



Afstudeerverslag

"Virtueel medicijnkastje voor een Android smartphone"

Lennart Schellingerhout
20060712
Versie 1.0

Referaat

Afstudeerder: L.C. Schellingerhout

Dit afstudeerverslag beschrijft het proces rondom de afstudeeropdracht 'Virtueel medicijnkastje voor de Android smartphone (Appoo)'.

Deze werkzaamheden hebben in de periode eind augustus 2010 tot begin januari 2011 plaatsgevonden.

Descriptoren	Google Android Java SQLite Scrum
Onderwijsinstelling	De Haagse Hogeschool Academie voor ICT en Media te Zoetermeer
Opleiding	Informatica voltijd
Studentnummer	20060712
Stageopdracht	Virtueel medicijnkastje voor een Android smartphone
Duur	31-08-2010 tot en met 07-01-2011
Organisatie	Infoprofs
Contactgegevens	Overschiestraat 184-H 1062 XK Amsterdam +31 (0)20 669 30 96 www.infoprofs.nl
Bedrijfsmentor	Martijn Noordeloos
Titulatuur	ING
Functie	Projectmanager
Contactgegevens	06-46321394 m.noordeloos@infoprofs.nl

Voorwoord

Dit is het afstudeerverslag van Lennart Schellingerhout, student Informatica aan de Haagse Hogeschool te Zoetermeer. In dit afstudeerverslag beschrijf ik het proces dat ik doorlopen heb tijdens mijn afstudeerperiode (september 2010 tot januari 2011) bij het bedrijf Infoprofs te Amsterdam.

Ik ben Infoprofs en Rick Dekker dankbaar voor het verschaffen van de afstudeeropdracht. Ook ben ik ze dankbaar voor de fijne samenwerking.

Ik wil van Infoprofs in het bijzonder de projectleider Martijn Noordeloos bedanken voor de geweldige begeleiding en het controleren van de door mij gemaakte documenten.

Als laatste wil ik mijn vader bedanken voor het taalkundig controleren van mijn verslag.

Veel leesplezier gewenst!
Lennart Schellingerhout,
Alphen aan den Rijn, januari 2011

Inhoudsopgave

Referaat	2
Voorwoord.....	3
Inleiding.....	5
1. Organisatie	6
1.1 Wat doet Infoprofs	6
1.2 Organisatiestructuur Infoprofs.....	7
1.3 Cultuur en werkomgeving	8
1.4 Projectleden + taken	10
1.5 Externe leden	11
2. Opdracht	12
2.1 Algehele projectopdracht	12
2.2 Individuele opdracht	12
2.3 Keuze methoden en technieken	13
2.4 Planning.....	19
2.5 Op te leveren producten	20
3. Voorbereiding.....	21
3.1 Concreet maken van de opdracht	21
3.2 Aanwezige producten	22
3.3 Keuze ontwikkelomgeving (IDE)	23
3.4 Werking van Android	26
3.5 Inwerken in Android	28
4. Uitvoering: Sprint 1	29
4.1 Systeemontwerp	30
4.2 Opzetten van een verbinding tussen de server en de smartphone....	32
4.3 Implementatie van de verbinding.....	34
4.4 Opstellen van het beveiligingsrapport	37
4.5 Resultaten uit het beveiligingsrapport	38
4.6 Implementatie van de resultaten uit het beveiligingsrapport	39
4.7 Testen.....	42
5. Uitvoering: Sprint 2	43
5.1 Opstartsessies voor Sprint 2.....	45
5.2 Backlog sprint 2.....	50
5.3 Systeemontwerp aanpassingen.....	51
5.4 Database ontwerpen en implementeren.....	52
5.5 Beveiligingsplan uitbreiden	56
5.6 Testen.....	56
6. Afronding	58
7. Evaluatie	59
7.1 Proces	59
7.2 Producten	62
7.3 Beroepstaken/competenties.....	65
7.4 Conclusie	69
Literatuurlijst	70
Verklarende woordenlijst	71

Inleiding

Dit document is een verslag over het proces dat ik doorlopen heb tijdens mijn afstudeerperiode bij Infoprofs te Amsterdam.

De opdracht is het ontwikkelen van een applicatie voor de Android smartphone. Deze applicatie kan medicijninformatie ophalen vanuit een database. Daarnaast moet de gebruiker van de applicatie ook zijn medicijngebruik bij kunnen houden binnen de applicatie. De applicatie heeft de naam Appoo gekregen.

Dit document begint met het beschrijven van de organisatie waar de afstudeeropdracht is gelopen. Hierin komt de geschiedenis van Infoprofs, de organisatiestructuur, de heersende cultuur en een beschrijving van de projectleden die betrokken zijn bij de ontwikkeling van de applicatie aan bod.

Vervolgens wordt de opdracht aangekaart. Hier worden als eerste de probleemstelling en doelstelling gegeven. Vervolgens worden de gebruikte methoden en technieken uitgelegd. Daarna volgt de planning voor het project. Het hoofdstuk sluit met het geven van de op te leveren producten.

In het derde hoofdstuk wordt de situatie voor het beginnen van het project beschreven. Ook komt de keuze voor de IDE (Integrated Development Environment) aan bod en wordt er een kleine uitleg gegeven over de werking van Android.

De hoofdstukken vier en vijf beschrijven het proces van het ontwikkelen van de applicatie.

Hoofdstuk zes beschrijft de afrondende fase van het project. Hier wordt onder andere beschreven hoe de opgeleverde producten zijn overgedragen aan Infoprofs.

In het laatste hoofdstuk evalueer ik de afstudeerperiode. Hierin zullen ook de vooraf gekozen beroepstaken/competenties aan bod komen.

1. Organisatie

In het hoofdstuk organisatie beschrijf ik hoe de organisatie eruit ziet waar ik mijn afstudeerproject heb uitgevoerd. Ik zal hierin uiteenzetten wat het bedrijf doet, hoe de organisatiestructuur er uit ziet, wat de heersende cultuur is en welke leden er bij het project betrokken zijn. Hiermee wil ik inzicht geven in de omgeving waarin ik terecht ben gekomen tijdens het afstuderen. Het is goed om te weten hoe deze omgeving in elkaar zit, om de rest van dit document in de juiste context te plaatsen.

1.1 Wat doet Infoprofs

Infoprofs is een softwarehuis dat is gespecialiseerd in ontwerp, realisatie en beheer van maatwerkoplossingen voor grote organisaties. Infoprofs ontwikkelt voornamelijk bedrijfsstrategische informatieconcepten eventueel gecombineerd met technologische vernieuwing. Ze biedt haar diensten aan vooraanstaande bedrijven binnen haar marktsegment. Infoprofs doet dit voornamelijk door het detacheren van specialisten. Naast detacheren heeft Infoprofs projectgroepen om innovatieve nieuwe projecten zelf te ontwikkelen.

Enkele bekende klanten van Infoprofs zijn:

- EPO (Europese patenten organisatie)
- ABN AMRO
- De Belastingdienst

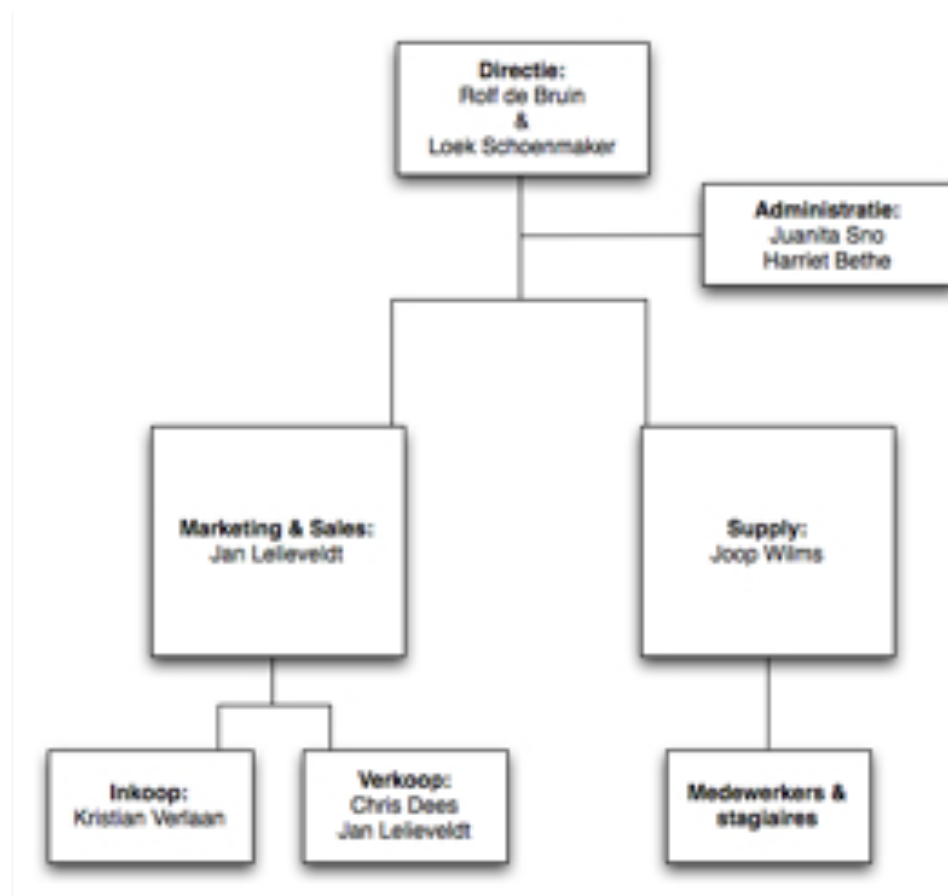
1.2 Organisatiestructuur Infoprofs

Het onderstaande diagram beschrijft de huidige situatie bij Infoprofs. De directie wordt gevormd door de twee oprichters van Infoprofs, namelijk Rolf de Bruin en Loek Schoenmaker.

Omdat Infoprofs zich specialiseert in detachering van IT'ers, is de organisatiestructuur daar ook op ingericht.

De administratie doet de administratieve taken van alle afdelingen. Marketing en Sales is verantwoordelijk voor het vinden van nieuwe opdrachten. De afdeling Inkoop houdt zich bezig met het werven van ZZP'ers. De afdeling Verkoop zoekt nieuwe opdrachten bij de klanten van Infoprofs. De afdeling Supply bevat alle medewerkers van Infoprofs die gedetacheerd kunnen worden. De taken van de afdeling Supply is het begeleiden van medewerkers in hun opdracht en in hun persoonlijke ontwikkeling. Ook werft de afdeling Supply nieuwe medewerkers.

Tot medewerkers en stagiaires worden de specialisten die gedetacheerd worden en de stagiaires (waaronder ik) gerekend.



Afbeelding 1.1: Organisatiestructuur van Infoprofs

1.3 Cultuur en werkomgeving

Wanneer men het kantoor bij Infoprofs binnenloopt, ziet men een moderne en verzorgde omgeving. Er zijn enkele vergaderruimtes die niet zijn gescheiden door muren, maar door glazen wanden. Naast de vergaderruimtes zijn er flexplekken om te kunnen werken. Een flexplek bestaat uit een bureau, een bureaustoel en een beeldscherm. Men heeft bij Infoprofs gekozen voor het gebruik van flexplekken, omdat men IT specialisten detacheert. Wanneer iemand niet gedetacheerd is, werkt deze op het kantoor van Infoprofs totdat deze weer een opdracht heeft. Aangezien dit aan veranderingen onderhevig is, is het verstandig om flexplekken te hebben waar gewerkt kan worden. De flexplekken zorgen er ook voor dat alle afdelingen in één ruimte werken. Dus zowel de directeuren, de administratie als het overige personeel werkt in dezelfde ruimte en werken allemaal op een flexplek. Dit zorgt voor een informele sfeer op de werkplek. Er wordt tussen personeelsleden geen hiërarchie gedefinieerd door de plek waar gewerkt wordt. Op de werkplek staat vaak zachtjes een radio aan om de stilte licht te doorbreken. Ook is er een hoekje waar een tv staat, die gebruikt kan worden om te 'socializen'. Verder is er nog een ruimte waar een espresso/koffiezetapparaat staat samen met enkele tafels en stoelen.

