deze manier kwam de intentie van mijn opdrachtgever ten opzichte van dit project meer naar voren. Deze was namelijk meer om antwoorden te vergaren betreffende vraagstukken als de uitbreidbaarheid van ZorgInfo en de aspecten die aan bod komen binnen een applicatie geoptimaliseerd voor mobiele platformen.

5.5 TWITTER BOOTSTRAP

Twitter Bootstrap is in essentie een front-end toolkit, bestaande uit een collectie van CSS en HTML conventies. Het helpt front-end ontwikkelaars bij het snel ontwikkelen van flexibele en mooi vormgegeven web-applicaties. Dit doet Bootstrap door de nieuwste web-technologieën te integreren en de ontwikkelaar te voorzien van een pakket van krachtige hulpmiddelen als een grid-systeem om componenten te positioneren, een navigatie-menu dat gemakkelijk geïntegreerd kan worden in iedere web-applicatie en een uniforme styling die door de hele web-layout gebruikt kan worden.

Het Bootstrap project is ontstaan binnen Twitter toen inconsistenties begonnen op te treden in de gebruikersinterface van verschillende deelsystemen binnen het bedrijf. Dit maakte het moeilijk om alle deelsystemen naar behoren te onderhouden en uit te breiden. In een van Twitter's befaamde hackweeks, waarin werknemers een week lang mogen experimenteren en ontwikkelen buiten hun dagelijkse werkzaamheden om, had één team als experiment een stabiel prototype van Bootstrap gebouwd. Bootstrap werd daarna binnen zeer korte tijd uitgerold binnen het bedrijf Twitter en uiteindelijk publiekelijk beschikbaar gesteld voor ontwikkelaars over de hele wereld om betere web-applicaties te kunnen maken.



Figuur 8: Highlights Twitter Bootstrap

Twitter Bootstrap zal in de toekomst nog veel verder uitbreiden, maar belooft hierbij om altijd zo compact mogelijk te blijven wat betreft de grootte van benodigde bestanden.

5.6 EMBEDDEDJS

Als web-ontwikkelaar kom je regelmatig in de situatie dat je data binnenkrijgt met behulp van bijvoorbeeld een AJAX-request. Vervolgens wil de ontwikkelaar dan deze informatie weergeven in de grafische gebruikersinterface. De gebruikelijke manier hiervoor zou zijn om in code handmatig, in de vorm van een gigantische tekst-string, html-code te genereren en deze vervolgens binnen de bestaande gebruikersinterface te plaatsen.

De genoemde methodiek maakt een gebruikersinterface opbouwen, onderhouden en uitbreiden een lastige en tijdrovende taak. De oorzaak hiervan is grote hoeveelheden, en hierdoor onoverzichtelijke, Javascript-code.

Een oplossing hiervoor is de Javascript-library EJS ofwel EmbeddedJS. Met behulp van EJS maakt de ontwikkelaar een apart template-bestand waarin de definitie van een gebruikersinterface volledig wordt uitgewerkt. Deze template file wordt vervolgens in de Javascript-code van de betreffende pagina ingeladen. Hierna wordt met behulp van een stuk data de template uitgewerkt tot de uiteindelijke HTML-code. Deze kan vervolgens direct in de bestaande gebruikersinterface geïnjecteerd worden.

De voordelen van deze aanpak beginnen met het feit dat de definitie van de gebruikersinterface wordt gescheiden van de Javascript-code. De enige echte Javascript-code die je in een template zult vinden is degene die HTML-code positioneert en de betreffende data hierin verwerkt. Hierdoor is het template-bestand zelf ook nog relatief netjes vormgegeven qua code. En omdat alle code om de gebruikersinterface te genereren uit de pagina-code wordt gehaald wordt deze ook meteen een stuk netter en overzichtelijker.

Alternatieven van EJS die ik heb overwogen zijn onder andere KnockoutJS en HandlebarsJS. De reden dat ik specifiek voor EJS heb gekozen is dat men hierbinnen Javascript-code in plaats van een framework-specifieke scripting taal in de templates gebruikt. Hierdoor heeft de ontwikkelaar veel meer vrijheid om voorwaardes te controleren en de template te structureren.