Er wordt van het personeel verwacht dat zij er netjes en verzorgd uit zien, maar men hoeft niet elke dag in 3-delig pak op kantoor te komen. Wanneer men gedetacheerd wordt bij een bedrijf, dan worden uiteraard de kledingvoorschriften van dat bedrijf aangehouden.

Op het kantoor zelf zijn er ook enkele ongeschreven sociale regels. Een voorbeeld hiervan is het halen van koffie of thee. Wanneer men koffie of thee gaat zetten, dan doet men dat niet voor zich alleen. Er wordt een rondje gemaakt langs alle andere aanwezigen of ze nog iets willen drinken. Een andere ongeschreven sociale regel is dat men zo tussen 8.30 uur en 10.00 uur verwacht wordt om te werken op kantoor. Af en toe is er sprake van een uitzondering, maar de meeste personeelsleden zijn rond deze tijden aanwezig.

Deze sociale manier van omgaan tussen collega's van alle afdelingen uit zich ook in de uitjes en speciale avonden die worden georganiseerd. Er is een feestcommissie die regelmatig een uitje voor het personeel organiseert. Ook zijn er enkele medewerkers die af en toe een gezellige pokeravond organiseren.

Deze sociale regels samen met de 'doorzichtige' vergaderruimtes en flexplekken, zorgen ervoor dat er een transparante werkwijze is en er een goede sociale sfeer wordt gecreëerd op het kantoor.

Naast het werken bij Infoprofs, werd er ook gewerkt in een qubical op de Academie voor ICT en Media te Zoetermeer. Binnen het deel Enterprize (bedrijven opgezet door studenten) van de school werd gewerkt aan de opdracht. Een qubical bij Enterprize bestaat uit twee bureau's met stoelen, twee whiteboards en twee kleine opbergkasten. Er heerst een rustige sfeer binnen Enterprize. Af en toe staat er wat muziek aan of hoor je vanuit een andere qubical wat geluid komen. Doordat er alleen maar studenten zijn binnen Enterprize, heerst er over het algemeen een erg relaxte sfeer. Wanneer er bij één van de bedrijfjes een klant op bezoek komt, dan heerst er meestal een zakelijke rust.

Er is gekozen om op twee locaties te werken om verscheidene redenen. De eerste is de beperkte ruimte binnen Infoprofs. Doordat Infoprofs een bedrijf is gespecialiseerd in detachering, is er niet voor iedere werknemer een eigen werkplek (zoals hierboven al beschreven is). Voor dit project zijn er drie studenten aangetrokken. Wanneer deze vijf dagen in de week een werkplek bezetten is er een tekort aan werkplekken. Daarnaast is de reistijd naar de academie in Zoetermeer korter dan die naar het kantoor in Amsterdam. Dit geldt met name voor Gunnar van Lit. De combinatie van deze twee factoren, plus het feit dat Infoprofs gewend is om mensen buiten het kantoor te stationeren, heeft er voor gezorgd dat er ook op de academie gewerkt wordt aan de opdracht. Op dinsdag en op vrijdag werd er standaard op het kantoor in Amsterdam gewerkt. De overige dagen in Zoetermeer. Er waren hierop uitzonderingen, bijvoorbeeld voor vergaderingen met de klant of wanneer er dringend overleg nodig was. De vele communicatie tussen het kantoor in Amsterdam en Zoetermeer verliep met name via de mail of via MSN Messenger. Ook zijn de klant, projectleider en de andere stagiair Koos van Marle geweest op de academie in Zoetermeer.



Afbeelding 1.2: Het maken van een videodemo in Enterprize

1.4 Projectleden + taken

Eén van de belangrijkste dingen bij het binnen komen van een nieuwe werkomgeving is het leren kennen van de collega's en/of projectleden. Hier volgt een beschrijving van de projectleden en hun taken:

Martijn Noordeloos, projectmanager:

Plant en coördineert het project. Hieronder vallen de zorg voor kwaliteitswaarborging (door het toepassen van een specifieke testmethodiek (TestFrame)), voortgangsbewaking, rapportage naar de klant en opdrachtgever toe en het beoordelen van de opgeleverde systeemmodules. Stelt daarnaast het testplan op.

Andries Evers, senior software engineer:

Was tot week 6 fulltime betrokken bij het project. Na week 6 parttime door detachering bij de Belastingdienst.

Is verantwoordelijk voor de backend server van het systeem en de database modules. Ook controleert hij of de kwaliteit van de code en documentatie van de junior software engineers voldoet aan de standaard die gehanteerd wordt bij Infoprofs.

Koos van Marle, onderzoeker ergonomisch design (stagiair):

Bepaalt door middel van onderzoek (o.a. interviews met mogelijke gebruikers) de ergonomie en de gebruikersinterface. Tevens probeert hij te achterhalen welke requirements de mogelijke gebruikers willen hebben in de applicatie.

Gunnar van Lit, junior software engineer (stagiair):

Is verantwoordelijk voor de implementatie van de gebruikersinterface en overige modules die voortvloeien uit het onderzoek door Koos van Marle.

Lennart Schellingerhout, junior software engineer (stagiair):

Is verantwoordelijke voor de communicatie van de smartphone met de backend, de database op de smartphone en de communicatie van de database richting de gebruikersinterface.

1.5 Externe leden

Rick Dekker, klant:

Is met het idee van de applicatie naar Infoprofs gestapt. Hij is de schakel tussen Infoprofs en de connecties binnen de medische wereld. Hij bepaalt ook volgens welk model dit project een financiële haalbaarheid heeft. Dit doet hij onder andere door op zoek te gaan naar extra sponsors.

Jan Lelieveldt, opdrachtgever:

De opdrachtgever van Infoprofs. De projectleider (Martijn) moet zich naar hem toe verantwoorden over de stand van zaken rond het project. Jan waarborgt of het project financieel nog verantwoord is.

Rolf de Bruin, sponsor:

Is naast financier van het project ook directeur van Infoprofs. Tijdens demonstraties van de applicatie geeft hij zijn mening over het gedane werk en adviseert hoe het project het beste kan worden voortgezet.

Loek Schoenmaker, sponsor:

Is naast financier van het project ook directeur van Infoprofs. Tijdens demonstraties van de applicatie geeft hij zijn mening over het gedane werk en adviseert hoe het project het beste kan worden voortgezet.

Rob Knijff, expert van Z-Index:

Is er om uitleg te geven over de structuur en inhoud van de G-Standaard (Z-Index database).

2. Opdracht

In dit hoofdstuk wil ik de opdracht naar voren brengen die uitgevoerd moet worden. Dit doe ik door middel van het noemen van de opdrachtoomschrijving, probleemstelling en doelstelling. Daarna volgt een uitleg van de gebruikte methodiek, de initiële planning en de op te leveren producten. Tijdens het bespreken van de uitvoer van de opdracht laat ik de planning weer terugkomen en bespreek ik welke aanpassingen er eventueel zijn geweest en waardoor deze zijn veroorzaakt.

2.1 Algehele projectopdracht

De opdracht is het ontwikkelen van een prototype Android applicatie, bestaande uit verschillende modules, voor het invoeren en beheren van medicatiedossiers. Hierbij is de medicijnspecificatie uit de G-Standaard (database met medicijninformatie) geïntegreerd met de noodzakelijke beveiligingsmaatregelen en een backend systeem.

Probleemstelling:

In de huidige zorgverlening wordt gebruik gemaakt van de G-Standaard. In deze G-Standaard staan alle medicijnen die op de Nederlandse markt verkrijgbaar zijn. De patiënt moet nu nog steeds een arts of apotheker raadplegen voor correcte informatie over medicijnen.

De patiënt kan niet zelf de informatie binnen de G-Standaard bekijken. Dit vergt veel tijd van zowel de patiënt als de arts/apotheker. De patiënt is geneigd om zelf op internet te zoeken naar informatie, maar de informatie die de patiënt aantreft is niet altijd juist of relevant. De G-Standaard is vrij beschikbaar, maar de gebruiker kan niets met de vorm waarin deze wordt aangeboden.

Doelstelling:

De doelstelling van dit project is het ondersteunen van patiënten door het verzorgen van een overzicht van hun medicijngebruik door middel van een applicatie op een Android mobiele smartphone.

2.2 Individuele opdracht

Individuele probleemstelling:

Er moet een manier gevonden worden waarop een Android smartphone kan communiceren met een backend server waar de G-standaard op aanwezig is. Daarnaast moet de gebruiker van de applicatie zijn eigen medicijngebruik bij kunnen houden. Er moet hierbij ook gekeken worden naar bijpassende beveiligingsmaatregelen.

Individuele doelstelling:

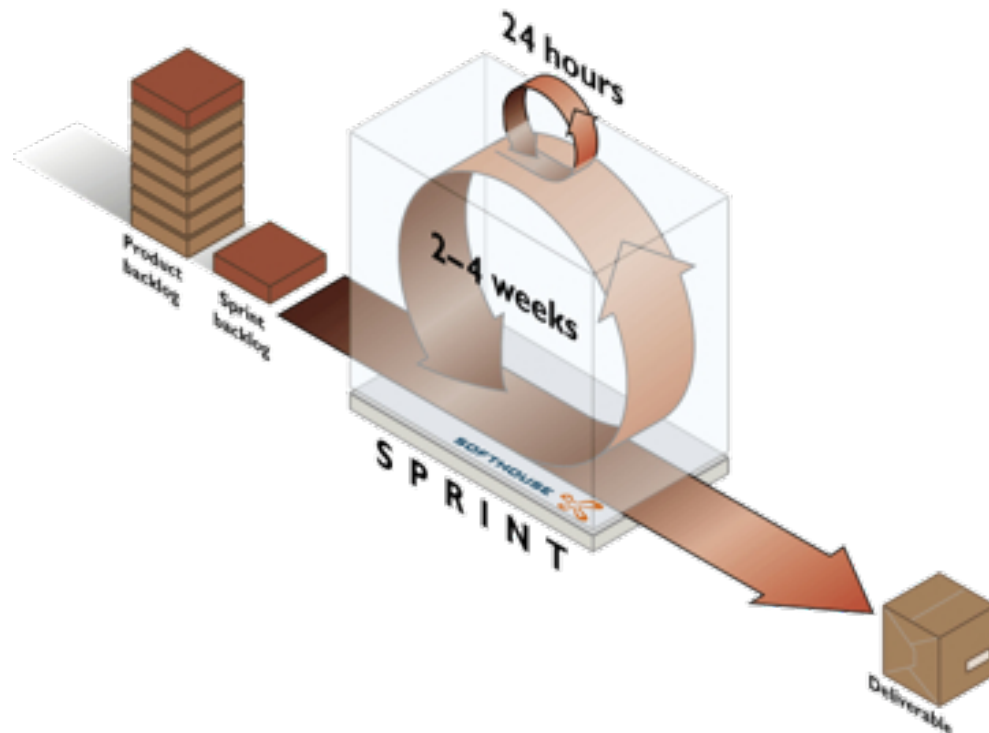
De Android smartphone moet gaan communiceren met een backend server waar medicijninformatie (G-Standaard) op staat. Deze gegevens plus eventuele persoonlijke gegevens (bijvoorbeeld hoeveel er van een bepaald medicijn wordt gebruikt) moeten in een database opgeslagen kunnen worden.

2.3 Keuze methoden en technieken

Tijdens een eerste sessie, voordat er begonnen was aan het afstuderen (zie hoofdstuk 3.1), is bepaald welke methodiek gehanteerd zou worden tijdens het project. Voor projecten als deze hanteert Infoprofs een op Scrum gebaseerde Agile methode. Er is door de projectleider, in overleg met de projectleden, gekozen voor deze methodiek.