Ik ben altijd al een sterke aanhanger geweest van het idee om logica en definities voor gebruikersinterfaces van elkaar te scheiden. Met behulp van een templating-engine als EJS wordt deze scheiding simpel en gestructureerd uitgevoerd.

6. CONSTRUCTIEFASE

De constructiefase lijkt in de essentie veel op de elaboratiefase. In beide fasen wordt het betreffende product op iteratieve wijze gebouwd. Het verschil zit in de balans tussen de constructie van het product en de combinatie van onderzoek en ontwerpen. In de elaboratiefase wordt namelijk veel aan functionele eisen, use cases en technische ontwerpen gewerkt, terwijl de nadruk binnen de constructiefase veel meer ligt bij het daadwerkelijke bouwen. De overgang van elaboratie naar constructie wordt in veel gevallen pas opgemerkt nadat deze heeft plaatsgevonden.

De constructiefase begon in de eerste sprint met het afmaken van het systeem dat verantwoordelijk is voor het opmaken en versturen, ofwel opslaan, van de aanvragen.

Zoals ik al verwachtte bij de implementatie versnelden de twee frameworks Twitter Bootstrap en EmbeddedJS mijn ontwikkelingsproces enorm. De grote hoeveelheden van afgeleverd werk per hoeveelheid tijd lieten mij een groot verschil tussen de elaboratie- en constructiefase zien die ik in eerdere stadia nog redelijk onderschatte.

Bijna alle requirements, die volgens het MoSCoW-model als zijnde “must” aangegeven stonden, waren in de tweede sprint van de constructiefase geïntegreerd binnen het product. Hetzelfde gold voor, op één na, alle andere requirements die redelijkerwijs nog mogelijk waren binnen de gegeven tijd. De enige requirement die nog ontbrak was de must-have requirement om de applicatie via SSL bereikbaar te maken, om dusdanig de privacy van de zorgverleners en de cliënten te beschermen.

Dankzij de kleine hoeveelheid werkzaamheden die nog over was voor deze laatste sprint besloot ik de opdrachtgever in de sprintmeeting te vragen naar details in de afwerking die hij graag in het eindproduct zou willen zien. Aangezien op dit moment de cliëntgegevens nogmaals ingevuld dienden te worden bij het opmaken van een aanvraag was één van de vragen om dit geautomatiseerd te laten gebeuren. Hiernaast had ik, omdat de applicatie met weinig schermruimte om moet kunnen gaan, specifiek gevraagd naar een selectie van weer te geven data in de tabellen door de applicatie heen. De enige wijziging die hierbij nog gewenst was, was op de pagina met cliëntdetails andere gegevens in de tabel van bijbehorende gemaakte aanvragen.

Nadat alle eerder genoemde werkzaamheden verricht waren, waren alle redelijkerwijs mogelijke eisen verwerkt in het eindproduct. Alle requirements die volgens het MoSCoW-principe aangegeven waren als zijnde “must” waren geïntegreerd.

Concluderend kan gesteld worden dat alle verplichte requirements, samen met nog enkele gewenste requirements, verwerkt waren binnen het eindproduct. Hierdoor was, al waren niet alle mogelijke requirements ingebouwd, het uiteindelijk opgeleverde product een succes.

7. TRANSITIEFASE

Normaal gezien wordt in de transitiefase het op te leveren product overgezet van de ontwikkelingsomgeving naar de productieomgeving. Echter omdat mijn eindproduct een proof of concept is, zal het product samen met conclusies en aanbevelingen alleen opgeleverd worden aan de opdrachtgever.

Helaas is het niet gelukt de transitiefase te bereiken voor de deadline om dit document op te leveren. Het is namelijk zo dat de transitiefase binnen het betreffende project gepland staat om op 11 juni 2012 te beginnen en dit document al 07 juni 2012 opgeleverd dient te worden.

De verwachting is dat de verdere loop van het afstudeerproject volgens planning zal verlopen. Zoals al eerder genoemd betekent dit dat vanaf 11 juni 2012 de transitiefase ingaat en daarmee alle beschreven documentatie samen het eindproduct opgeleverd worden aan de opdrachtgever.

8. CONCLUSIES EN AANBEVELINGEN

CONCLUSIES

Er is een proof of concept gerealiseerd die in feite een revisie is van de bestaande ZorgInfo applicatie. Deze revisie bevat een nieuwe manier van de templates voor de aanvragen te definiëren, en om deze weer te geven binnen de gebruikersinterface. Dit heeft zowel invloed op de gebruikersinterface als de achterliggende code, die nu nog meer modulair ingedeeld is. Verder is de gehele gebruikersinterface vervangen om de applicatie geschikt te maken voor mobiele platformen. Niet alle functionaliteiten van het oorspronkelijke ZorgInfo zijn geïmplementeerd, dit vanwege het feit dat veel van de bestaande functionaliteiten geen waarde toevoegen bij de activiteit van het indiceren van cliënten. En bij het indiceren van cliënten is waar de applicatie op mobiele platformen gebruikt zou worden.

Verder is er, ter voorbereiding van de realisatie van het eerder beschreven proof of concept, een onderzoek gedaan naar de keuze om voor de front-end een native iPad applicatie of een web-applicatie te bouwen. Uit dit onderzoek bleek dat, specifiek voor het betreffende project, de betere keuze was om een web-applicatie te bouwen. De redenen hiervoor waren dat bij het gebruik van web-applicaties updates direct doorgevoerd kunnen worden en de applicatie direct op meerdere platformen werkt. Twee voorwaarden waar de opdrachtgever bijzonder veel waarde aan hechtte.

Wat de opdrachtgever met het proof of concept wilde bereiken was meer kennis vergaren over de uitbreidbaarheid van de huidige ZorgInfo applicatie en over welke aspecten naar voren komen bij het ontwikkelen van applicaties die geoptimaliseerd zijn voor mobiele platformen.

Wat betreft de uitbreidbaarheid van de huidige ZorgInfo applicatie kan geconcludeerd worden dat deze redelijk beperkt is. Dit is te danken aan het feit dat sommige componenten binnen ZorgInfo verschillende functionaliteiten en daarbij verantwoordelijkheden op zich nemen die in essentie weinig met elkaar te maken hebben. Een voorbeeld hiervan is de definitie van de grafische gebruikersinterface die tegelijkertijd de regels voor de validatie definieert.

De aspecten die je tegenkomt bij het optimaliseren van applicaties voor mobiele platformen zijn zeer divers, maar er zijn een aantal aspecten die het meeste naar voren komen. Dankzij de variërende schermgrootte, die in veel gevallen ook aan de kleine kant is, moeten ontwikkelaars er rekening mee houden dat er niet tegelijkertijd te veel informatie weergegeven moet worden op het scherm en dat de gehele indeling voor de gebruikersinterface gebaseerd moet zijn op relatieve in plaats van gefixeerde verhoudingen. Ten slotte is het verstandig onderdelen waarmee de gebruiker interactie heeft wat groter te maken in omvang. Aangezien er in veel gevallen op mobiele platformen gebruik wordt gemaakt van een touch-screen kunnen controls in de gebruikersinterface beter te groot zijn dan te klein.

AANBEVELINGEN

Na alle ervaringen die ik binnen mijn afstuderen heb gehad zou ik graag nog enkele technische aanbevelingen willen geven. Deze zijn gericht aan mezelf en de ontwikkelaars binnen Info Support met als doel betere producten te leren maken.

Template-Engines

De eerste aanbeveling is om gebruik te maken van template-engines. Dit houdt in dat in een taal als XML of een combinatie van HTML en Javascript een dynamische gebruikersinterface gedefinieerd wordt. Deze kan dan vervolgens met behulp van de template-engine in code aangeroepen worden om vervolgens weergegeven te kunnen worden voor de gebruiker.

Dit scheidt de definitie van de gebruikersinterface van de code die verantwoordelijk is voor de logica binnen het project. Hierdoor wordt zowel de code voor de gebruikersinterface als de logica meteen een stuk netter en overzichtelijker gemaakt. Deze aanpak resulteert in een project wat makkelijker is om te onderhouden en uit te breiden.