Ik had voordat ik aan deze afstudeerstage begon nog niet gewerkt met Scrum, maar wel met een andere Agile methodiek: Feature Driven Development. Dit is een methodiek waarbij van te voren alle 'features' (onderdelen van het systeem die moeten worden gemaakt) worden vastgelegd. Een ontwikkelaar ontwikkelt één feature per keer, door deze eerst te ontwerpen, uit te voeren, te testen en te documenteren.

Scrum



Afbeelding 2.1: Overzicht van de Scrum methodiek

Bron: <http://ceswp.files.wordpress.com/2009/11/scrum-overview1.gif>

Bij Scrum worden eerst alle requirements opgesteld. Deze requirements worden ingeschaald op tijdsduur, complexiteit en prioriteit. Wanneer deze bepaald zijn, komen deze in de 'product backlog'. Vervolgens wordt een aantal requirements uit de product backlog gehaald en in de 'sprint backlog' gezet.

Aan de hand van de requirements uit de sprint backlog wordt een 'sprint' uitgevoerd. Een sprint is een periode (meestal twee tot vier weken) waarin wordt geprobeerd om zo ver mogelijk te komen in het uitwerken van de requirements. Dit levert een 'deliverable' (product) op. Aan het einde van een sprint worden de opgeleverde deliverables voorgelegd aan de opdrachtgever en/of klant. Deze geeft dan aan of de requirements goed zijn omgezet in deliverables. Wanneer dit niet het geval is, kan deze requirement weer opnieuw worden opgenomen in de backlog. Deze kan dan bij een volgende sprint weer meegenomen worden.

Voor een volgende sprint wordt er een selectie gemaakt uit de product backlog, welke geplaatst worden in de sprint backlog. Deze requirements worden weer binnen de bepaalde sprinttijd uitgevoerd. Zo kan men doorgaan tot alle (benodigde) requirements uit de product backlog zijn uitgevoerd en de klant/opdrachtgever tevreden is.

Door te ontwikkelen op deze wijze, kunnen er zoveel mogelijk requirements omgezet worden in een product (deliverable). Omdat het doel van deze opdracht is om een prototype neer te zetten, is dit een ideale manier van ontwikkelen. Nu kan er een, mogelijk niet helemaal perfect, prototype neergezet worden, waarmee nieuwe partijen/sponsors aangetrokken kunnen worden.

Bij deze methodiek is ook gekozen voor de meeting structuur 'Daily Scrum'. Aan het begin van de dag wordt door de projectleden ongeveer tien minuten besproken welke taken er uitgevoerd gaan worden. Aan het einde van de dag wordt gekeken welke taken afgerond zijn en welke problemen zich voorgedaan hebben bij het uitvoeren van de taken. Hierdoor houdt de projectleider een goed overzicht betreffende de status van het project.

Naast deze Daily Scrum zijn er ook meetings geweest met de opdrachtgever. Deze meetings duren langer en worden gebruikt om onder andere requirements op te stellen en om de opdrachtgever optimaal te betrekken bij het project. Deze meetings zijn soms alleen met de projectleider, maar vaak ook met de overige projectleden. Tijdens de uitwerking van de sprints later in dit document, zal dit uitgebreider ter sprake komen.

JTrac

Ter ondersteuning van de Scrum methodiek is er gebruik gemaakt van JTrac. JTrac is een open source issue tracker, een (in dit geval) webpagina waarin naast de backlog ook geconstateerde problemen gemeld kunnen worden.

Ieder projectlid heeft een account gekregen op deze issue tracker. Wanneer men tijdens het ontwikkelen tegen een probleem aan loopt kan men dit probleem melden in de issue tracker. Er wordt bij het melden van het probleem ook meteen vermeld wie het probleem heeft aangemeld en wie het probleem hoort op te lossen. Wanneer ik tijdens het ontwikkelen aanloop tegen een feature die (nog) niet of foutief aanwezig is op bijvoorbeeld de backend server, kan ik een melding aanmaken voor degene die de ontwikkeling van de backend doet. Deze ziet de melding en kan dan het probleem oplossen.

De JTrac bevat een actielijst, een backlog (toekomstige features) en een findingslog (problemen die gevonden zijn). De issue tracker zorgt er voor dat alle issues overzichtelijk bijgehouden worden. Dit is voor zowel de ontwikkelaars als de projectleider gunstig. De ontwikkelaars hebben een centrale plek waar ze overzichtelijk hun problemen neer kunnen zetten en kunnen zien welke problemen zij op moeten lossen. De projectleider heeft het voordeel dat hij op deze wijze een goed overzicht houdt op het project.










DASHBOARD

SEARCH

OPTIONS

LOGOUT

Lennart Schellingerhout

Space	Action	Status	Logged By Me	Assigned To Me	All	
Actielijstmedkast				1	1	7
Backlog_Medkast				0	0	12
MedkastFindingslog				1	1	5
			2	2	24	

Afbeelding 2.2: Screenshot van JTrac

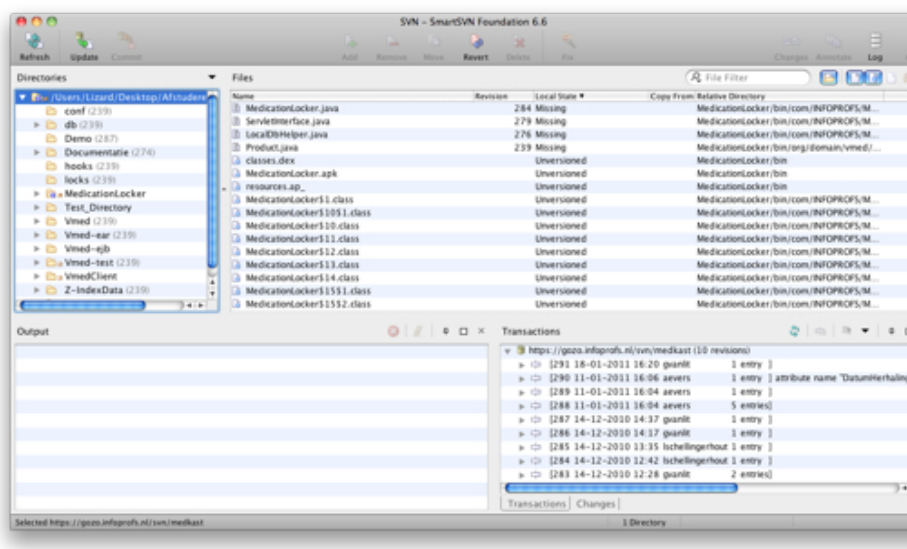
Op aandringen van de klant is tijdens de opstartsessies van het project (zie hoofdstuk 3.1) al van te voren globaal bepaald welke requirements in welke sprint uitgevoerd moeten worden. Dit wilde hij graag om een beter zicht te kunnen krijgen in wanneer hij welke requirements opgeleverd krijgt. De klant vond het vooral belangrijk om aan het einde van de stageperiode een werkende applicatie te hebben die hij kan gebruiken om geld los te krijgen bij nieuwe sponsors. Martijn heeft toen als projectleider besloten om op deze eis in te gaan. Ik vind het jammer dat ik hierdoor minder ervaren heb hoe de Scrum Agile methode echt werkt. Je wordt op deze manier toch meer tot een vaste planning gedwongen en kan daardoor ook eventueel geremd worden. Ik heb begrip voor het feit dat Martijn gehoor geeft aan de wensen van de klant. Nu het op deze manier verloopt geeft het mij verder wel meer duidelijkheid in de ontwikkeling van het project.

Subversie

Naast het gebruik van JTrac ter ondersteuning van Scrum, is er ook gebruik gemaakt van een subversiesysteem (SVN). Een SVN is een (deel van) een server die zo is ingericht dat men er content naar kan uploaden ('committen'). De content die is geupload wordt door het subversiesysteem gearchiveerd onder een subversienummer. Wanneer men een zelfde bestand upload die al op de SVN aanwezig is, krijgt deze een nieuw subversienummer. Het oude bestand blijft (niet zichtbaar) op de SVN staan onder zijn eigen subversienummer. Men kan dan te allen tijde een oudere versie van een bestand ophalen. Dit is met name handig wanneer in een later stadium blijkt dat het huidige bestand niet goed is en men verder wil met een oudere variant van het bestand. Tijdens het project wordt de SVN gebruikt om de gemaakte producten te uploaden ('committen'), zodat deze beschikbaar is voor alle projectleden. Alle projectleden krijgen dus net als bij JTrac een eigen account. Zo kan men in een logboek op de SVN bijhouden wie wat heeft geupload en op welk tijdstip dat plaats heeft gevonden.

We hebben als projectleden ook enkele regels afgesproken voor het committen van content op de SVN:

- De programmeurs mogen alleen een werkende built committen
- Tijdens het committen moet er altijd beschreven worden in het logboek welke wijzigingen er aangebracht zijn



Afbeelding 2.3: Screenshot van de SVN-applicatie SmartSVN

De data die te vinden zijn op de SVN:

- Alle documentatie die gemaakt wordt tijdens het project
- De code van de applicatie voor de smartphone
- De code van de backend server
- Testdata
- De backend database (G-Standaard)

Testen

Voor het testen is gebruik gemaakt van TestFrame. Dit is een methode die gekozen is door de projectleider. TestFrame is een open standaard, die ontwikkeld is door CMG in 1995 en tegenwoordig onderhouden wordt door Logica.

TestFrame gaat uit van drie principes:

- Het traject moet passen bij de organisatie en haar manier van systeemontwikkeling.
- Testen biedt alleen kwaliteitsbeheersing als het gestructureerd en onderhoudbaar wordt opgezet.
- Het gebruik van geautomatiseerde hulpmiddelen is onderdeel van een modern testproces.

Allereerst worden testscenario's opgezet. Deze testscenario's zijn onderverdeeld in functionele en technische scenario's.

In het functionele testscenario zijn de scenario's opgesteld die te maken hebben met de functionaliteiten die de applicatie biedt. Een voorbeeld hiervan is het zoeken van een bijsluiter van een medicijn. Er wordt in het scenario beschreven hoe het menu doorlopen moet worden om van een medicijn 'X' de bijsluiter te kunnen krijgen.

In het technische testscenario staan scenario's beschreven om te onderzoeken hoe de applicatie reageert wanneer deze fout wordt benaderd. Een voorbeeld hiervan is foute data invoeren of het ontbreken van een internetverbinding.

Deze testscenario's worden in een Testscript gezet. Het testscript is een Excel document, dat wordt gebruikt om de testresultaten overzichtelijk bij te kunnen houden.

Wanneer de tests uitgevoerd zijn, kunnen de resultaten ingevuld worden. Wanneer een test negatief uitvalt, doordat bijvoorbeeld een functionaliteit niet goed werkt, wordt dit gemeld in de JTrac (issue tracker). De verantwoordelijke voor de issue kan dan het probleem verhelpen. Wanneer men het probleem verholpen heeft kan men voor dat testscenario een tweede run draaien. Net zolang totdat de test positief uitvalt. Alle uitkomsten van de tests worden bijgehouden in het testscript.

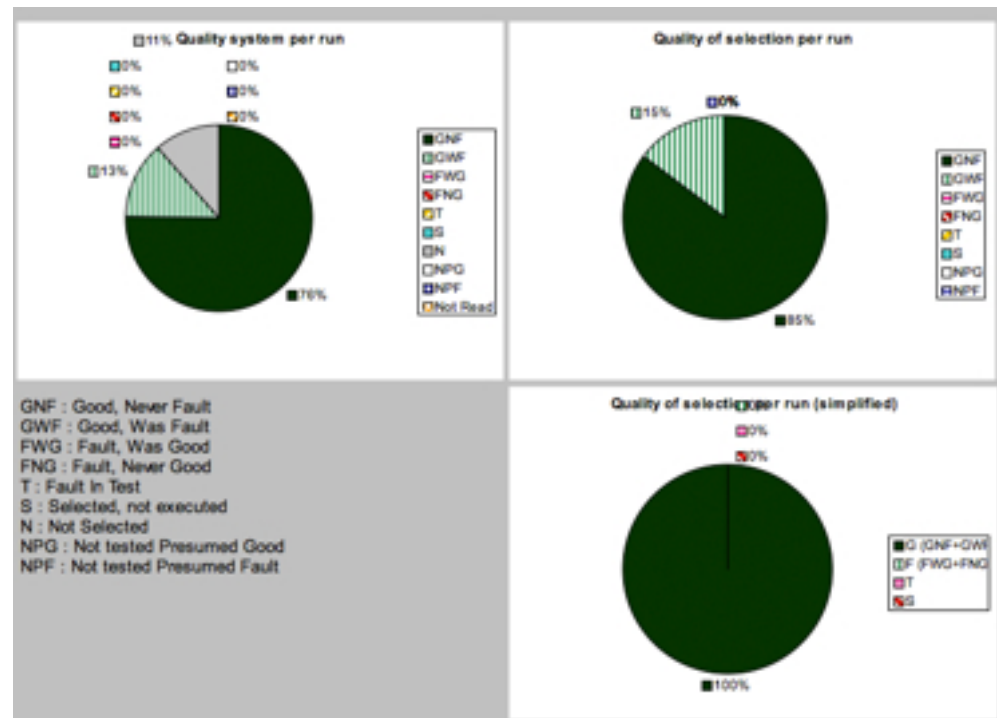
Het testscript kan nu tabellen en diagrammen weergeven. Een voorbeeld van een diagram dat gegenereerd wordt door het testscript is hieronder te vinden. Door de resultaten te visualiseren is in één oogopslag te zien of de kwaliteit van de software gewaarborgd is.

De vormen van testen die gebruikt worden zijn functionele testen, systeemtesten en unit testen.

Bij unit testen worden stukjes code los getest of deze correct werken. Hiervoor zal debugging gebruikt worden, om deze per stap in de code te controleren.

Ook zal er een gebruikerstest uitgevoerd worden door Koos van Marle. Hij gaat op de Haagse Hogeschool enkele mensen uitnodigen om met de applicatie te werken. Deze personen wordt de applicatie gegeven en ze moeten met minimale begeleiding deze proberen te doorlopen. Door deze sessies te filmen en de stappen na te kunnen kijken, kan er getest worden of de interface van de applicatie gebruikersvriendelijk is opgesteld.

Het testplan zal opgesteld worden door de projectleider Martijn Noordeloos. Dit testplan bevat de testscenario's voor de modules van alle ontwikkelaars. Deze zal deels gebaseerd zijn op de use cases die gemaakt zijn door Koos van Marle. Deze use cases vloeien voort uit het gebruikersonderzoek die hij uitvoert.



Afbeelding 2.4: Een voorbeeld waarin testresultaten worden weergegeven door middel van een taartdiagram

2.4 Planning

Zoals bij elk project is er ook bij dit project een planning opgesteld. Deze planning is een globale planning die na overleg met de klant gemaakt is (zie hoofdstuk 3.1). Dit is niet volledig conform SCRUM, maar is gedaan om de klant een beter overzicht te geven over wat hij kan verwachten aan het einde van de periode van 17 weken. Aan het begin van 'Fase 1' en 'Fase 2' wordt er nog een keer gekeken naar de planning en of deze wel haalbaar is. Zo niet, dan wordt deze aangepast. Ook kunnen er op die momenten eventueel andere functionaliteiten uit de backlog gehaald worden om te ontwikkelen. Uiteraard na goed overleg met en instemming van de klant.

Er is voor gekozen om voor de eerste fase een inwerkperiode te nemen van enkele weken. Ook wordt er tijdens die weken een plan van aanpak door ieder projectlid individueel opgesteld. Dit plan van aanpak zal grotendeels gebaseerd zijn op het 'werkplan' dat opgesteld wordt door de projectleider. In de losse fases worden de functionaliteiten uit de backlog ontwikkeld, getest en gerapporteerd.

	Week	1	2	3	4	5	6	7	8	9
Product										
Inwerk Periode										
Plan van aanpak										
Android verkennen										
Fase 1										
Systeemontwerp										
Verbinding smartphone met server										
Beveiliging smartphone										
Testen en rapporteren										
Verslag										
	Week	10	11	12	13	14	15	16	17	
Fase 2										
Database medicijnkastje implementeren										
Beveiliging medicijnkastje										
Testen en rapporteren										
Verslag										

Tabel 2.1: Initiële planning van het gehele project

2.5 Op te leveren producten

De producten die aan het einde van het project opgeleverd dienen te worden zijn:

- Plan van aanpak
- Systeemontwerp
- Een prototype Android-applicatie in modules
- Testrapport
- Beveiligingsplan

3. Voorbereiding

In dit hoofdstuk wil ik de voorbereiding van het afstuderen weergeven. Dit is het proces dat ik doorlopen heb voordat ik kon gaan beginnen met ontwikkelen. Hierbij moet men denken aan de eerste stappen die gezet zijn binnen Android en het kiezen van een IDE (Integrated Development Environment), maar ook de eerste meeting die gehouden is voor het begin van het afstuderen. Ook zal ik wat basisinformatie over Android uiteenzetten. Dit om meer context te kunnen bieden aan de opdracht waar ik aan gewerkt heb.

3.1 Concreet maken van de opdracht

Voordat er begonnen werd aan de stage, is er al een sessie gehouden om de opdracht concreet te maken. Deze sessie vond plaats op het kantoor van Infoprofs te Amsterdam.

Bij deze sessie waren de projectleden aanwezig, zoals genoemd in hoofdstuk 1.5 met uitzondering van Andries Evers die door vakantie afwezig was. Ook de klant Rick Dekker was aanwezig. Van hem komt het idee voor deze applicatie.

Tijdens deze sessie heeft Rick Dekker uitgelegd hoe volgens hem de applicatie er uit moet zien. Daarnaast heeft hij een korte uitleg gegeven over hoe het zorgsysteem in Nederland in elkaar zit. Hierin werd onder andere uiteen gezet hoe de relaties zijn tussen artsen, apotheken, distributeurs van medicijnen en medicijnfabrikanten. Ook werd er een beeld geschetst van de huidige situatie binnen de zorg en op welke punten er tekortkomingen zijn die opgelost kunnen worden door de applicatie. Hierbij gaat het vooral om de kennis die er is binnen de medische wereld, die lastig te bereiken is voor de gebruiker van medicijnen. Een belangrijk aspect voor de applicatie is dan ook dat het patiëntgedreven is. Hiermee wordt bedoeld dat er bij het ontwikkelen steeds gedacht moet worden uit het oogpunt van de patiënt. De patient moet dus centraal staan en niet een arts, apotheker, verzekeraar etc.

De eerste requirements zijn vervolgens opgesteld. Aan de hand daarvan zijn de taken verdeeld en heeft de projectleider een werkplan, met daarin onder andere de planning, opgesteld. Deze sessie is ook gebruikt voor het opstellen van mijn afstudeerplan.

3.2 Aanwezige producten

Bij aanvang van het afstuderen liep het project ongeveer twee weken. Koos van Marle heeft in die tijd onderzoek gedaan naar welke functionaliteiten er in de applicatie gewenst zijn. Hiervoor heeft hij interviews gehouden met mensen uit de doelgroep van de applicatie. De resultaten staan verwerkt in een rapport. Een deel van dit rapport is erg belangrijk voor Gunnar van Lit, die zich voornamelijk bezig houdt met de lay-out van de applicatie.

Een deel van het rapport beschrijft hoe de applicatie moet worden opgebouwd, zodat deze intuïtief werkt voor de gebruiker. Dit was niet het enige dat uit het rapport voortgevloeid is. In het rapport staat ook beschreven welke informatie getoond moet worden in de applicatie.

Deze informatie heeft Senior Software Engineer Andries Evers, die ook al begonnen was aan het project bij aanvang van het afstuderen, gebruikt om een deel van de backend server op te zetten. Toen ik aan de slag ging was al een deel van de backend server aanwezig. Er konden enkele zoekopdrachten afgehandeld worden door de server. Dit waren nog niet alle zoekopdrachten en tijdens het verdere verloop werden de overige zoekopdrachten geïmplementeerd.

Dit gaf mij het voordeel dat ik bijna meteen kon beginnen met het verbinden van de smartphone met de backend server. Het feit dat ik afhankelijk ben van het werk dat door andere projectleden gedaan moet worden, heeft er niet toe geleid dat ik het project begonnen ben met een achterstand. Hier was ik uiteraard erg blij mee.

3.3 Keuze ontwikkelomgeving (IDE)

De eerste keuze die gemaakt moet worden bij het werken met Android is de IDE (Integrated Development Environment) waarin gewerkt gaat worden.

Hiervoor heb ik een lijstje met voorwaarden opgesteld waaraan ik het liefst de IDE wil laten voldoen.

De IDE moet:

- Overweg kunnen met Java en de Android SDK
- Een Android applicatie kunnen emuleren
- De code kunnen debuggen en daarbij nuttige feedback kunnen geven.
- Een intuïtieve gebruikersinterface hebben
- Het liefst Open Source zijn

Er zijn enkele IDE's beschikbaar waarin ontwikkeld kan worden voor het Android besturingssysteem, die voldoen aan de bovengenoemde eisen, zoals Netbeans en Eclipse. Wanneer men zich enigszins verdiept in Android, komt men er snel achter dat Google zelf aanraadt om met de IDE Eclipse aan de slag te gaan.

De belangrijkste reden waarom Google zelf Eclipse aanraadt heeft waarschijnlijk te maken met de licentie die Eclipse hanteert. Eclipse heeft zijn eigen licentie opgezet (Eclipse Public License). Deze lijkt op het Common Public License, maar EPL heft enkele beperkingen die te maken hebben met patenten op. Dit zorgt ervoor dat EPL vrijer is dan CPL. NetBeans maakt gebruik van een 'dual license', namelijk Common Development and Distribution License en General Public License versie 2. Deze licenties bevatten meer beperkingen dan de EPL licentie. Maar zowel Eclipse als NetBeans zijn Open Source en voldoen dus aan die gestelde voorwaarde.

Waarschijnlijk wordt om deze reden plus de eenvoudige pluginfunctionaliteit, Eclipse door Google zelf aangeraden. Verder valt het ook op dat verreweg de meeste documentatie die te vinden is over ontwikkeling voor Android, specifiek is gericht op het gebruik binnen Eclipse.