In de context van front-end webdesign raad ik de templating-engines EmbeddedJS en HandlebarsJS aan. In het geval dat er veel waarde-controles plaatsvinden binnen de template ligt mijn voorkeur bij EmbeddedJS, omdat deze Javascript in plaats van een specifieke scripting-taal gebruikt binnen zijn templates. Dit geeft veel meer vrijheid in de verwerking van data binnen de template. Indien een simpele templating-engine waarbij slechts enkele controles plaatsvinden raad ik HandlebarsJS aan. Deze is weliswaar iets beperkter in functionaliteit maar voegt veel waarde toe op het vlak van leesbaarheid en daarbij onderhoudbaarheid.

Response Webdesign

De tweede aanbeveling betreft het schaalbaar maken van de gebruikersinterface binnen een web-applicatie. Wat hiermee bedoeld wordt is om het mogelijk te maken dat de gebruikersinterface zich kan aanpassen aan de form-factor van het apparaat waarop de web-applicatie bekeken wordt. Deze techniek wordt hedendaags ook wel “Response Webdesign” genoemd.

Frameworks die gebruikt kunnen worden om te assisteren in het schaalbaar maken van je interface zijn over het algemeen combinaties van CSS en Javascript-code. Een framework wat ik persoonlijk kan aanraden is “Twitter Bootstrap”, maar alternatieven hiervoor zijn onder andere “Easy front-end framework” en “Foundation”.

Scheiding van Verantwoordelijkheden

De laatste aanbeveling gaat niet zo zeer over tools of frameworks maar meer over een manier van werken. Een probleem wat ik tijdens mijn afstuderen ben tegengekomen is namelijk dat onderhoudbaarheid en vooral uitbreidbaarheid worden geremd op het moment dat componenten te diverse functionaliteiten op zich nemen.

Een voorbeeld van binnen mijn afstuderen zou zijn dat de regels voor validatie samen met de opmaak van gebruikersinterface gedefinieerd worden.

De oplossing hiervoor is simpel: Geef componenten binnen een software-project maar één taak of verantwoordelijkheid samen met een duidelijke interface voor andere componenten om aan te kunnen spreken. Dit resulteert in een projectstructuur waarin componenten makkelijk te onderhouden, te vervangen en te verwisselen zijn.

EVALUATIE

Het afgelopen half jaar heb ik veel ervaring en kennis opgedaan. Wat betreft vaardigheden heb ik me in het bijzonder verbeterd in het zelfstandig werken en het professioneel communiceren. Maar ook qua kennis heb ik me bijzonder ver ontwikkeld. Beide aspecten zullen naar voren komen in deze evaluatie van mijn afstuderen.

Zelfstandig werken en professioneel communiceren

In het verleden had ik voornamelijk in teamverband aan projecten gewerkt. Hierin nam ik niet al te vaak de vooraanstaande rol om met de opdrachtgever te onderhandelen over de requirements of de voortgang. Omdat ik tijdens mijn afstuderen geen projectteam had om op terug te vallen, dwong dit me mezelf verder te ontwikkelen.

Het leerproces begon in de eerste sprint van de elaboratiefase. Toen werd ik erop gewezen dat mijn communicatie met de opdrachtgever niet voldoende was. Vanaf dat moment heb ik alles op alles gezet om mijn zelfstandig werken en professionele communicatie te verbeteren. Ik ben netjes mijn voortgang en planning gaan documenteren, om deze vervolgens te overleggen met mijn opdrachtgever en overige begeleiders.

Toen in de tweede sprint het niveau van de afstudeeropdracht te hoog dreigde te worden voor mijn ervaring en beschikbare tijd werd ik wederom gedwongen zelf actie te ondernemen. Ik heb toen mijn begeleiders en opdrachtgever ingelicht over de situatie en mogelijke oplossingen besproken.

Tegen het einde van de elaboratiefase nam ik een lichtelijk risicovolle beslissing om grote delen van het product van een revisie te voorzien. Dit kostte aardig wat tijd, maar beloofde in mijn ogen een beter product en een snellere ontwikkeling in latere stadia. Aan de hand van het feit dat ik dit soort beslissingen redelijk makkelijk kon maken merkte ik zelf al grote vooruitgang in mijn zelfstandigheid. Ik durfde op eigen initiatief meer verantwoordelijkheid te nemen.

Toen ik vroeg naar feedback van mijn opdrachtgever en begeleiders was de reactie zeer positief. Ik was weliswaar begonnen met een slechte start maar had uiteindelijk een grote verbetering laten zien.