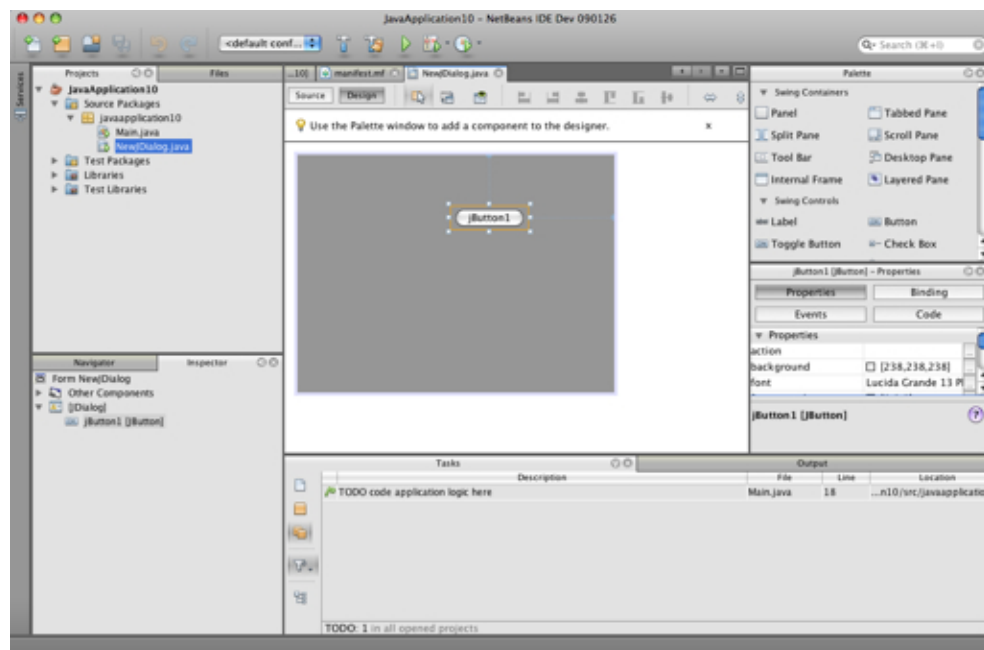
Wanneer de Android SDK (software development kit) geïnstalleerd is en toegewezen binnen Eclipse, met behulp van de Android SDK en AVD manager plugin, kan men de code compileren, debuggen en emuleren binnen een Android emulator die met de SDK meegeleverd wordt. Ook is er de mogelijkheid om een Android smartphone via USB aan te sluiten en direct op de smartphone de software te testen of te debuggen.

Het was nodig om de software op de smartphone te laten draaien, omdat niet alle functionaliteiten van een Android smartphone kunnen worden geëmuleerd door de emulator. Zo is het niet mogelijk om binnen de emulator gebruik te maken van een camera.

NetBeans heeft deze mogelijkheden ook, maar het installeren van de manager plugin vergt iets meer inspanning.

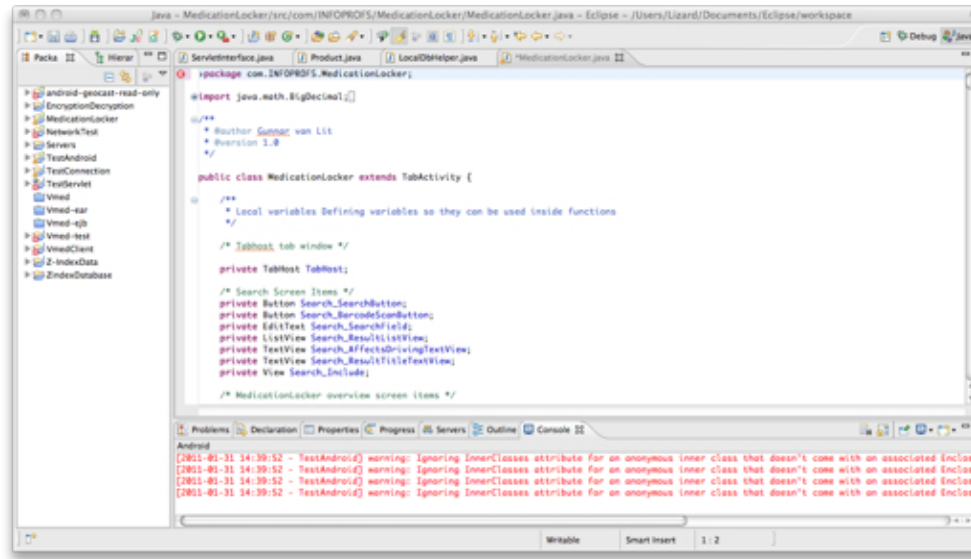
Ook de gebruikersinterface is erg belangrijk bij de keuze voor een IDE. Uiteindelijk moeten er toch vele dagen gewerkt worden binnen de omgeving. Het werk kan voor het gevoel aardig verlicht worden wanneer er sprake is van een goede intuïtieve gebruikersinterface.

De NetBeans interface is vrij donker qua kleur door het vele grijsgebruik. Hierdoor vind ik het op het eerste gezicht een vrij onduidelijke en 'depressief' ogende IDE. Er zijn vrij weinig knoppen onder de taakbalk te vinden. Hierdoor zijn er veel handelingen die moeten plaatsvinden via de taakbalk. Dit vergt meer tijd.



Afbeelding 3.1: NetBeans gebruikersinterface

De Eclipse omgeving is vrij licht qua opzet. De achtergronden van de verschillende frames hebben een witte achtergrond. Daarnaast zijn er veel, maar compacte icoontjes aanwezig. Dit zorgt ervoor dat er sneller veelgebruikte handelingen uitgevoerd kunnen worden. Het is ook vrij makkelijk om extra frames aan en uit te zetten, bijvoorbeeld de debug frames.



Afbeelding 3.2: Eclipse gebruikersinterface

Qua gebruikersinterface heeft Eclipse door zijn lichtgetinte opzet en de eenvoud van het aan- en uitzetten van frames mijn voorkeur.

Ondanks het feit dat ik zelf nog geen ervaring had met het werken in Eclipse en wel in Netbeans, heb ik vanwege de bovengenoemde argumenten toch gekozen voor Eclipse. Dit in overleg met de projectleden Gunnar van Lit en Andries Evers. Daardoor werkten we allemaal in dezelfde IDE.

3.4 Werking van Android

Het Android besturingssysteem en Android applicaties werken op een wijze waarmee ik nog niet bekend was. Dit was voor mij een aardige eye-opener. Het heeft een iets andere aanpak dan software voor een desktop computer. Dit is anders dan binnen de informaticaopleiding, waar de nadruk nog ligt op desktop applicaties en webontwikkeling (ASP.NET, PHP).

Om de komende hoofdstukken beter te kunnen begrijpen wil ik graag een 'basiscursus' Android geven. De informatie over de onderdelen van Android is te vinden op de officiële pagina voor Android ontwikkelaars: <http://developer.android.com>.

Een Android applicatie kan bestaan uit de volgende onderdelen: Activities, services, intents, broadcast receivers, content providers en de Manifest.XML. Deze wil ik hieronder graag toelichten.

Activity:

Binnen een Android applicatie zijn de activiteiten de bouwblokken. Elke activiteit die uitgevoerd kan worden is zijn eigen activity. Een activity is niet alleen een uit te voeren activiteit, maar bevat ook de grafische interface (view) die daar bij hoort. Dit kan gebeuren in een los XML bestand, maar ook binnen de code zelf.

Een applicatie kan andere activiteiten aanroepen door middel van een 'event', binnen Android een 'intent' genoemd. Voorbeelden van intents zijn: sms-functie, wekker, telefoneren, mail etc. Vanuit de applicatie wordt dan doorgeslagen naar de applicatie of een onderdeel van het besturingssysteem dat de handelingen verder afhandelt.



Afbeelding 3.3: Weergave van het uitvoeren van een intent.

Bron: http://developer.android.com/guide/practices/ui_guidelines/activity_task_design.html

Service:

Een applicatie of een activity zonder een grafische gebruikersinterface wordt een service genoemd. Hierdoor kan een gebruiker nooit direct een service aanroepen. De service kan aangeroepen worden via een intent uit een andere applicatie of activity. Een voorbeeld van een service zou een wekker kunnen zijn die een grafische interface aanroept wanneer de wekker af gaat. Hierbij wordt de wekker dan aangeroepen vanuit een andere applicatie of activity.

Overige onderdelen:

Broadcast receivers zijn er om hardware- of besturingssysteem-elementen te kunnen benaderen. Zo kunnen een camera, gps-systeem, maar ook de batterijduur of netwerkkwaliteit worden opgevraagd.

Content providers zijn interfaces naar opslag op de smartphone, zoals een geheugenkaart of database.

De *Manifest.xml* is een bestand waarin de structuur van de applicatie beschreven staat en hoe deze aangeroepen kan worden door andere applicaties.

In dit bestand staan onder andere de eisen voor installatie en de besturingssystemen waarmee de applicatie compatible is. Ook worden hierin de rechten bepaald die een applicatie heeft. Dit voornamelijk met betrekking tot de beveiliging die Android heeft. Zo moet er een uitzondering toegevoegd worden in het manifest bestand om gebruik te kunnen maken van het internet.

Een *toast* is een kleine pop-up die gebruikt kan worden om de gebruiker informatie te geven. Een voorbeeld is een pop-up die wordt getoond wanneer er geen verbinding gemaakt kan worden met het internet, terwijl een applicatie dit wel vraagt.

3.5 Inwerken in Android

Voor zowel mij als het bedrijf Infoprofs is Android onbekend terrein. Er was dus aan het begin van het project nog geen kennis aanwezig over deze omgeving. Android is gebaseerd op Java, een taal waarin men binnen Infoprofs gespecialiseerd is. De experts van Java binnen Infoprofs konden mij relatief gemakkelijk helpen als ik tegen problemen aanliep die te maken hadden met de Android code. Doordat er tijdens de Informatica-opleiding ook aandacht besteed is aan Java, was dit voor mij geen onbekend terrein.

De wijze waarop ik bij voorkeur te werk ga wanneer ik in aanraking kom met een applicatie/omgeving die onbekend voor mij is, is door mij in te lezen en tutorials uit te voeren. Vaak bouw je dan met behulp van een tutorial een eenvoudige applicatie, waarbij je de basisfunctionaliteiten van een omgeving leert kennen.

Omdat deze aanpak mij goed bevalt, heb ik deze methode ook toegepast bij dit project. Ik weet dat ik het beste leer door dingen uit te voeren en niet door bergen documentatie te lezen.

Op de website www.hackaday.com staan lessen waar in enkele delen een applicatie wordt ontwikkeld. Hierdoor maak je kennis met de wijze waarop Android functioneert (activities, services en broadcasts) en hoe je een gebruikersinterface opzet.

Ook kwam het ontwikkelen van een database op de Android smartphone naar voren in de tutorial. Dit was iets waar ik in de tweede sprint zeker profijt van had.

4. Uitvoering: Sprint 1

De eerste sprint van het project heeft een doorlooptijd van vijf weken. Voor deze eerste sprint zijn requirements uit de backlog geselecteerd. Hierbij hebben Gunnar, Martijn en ik overlegd wie welke requirements krijgt. Gunnar had een voorkeur voor de barcodescanner. Hij heeft deze toegewezen gekregen samen met de grafische gebruikersinterface. We hebben besloten om deze bij de barcodescanner te doen, omdat de barcodescanner voornamelijk te maken zou krijgen met de gebruikersinterface. Deze moet namelijk via een intent uit de grafische gebruikersinterface aangeroepen worden. De requirements die overbleven voor deze eerste sprint zijn: het opzetten van een verbinding tussen de Android smartphone en de server backend en het uitzoeken van passende beveiliging voor de applicatie. Daarnaast moesten Gunnar en ik samen het systeemontwerp opzetten voordat we aan de andere requirements konden beginnen.

Ondanks het feit dat ik werk aan requirements waar op het eerste oog weinig van terug te zien is, ben ik er toch tevreden mee. Er zit een grote uitdaging in beide aspecten. Vooral het beveiligingsaspect biedt een grote uitdaging, aangezien het hier gaat om persoonlijke gegevens met betrekking tot medicijngebruik. Iets wat momenteel een 'hot issue' is in verband met de invoering van het elektronische patiëntendossier. Ik ben vol goede moed en met enthousiasme deze sprint in gegaan.

De voor mij relevante onderdelen uit de sprint backlog:

- Systeemontwerp (samen met Gunnar)
- Communicatie tussen de smartphone en het backend systeem
- Beveiligingsrapport opstellen en de resultaten hiervan implementeren

Daarnaast was ik verantwoordelijk voor het testen, documenteren en rapporteren van de door mij opgeleverde functionaliteiten.