Kennis in de IT

Ik heb nog nooit in mijn carrière binnen de ICT zoveel nieuwe dingen bij kunnen leren binnen zulke korte tijd als dat ik het afgelopen half jaar heb gedaan. Toen ik bij Info Support begon kreeg ik echt gevoel in het diepe gegooid te zijn. Het voelde alsof ik qua kunde weer helemaal onderaan de ladder stond.

Het begon in de eerste weken van het afstuderen al met de trainingen aangaande het Microsoft framework MVC 3 en de taal Javascript in combinatie met de Javascript-library jQuery. Binnen deze trainingen werd in zeer rap tempo bijzonder veel kennis overgebracht. Na afloop weet je niet alle details meer uit het hoofd, maar hiervoor heb je een map met aantekeningen meegekregen vanuit de training zelf.

Eenmaal in elaboratie- en constructiefase kwam ik binnen mijn werkzaamheden allerlei technische uitdagingen tegen. Een deel hiervan heb ik zelfstandig kunnen oplossen met behulp van informatie op het internet. Er waren echter ook veel gevallen waar ik liever een van mijn collega's om hulp vroeg, onder andere dankzij hun ervaring met de stof. Op zulke momenten

bespraken we dan met enkele collega's, die in dezelfde ruimte werken, het probleem en probeerden gezamenlijk tot een oplossing te komen.

Wat ik een enorm pluspunt vindt binnen Info Support is dat mensen niet alleen kennis en ideeën uitwisselen wanneer er zich een probleem voordoet. Als iemand iets vindt, een techniek of toolkit waar hij of zij enthousiast over is, dan zal deze dit vaak delen met zijn directe collega's. Op deze manier leert iedereen bijna elke dag wel weer iets bij.

Wat ook veel kennis oplevert zijn de wekelijkse ISKA's. Dit zijn bijeenkomsten die iedere woensdagavond plaatsvinden. De avond begint een lopend buffet waarbij alle medewerkers mee kunnen eten. Vervolgens vinden er dan presentaties plaats. Deze worden gegeven door collega's die ideeën, kennis of hun voortgang binnen een project willen delen met de rest van hun collega's binnen Info Support.

Conclusie

Betreffende mijn voortgang het afgelopen half jaar kan ik concluderen dat ik geen beter bedrijf had kunnen uitkiezen voor mijn afstuderen. Binnen een half jaar heb ik in grote mate zowel kennis als ervaring opgedaan. Hiernaast heb ik ook een persoonlijke groei meegemaakt waarin ik zelfstandiger ben geworden en daardoor meer verantwoordelijkheid durf te nemen. Ik ben zeer tevreden over het werk dat ik heb afgeleverd en de voortgang die ik heb gemaakt binnen mijn afstuderen.

LITERATUURLIJST

Apple. Apple Developer Website. [Online] <https://developer.apple.com/>.

Bitovi. EmbeddedJS (EJS). [Online] <http://embeddedjs.com/>.

Info Support. Info Support. [Online] <http://www.infosupport.com/>.

—. Info Support Intraportal. [Online] <http://intraportal/default.aspx>.

—. Info Support Organisatiestructuur. [Online]
<http://carriere.infosupport.com/BVKwOrganisatie>.

—. ZorgInfo GAT (Gebruiker Acceptatie Test) omgeving. [Online]
<http://gat.zorgverstreking.infosupport.net>.

jQuery. jQuery. [Online] <http://jquery.com/>.

Microsoft. Kenniscentrum ASP.NET en daarbij horende frameworks. [Online]
<http://www.asp.net/>.

—. Visual Studio. [Online] <http://www.microsoft.com/visualstudio/nl-nl>.

Mono. Mono. [Online] http://www.mono-project.com/Main_Page.

—. MonoDevelop. [Online] <http://monodevelop.com/>.

Twitter. Twitter Bootstrap. [Online] <http://twitter.github.com/bootstrap/>.

Xamarin. Mono Touch. [Online] <http://xamarin.com/monotouch>.

BIJLAGE A: PLAN VAN AANPAK



BIJLAGE B: ONDERZOEKSDOCUMENT



BIJLAGE C: FUNCTIONEEL ONTWERP



BIJLAGE D: TECHNISCH ONTWERP