	Week	1	2	3	4	5	6	7	8	9
Product										
Fase 1										
Systeemontwerp										
Verbinding smartphone met server										
Beveiliging smartphone										
Testen en rapporteren										
Verslag										

Tabel 4.1: De planning voor sprint 1

4.1 Systeemontwerp

Het eerste wat gemaakt moest worden, voordat er geprogrammeerd kon worden, was het systeemontwerp. Met systeemontwerp bedoel ik het systeemontwerp van de applicatie op de smartphone. Het systeemontwerp van de backend server laat ik buiten beschouwing, aangezien zowel Gunnar als ik daar niet verantwoordelijk voor waren. Dit was de taak van Andries die de backend ontwikkelde.

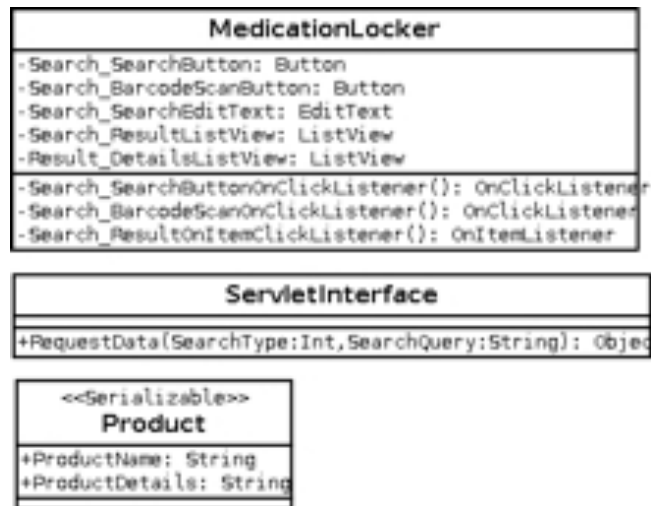
Allereerst hebben Gunnar en ik onze bevindingen die zijn opgedaan tijdens onze eerste stappen in Android gedeeld.

Dit vond ik erg nuttig en leerzaam. Door je eerste bevindingen met iemand te delen zorg je er voor dat je beiden nog meer leert over het systeem. We hebben elkaar een aantal aspecten van Android kunnen leren.

Een van de dingen waar Gunnar tegenaan was gelopen had te maken met het gebruik van activiteiten. Gunnar heeft er voor gekozen om de hele User Interface in één Activity te zetten. Zijn eerste idee was voor elk scherm een losse Activity, met daarin de functionaliteit. Het probleem daarbij was dat Activities niet met elkaar kunnen communiceren. Dit zorgt ervoor dat er geen reactie kan zijn van een scherm, wanneer er in een ander scherm een actie gedaan wordt. In het ontwerp is hier rekening mee gehouden.

Nadat we deze kennis en onze eerste ideeën voor het systeemontwerp hadden uitgewisseld hebben we overlegd hoe we dit ontwerp uit zouden werken. Al vrij snel kwamen we tot de conclusie dat we gebruik moesten maken van Unified Modelling Language (UML). UML hebben wij veel gebruikt tijdens onze studie en wordt al in een vroeg stadium aangeleerd op de opleiding. Het wordt ook veel gebruikt in het bedrijfsleven en Martijn en Andries waren eveneens bekend met het gebruik van UML. Zij begrepen onze diagrammen ook direct.

We hebben de functionaliteiten achter de grafische interface in losse klassen geïmplementeerd. De functionaliteiten uit deze klassen kunnen worden aangeroepen vanuit de grafische interface. De klassen van de grafische interface bevatten dus alleen de invoercontrole en de aansturing van de applicatie. De overige functionaliteiten zoals het leggen van een verbinding naar de backend zijn ontwikkeld in een losse service. Deze heeft daarom ook zijn eigen klasse gekregen.



Abbeelding 4.1: De klassen van het systeemontwerp

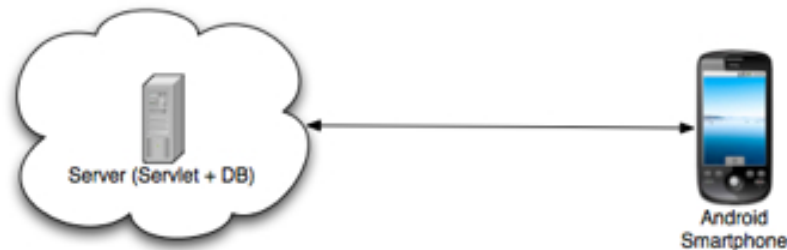
In het bovenstaande diagram staan de klassen die tijdens de eerste sprint ontwikkeld zijn. De klasse `MedicationLocker` was de verantwoordelijkheid van Gunnar. De twee andere klassen, `ServletInterface` en `Product` vielen onder mijn verantwoordelijkheid. In de `ServletInterface` wordt de connectie tussen de smartphone en de backend server bijgehouden. De `Product` klasse bevat het object dat vanaf de backend server naar de smartphone wordt gestuurd.

Ik was heel erg tevreden over de samenwerking met Gunnar. Vreemd is dit niet, want ik heb tijdens de opleiding al vele projecten met hem samengewerkt. Doordat we elkaar goed kennen zijn we op de hoogte van elkaars visie en werkwijze. Dit zorgde ervoor dat we in relatief korte tijd het eerste systeemontwerp aan Martijn ter beoordeling konden voorleggen.

4.2 Het opzetten van een verbinding tussen de server en de smartphone

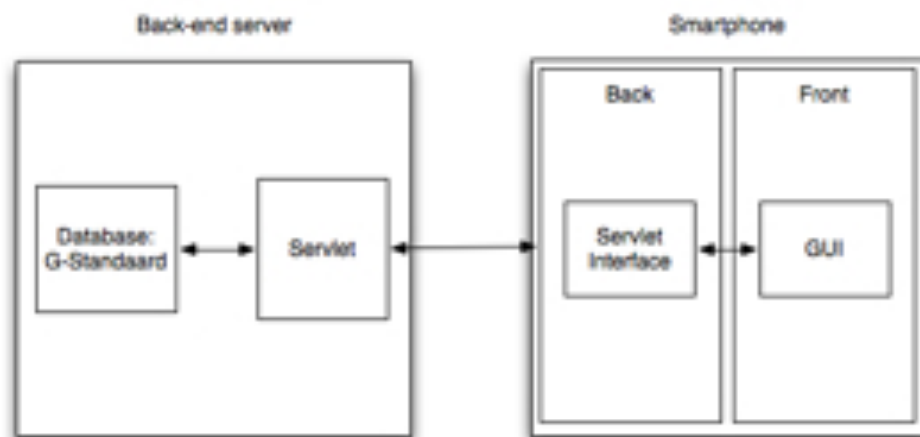
Op de server draait een servlet die connectie maakt met een MySQL database. In deze database staan de gegevens uit de G-Standaard. De G-Standaard is de database waarin alle medicijngegevens te vinden zijn van de medicijnen die verkrijgbaar zijn op de Nederlandse markt.

Iemand met de applicatie op zijn smartphone moet bijvoorbeeld kunnen zoeken op een medicijnnaam. De applicatie moet de zoekopdracht door kunnen sturen naar de servlet, die dan op zijn beurt de data uit de database haalt en terugstuurt naar de smartphone. De opgehaalde data moet dan goed weergegeven worden in de Graphical User Interface (GUI) van de smartphone applicatie. Mijn verantwoordelijkheid was om de verbinding tussen de smartphone en de server op te zetten.



Afbeelding 4.2: Verbinding opzetten tussen een Android smartphone en een backend server met een database

Het opzetten van de verbinding tussen de server en de smartphone zit in 'het midden' van het project. Hiermee bedoel ik dat wanneer het systeem in drie delen opgedeeld wordt (server backend, smartphone backend en smartphone frontend), de verbinding tussen de server en de smartphone tussen de server backend en de smartphone frontend (GUI) in staat. Het bevindt zich dus in de smartphone backend en wordt binnen het project 'ServletInterface' genoemd.



Afbeelding 4.3: Schematisch weergave van de frontend en de backend

Het proces van een zoekopdracht staat weergegeven in de onderstaande tabel.

Smartphone	Server
De gebruiker voert een zoekopdracht in via de GUI	
	De server zoekt de gevraagde data op in de database
	Het verkregen resultaat wordt teruggestuurd naar de smartphone
Toont het ontvangen resultaat in de GUI	

Tabel 4.2: Schematische weergave van een zoekopdracht

Ik had contact met zowel Gunnar (frontend smartphone) als Andries (backend server) om deze verbinding tot stand te kunnen brengen. De ergonomisch designer heeft, na uitgebreid onderzoek, bepaald hoe de applicatie er uit moet zien, welke data er getoond moet kunnen worden en op welke wijze deze getoond zal worden. Dit zorgt ervoor dat ik de ontvangen data vanaf de backend server op een bepaalde manier moet versturen naar de GUI. Ook moet de door de gebruiker ingevoerde data op een bepaalde manier naar de ServletInterface gestuurd worden, zodat deze richting de backend server kan.

Ik was bij het ontwikkelen van de verbinding afhankelijk van de wijze waarop de backend server is opgezet en hoe deze benaderd kan worden. Ik had dus intensief contact met Andries die de backend ontwikkelde.

4.3 Implementatie van de verbinding

Op het moment dat ik vertrouwd was geraakt met de Android omgeving en ik met Gunnar het systeemontwerp had gemaakt, ben ik gaan kijken hoe ik de smartphone kon laten communiceren met de backend. Op dat moment was de backend server nog niet zover ontwikkeld, dat ik er direct verbinding naar kon leggen. Deze was nog in ontwikkeling. Dit viel mij tegen, omdat dit betekende dat het mij extra werk zou kosten om een verbinding op te zetten. Maar om een gevleugelde uitspraak van Johan Cruyff aan te halen: 'Elk voordeel heb zijn nadeel en elk nadeel heb zijn voordeel'. Ook hier heeft het nadeel zijn voordelen gehad. Doordat ik zelf de hele omgeving heb opgezet in miniatuur, heb ik goed begrepen hoe deze functioneert. De kennis die ik hierbij heb opgedaan heb ik gebruikt bij het schrijven van het 'Beveiligingsplan'.

Het idee was om de backend op te zetten met behulp van JBoss SEAM. Kort gezegd zijn dit Java Server Pages (JSP) gecombineerd met Enterprise JavaBeans (EJB3). Deze draait op een Tomcat server. Dit is een aantal complexe termen, maar het verdiepen in deze termen is niet relevant om de rest van het document te kunnen begrijpen en zullen dus niet in detail uitgelegd worden. Binnen Infoprofs is voor een (externe) opdracht een keer dezelfde omgeving opgezet, dus is er intern genoeg documentatie en kennis aanwezig om deze goed op te kunnen zetten.

Omdat ik deze JBoss SEAM omgeving nog niet kon bereiken, omdat deze nog niet ver genoeg ontwikkeld was, heb ik zelf een miniatuurversie van deze omgeving opgezet.

Ik heb in mijn miniatuur omgeving zelf een MySQL database opgezet draaiend op een Tomcat server. Deze Tomcat server is te configureren binnen Eclipse. Op deze Tomcat server heb ik een eenvoudige servlet laten draaien die informatie ophaalt uit een eenvoudige MySQL database.

Na het opzetten van deze omgeving ben ik gaan onderzoeken op welke manieren er verbinding kon worden gelegd tussen een Android smartphone en een server. Na enkele voorbeelden en tutorials op het internet te hebben gevonden, ben ik tot de conclusie gekomen dat het beste gebruik kon worden gemaakt van de 'HttpClient' uit de Java Apache Jakarta Commons.

Dit is een library binnen Java die zich richt op de soort verbindingen die ik moest leggen (tussen een applicatie en een backend server). Hierbij moet vermeld worden dat de Tomcat server onderhuids gebruik maakt van de Apache serveromgeving.

De HTTPClient maakt gebruik van het HTTP protocol en gebruikt input en output streams om direct met een socket te kunnen praten. Hiermee kunnen direct objecten en lijsten (als array, hashmap, sorted map, dictionary, etc) verzonden en ontvangen worden. Daarnaast ondersteunt het de volledige implementatie van de HTTP methodes (Get, Post, Put, Delete, etc). Ook biedt het ondersteuning voor encryptie met HTTPS (SSL (Secure Socket Layer)). Dit is erg handig wanneer uit het beveiligingsplan blijkt dat er een wijze van encryptie gebruikt moet worden om data te beschermen. De HttpClient uit de Jakarta Commons wordt vanuit zichzelf al ondersteund door Android, dus deze hoeft niet los aan het project te worden toegevoegd.



Afbeelding 4.4: De HTTPClient

Vlak voordat ik deze omgeving werkend had, was Andries klaar met de basis van de backend server. Dit betekende dat hij in zijn eigen testomgeving data uit een database op kon halen via een servlet. Deze data kon via de servlet doorgegeven worden aan een losse Java klasse die hij had aangemaakt. Hiermee wordt het direct bevragen van de database volledig aan de kant van de server afgehandeld. Dit vermindert de druk aan de kant van de smartphone en er is een grotere mate van controle voor het opvragen van data. Foutieve zoekresultaten kunnen hiermee netter worden afgevangen en er gaat geen onnodige data over de verbinding richting de smartphone.

Ook Andries maakte gebruik van de HttpClient, waardoor onze implementaties erg op elkaar leken. Het enige dat ik nog aan mijn implementatie toe moest voegen waren de juiste 'headers' bij de inputstream. Deze headers bevatten de informatie die gestuurd moet worden naar de backend server, zodat deze met behulp van die informatie de database kan bevragen.

Daarnaast heb ik nog een 'serializable' klasse Product aangemaakt. Dit is het Product object dat van de server wordt verstuurd naar de smartphone. Deze klasse werd door mij onderhouden, maar ik had daarover nauw contact met Andries. Wanneer Andries een nieuwe functionaliteit op de backend server implementeerde, moest ik deze klasse aanpassen zodat deze weer goed werkte met de servlet. Op het moment dat ik deze klasse een update had gegeven, moest ik dit doorgeven aan Gunnar van Lit. Deze kon dan het nieuwe zoekresultaat gaan implementeren in GUI.

De samenwerking met Gunnar verliep, evenals bij het maken van het systeemontwerp, erg goed.

De samenwerking met Andries verliep ook prima. Andries is een rustig persoon en plezierig in de omgang. Wanneer er weer een onderdeel van de backend server af was of er was iets veranderd, dan gaf hij dat altijd correct aan mij door. Persoonlijk (wanneer wij beide op het kantoor in Amsterdam aanwezig waren) of via de mail. Wanneer ik vragen had of iets niet begreep, maakte hij altijd tijd om mijn vragen te beantwoorden en uitleg te geven. Dit is in mijn ogen ook hoe een senior software engineer hoort te handelen en ik heb veel van hem op kunnen steken. Het was ook mooi om te zien dat Andries open staat voor nieuwe ideeën. Wanneer ik een stukje implementatie had ontwikkeld waar hij nog niet bekend mee was, wilde hij altijd weten hoe het werkte. Dit om er in de toekomst eventueel zelf gebruik van te kunnen maken.

4.4 Opstellen van het beveiligingsrapport

Nadat de basis was gelegd voor een verbinding tussen de Android smartphone en de backend server, was het mijn taak om uit te zoeken welke beveiligingsmaatregelen er moesten worden genomen. Het was de bedoeling dat er ook in de prototype applicatie nagedacht zou zijn over de beveiliging van gegevens. Dit om nieuwe sponsors te laten zien dat er bij deze applicatie aan alle belangrijke aspecten wordt gedacht. Hier neemt de beveiliging ook een belangrijke plaats in, aangezien er gebruikt gemaakt wordt van privacygevoelige gegevens.

Hiervoor heb ik een 'Beveiligingsplan' opgesteld. Dit beveiligingsplan werd naarmate de sprints vorderden verder uitgewerkt.

Tijdens de eerste sprint werd er gekeken naar een 'veilige' dataoverdracht tussen de Android smartphone en de backend server. De resultaten van dit onderzoek zijn te lezen in het 'Beveiligingsrapport'. Ik zal echter de belangrijkste resultaten ook in dit hoofdstuk weergeven aangezien ze van belang zijn voor het beschrijven van de keuze van mijn oplossing. Het is echter verstandig om ook delen van dit rapport te lezen. Ik zal dan ook af en toe naar delen in dit rapport verwijzen.

Voor het opstellen van het beveiligingsrapport heb ik gebruik gemaakt van het boek 'Informatiebeveiliging onder controle' van Paul Overbeek, Edo Roos Lindgreen en Marcel Spruit.

De voornaamste reden om voor dit boek te kiezen is het feit dat de minor 'Informatiebeveiliging' gebruik maakt van dit boek.

Het boek raadt aan om eerst een risicoanalyse op te stellen voordat er maatregelen worden geïmplementeerd. Met deze risicoanalyse wordt bekeken welke informatie er beveiligd moet worden en op welke wijze dat het beste mogelijk is.

Uit de risicoanalyse is gekozen om een afhankelijkheidsanalyse en een maatregelenanalyse te doen.

4.5 Resultaten uit het beveiligingsrapport

Uit de afhankelijkheidsanalyse (pagina 7 uit het beveiligingsplan) is gebleken dat voor deze sprint, waarbij er alleen maar informatie opgezocht kan worden vanaf de smartphone, het zwaartepunt ligt op de integriteit van de data en de vertrouwelijkheid daarvan. De beschikbaarheid speelt in mindere mate mee, aangezien de applicatie nu alleen nog maar een ondersteunende functie heeft voor de gebruiker. Dit is het verschaffen van informatie, die men ook kan verkrijgen via andere kanalen.

De beveiligingsmaatregelen moeten zich voornamelijk concentreren op de integriteit en de vertrouwelijkheid.

In de maatregelanalyse (pagina 10 uit het beveiligingsplan) is gekeken naar welke maatregelen genomen kunnen worden om de integriteit en de vertrouwelijkheid te beveiligen. Hieruit zijn globaal gezien twee mogelijkheden gekomen: het gebruik van encryptie en het gebruik van SSL (Secure Socket Layer). Om tot een definitieve keuze te komen, is gekeken naar de voor- en nadelen van beide maatregelen (pagina 15 uit het beveiligingsplan). Door deze afweging te maken is gebleken dat SSL de meest volledige en het eenvoudigst te implementeren methode van encryptie is.

4.6 Implementatie van de resultaten uit het beveiligingsrapport

Nadat uit de risicoanalyse gebleken is dat SSL de beste maatregel is om de data over de verbinding tussen de smartphone en de server te beveiligen, heb ik gekeken hoe ik deze implementatie het beste aan kon pakken.

Allereerst ben ik gaan kijken welke SSL implementaties er te vinden zijn voor de Android smartphone. Hierbij heb ik de nadruk gelegd op het gebruik van self-signed certificates (pagina 13 uit het beveiligingsplan). Doordat het Android besturingssysteem nog niet zo lang bestaat, is de documentatie die hierover te vinden is vrij beperkt. Uit de documentatie die er was, kon vrij snel geconcludeerd worden dat Android van zichzelf geen goede implementatie voor SSL met self-signed certificates heeft.

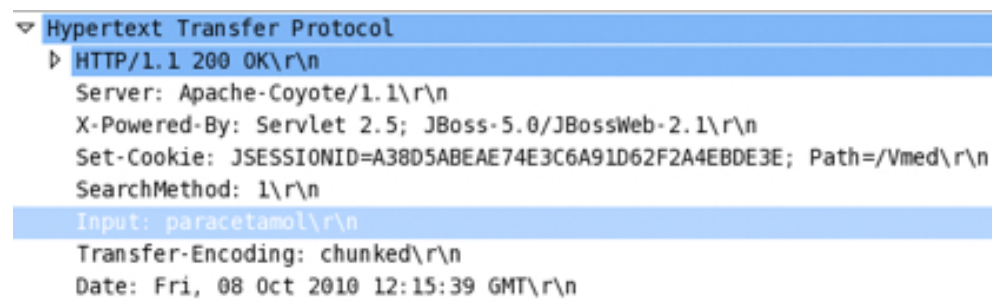
Ik kon dus niet vanuit de standaard Android documentatie halen op welke wijze ik het beste deze variant voor SSL kon implementeren. Gelukkig ben ik niet de eerste die tegen dit probleem aangelopen is, dus was er, hoewel schaars, informatie te vinden over hoe dit aangepakt kon worden. Er moet wel goed gekeken worden of de oplossingen die men biedt wel veilig zijn. Doordat er gebruik gemaakt werd van een self-signed certificate moet deze handmatig gecontroleerd worden. Enkele workarounds voor Android om toch gebruik te kunnen maken van SSL met self-signed certificates, zeggen dat elk certificaat geaccepteerd kan worden en zet dan de beveiligde SSL-tunnel op. Hiermee gaat men echter voorbij aan de authenticatie. Het kan zijn dat men verbindt naar een malafide server en daar een verbinding mee opzet, omdat er geen controle is van het certificaat. Dit heeft invloed op de kwaliteit van de integriteit en de vertrouwelijkheid van de data. Het was dus geen optie om deze implementaties toe te passen op de omgeving. Door onder andere te kijken naar open source applicaties die ook gebruik maken van self-signed certificates, kwam ik er achter dat een certificaat zelf gecontroleerd kan worden door gebruik te maken van de 'X509Trustmanager'. Deze trustmanager is te vinden in de javax.net.ssl library die ook standaard in de Android omgeving beschikbaar is. Met behulp van deze X509Trustmanger kan een certificaat worden gecontroleerd of het betrouwbaar is en geaccepteerd kan worden. Dit wordt eenmalig bij het opzetten van een verbinding (singleton) uitgevoerd.

Omdat de implementatie van SSL ook voor een deel plaats vindt aan de serverkant, heb ik hiervoor samengewerkt met Andries die verantwoordelijk is voor de serverkant. Ik heb uitgezocht hoe een certificaat geïnstalleerd kan worden op de server. Doordat er op de server gebruik gemaakt wordt van JBoss, was deze implementatie voor certificaten iets anders dan voor een standaard Tomcat server. Voor een standaard tomcat server wordt er een certificaat aangemaakt en zelf getekend.

Deze wordt op de server geplaatst en in de configuratie van Tomcat wordt dan aangegeven waar dit certificaat zich bevindt en welke poort deze voor SSL moet gaan gebruiken. Het verschil tussen de standaard Tomcat implementatie en de Jboss implementatie zit in de plaats waar het certificaat zich bevindt. Jboss maakt gebruik van een 'keystore' waar certificaten in opgeslagen worden. Er kan bij het opzetten van de keystore direct een certificaat gemaakt worden, die meteen toegevoegd wordt aan de keystore. Daarna kan er in de configuratie aangegeven worden waar deze keystore staat. Vanaf dit punt is het dus weer hetzelfde als een standaard Tomcat implementatie. Na dit uitgezocht te hebben, heb ik dit samen met Andries geïmplementeerd op de server. Deze samenwerking verliep prima en ik kon hem goede aanwijzingen geven hoe dit het beste geïmplementeerd kan worden. Hierbij heb ik hem ook uitgelegd hoe SSL werkt.

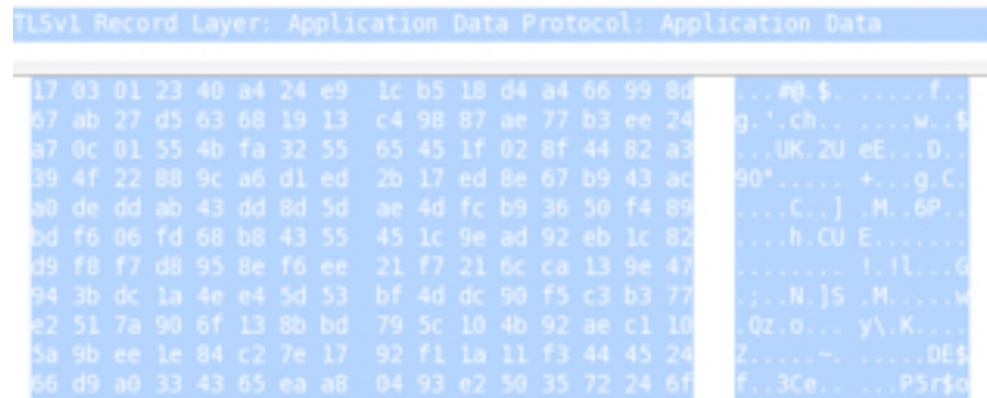
Nadat zowel de implementaties aan de server als de applicatie werkend waren, moest er getest worden of de verbinding werkelijk de SSL beveiliging had. Dit heb ik gedaan door het netwerkverkeer te analyseren met Wireshark. Wireshark is een applicatie waarmee netwerkverkeer bekeken en uitgelezen kan worden.

Met behulp van deze applicatie heb ik eerst een test gedaan waarbij er geen SSL gebruikt werd. Door het netwerkverkeer uit te lezen kon ik er achter komen op welk medicijn er gezocht werd door de applicatie.



Afbeelding 4.5: Een zoekopdracht zonder SSL

Door dezelfde test uit te voeren, alleen dan met het gebruik van SSL, kon ik controleren of de data niet meer te lezen was. Dit bleek inderdaad het geval te zijn, dus was de implementatie van SSL geslaagd. Meer over deze test is te vinden in het Beveiligingsplan onder het kopje 'Situatie na implementatie van SSL' (vanaf pagina 18 uit het beveiligingsplan). Hier wordt dieper ingegaan op de analyse van het netwerkverkeer.



Afbeelding 4.6: Niet leesbare data na de SSL impementatie

4.7 Testen

Voor het testen is gebruik gemaakt van TestFrame. Het testplan is opgesteld door de projectleider. Meer informatie over TestFrame en waarom de keuze hiervoor gemaakt is staat in hoofdstuk 2.3 van dit document.

Zoals al eerder vermeldt in dit document heeft Koos aan het einde van de sprint een gebruikerstest gehouden op de Haagse Hogeschool in Den Haag. De resultaten uit deze test waren positief. Er zijn wel enkele kleine aanpassingen gemaakt in de ergonomie en de gebruikersinterface.

Tijdens deze sprint heb ik technische testen uitgevoerd. Deze hebben te maken met het opzetten van de verbinding tussen de smartphone en de server. Door het feit dat ik tijdens het ontwikkelen veel testen heb uitgevoerd om te achterhalen of de gemaakte programmeercode goed functioneert, verliep het testen aan het einde van de sprint voorspoedig.

De aanpassing die gemaakt is na de eerste testrun is het toevoegen van een time-out functie. Wanneer de server niet door de smartphone bereikt kan worden, moet na een vastgelegde tijdsperiode een melding gegeven worden aan de gebruiker dat de applicatie de server niet kan bereiken.

Daarnaast heb ik ook de SSL implementatie getest. Deze test verliep naar wens. Ik was blij met de resultaten uit deze testen. Het toonde aan dat ik daadwerkelijk een beveiligde verbinding tussen de smartphone en de server op heb weten te zetten. De resultaten van deze testen zijn te vinden in het beveiligingsplan.

Test	Goed (%)	Fout (%)	Was fout, nu goed (%)	Was goed, nu fout(%)
Technisch run 1	87	13	0	0
Technisch run 2	100	0	13	0

Tabel 4.3: Testresultaten van de technische run in sprint 1

5 Uitvoering: Sprint 2

In dit hoofdstuk beschrijf ik het proces dat ik heb doorlopen tijdens sprint 2. In deze sprint zijn enkele veranderingen doorgevoerd. Deze zal ik uitvoerig aan bod laten komen.

Aan het einde van de eerste sprint is er een demo gegeven. Deze demo bestond uit een filmpje waarin de functionaliteiten die tijdens de eerste sprint zijn ontwikkeld werden getoond, waarna er vragen gesteld konden worden aan de projectleden. Toen dit had plaatsgevonden en er een duidelijk beeld was van de voortgang, werd er gediscussieerd over hoe er verder moest worden gegaan met de ontwikkeling van de applicatie.

Bij deze demo waren naast de projectleden, de klant, de twee sponsors/directeuren en de opdrachtgever aanwezig. Deze demo is gegeven om de vorderingen te laten zien en om feedback te krijgen op ons werk. Deze feedback kon dan meegenomen worden naar de tweede sprint.

De feedback die wij als projectgroep gekregen hebben tijdens deze demo was zowel positief als negatief. De positieve feedback was dat er veel werk verzet was betreffende de applicatie. De opdrachtgever en sponsors waren blij delen van de applicatie functionerend te zien. De klant had echter enigszins andere verwachtingen.

Dat de klant andere verwachtingen had dan hij zag tijdens de demo kwam voor een belangrijk deel door miscommunicatie tijdens de eerste sprint. Deze miscommunicatie zat vooral tussen de klant en de projectleider. De klant reageerde niet altijd op berichten van de projectleider en de projectleider heeft de klant niet regelmatig genoeg geraadpleegd.

De klant voelde zich niet voldoende betrokken bij 'zijn' project en wilde meer inspraak hebben. Een van de verwijten die hij maakte, was het feit dat er beslissingen zijn genomen zonder dat hij er van op de hoogte was.

Ik verbaasde me enigszins over deze uitspraak, aangezien er van tevoren globaal bepaald was wat er tijdens iedere sprint uitgevoerd zou worden. Dit was al een afwijking van de Scrum methodiek, die speciaal voor de klant gemaakt was.

Daarnaast zijn er tussendoor nog meetings met de klant geweest en daarin had hij niet voldoende door laten schemeren dat hij zich onvoldoende betrokken voelde bij het project.

Maar wanneer een klant zich niet voldoende betrokken voelt bij een project, moet daar verandering in komen. Wij hebben als projectleden met elkaar overlegd waar de fouten zijn gemaakt en waar we verbeteringen in aan konden brengen. Dit onder leiding van de projectleider. De projectleider heeft ook met de klant gesproken over hoe de communicatie en de betrokkenheid bij het project bevorderd konden worden.

Uit deze gesprekken zijn de volgende punten naar voren gekomen:

- De klant krijgt een meer directe inspraak in het project. Dit wil zeggen dat hij goedkeuring moet geven over onderdelen die gemaakt worden in de tweede en de daarop volgende sprints.
- De klant krijgt de rol van senior gebruiker.
- Er zullen meer meetings plaatsvinden met de klant.
- Tijdens de opstart van de tweede sprint zullen sessies gehouden worden waarin er naast onderdelen selecteren uit de backlog ook gekeken wordt naar de toekomst van de applicatie.

Deze punten hebben als nadeel dat de ontwikkeling van de applicatie trager verloopt. Ik vond dit erg jammer, want er was sprake van een gestage ontwikkeling en er werd veel progressie geboekt. Nu kwam er meer overleg dan nodig zou zijn in mijn ogen.

In het volgende hoofdstuk wordt dieper ingegaan op de sessies die gehouden zijn naar aanleiding van de demo. Er zal dan ook een aanpassing plaatsvinden in de planning die opgesteld is aan het begin van het project. De originele planning staat hieronder nog een keer weergegeven. In hoofdstuk 5.2 (backlog sprint 2) zal de aangepaste versie naar voren komen met enige uitleg bij de aanpassingen.

	Week	10	11	12	13	14	15	16	17	
Fase 2										
Database medicijnkastje implementeren										
Beveiliging medicijnkastje										
Testen en rapporteren										
Verslag										

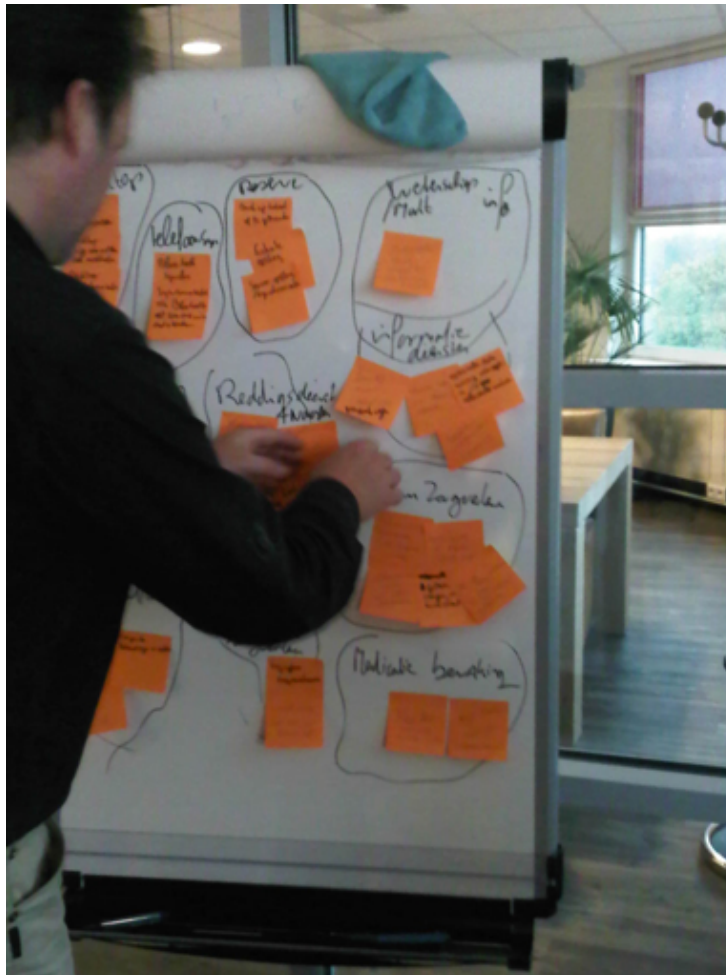
Tabel 5.1: De initiële planning voor sprint 2

5.1 Opstartsessies voor Sprint 2

Zoals genoemd zijn er bij de opstart van sprint 2 extra sessies gehouden om de klant meer te betrekken bij het project. Hierin zou hij meer sturing en feedback kunnen geven. Ik wil hier graag verder ingaan op het verloop van deze sessies.

Sessie één

De klant wilde graag in de eerste sessie met de projectleden brainstormen over de applicatie. Hierbij wilde hij geen grenzen stellen. Dus alle ideeën die in ons opkwamen met betrekking tot de applicatie mochten wij aangeven. Deze ideeën hebben wij op Post-It blaadjes geschreven en op een whiteboard bevestigd. Op het whiteboard werden alle ideeën die onder dezelfde categorie vallen bij elkaar gezet. Uit deze categorieën kwamen featuresets. Deze featuresets bestaan uit onderdelen die mogelijk door een ontwikkelaar gemaakt kunnen worden.



Afbeelding 5.1: Het sorteren van de ideeën op een whiteboard

Nadat deze featuresets gemaakt zijn, hebben wij ze gescoord door ze een gewicht mee te geven. Hoe hoger het getal dat gegeven wordt aan een featureset, hoe moeilijker deze te implementeren is.

Dit scoren is gedaan met behulp van spelkaarten. Iedereen kreeg een set kaarten die genummerd waren van 2 tot en met 10. Er werd een featureset naar voren gebracht en iedereen bepaalde voor zichzelf het 'gewicht' van deze featureset op bovengenoemde schaal. Van het nummer dat men koos, pakte men de kaart en liet deze tegelijk met de andere projectleden zien. Wanneer het verschil in gewicht groot was, werd er beargumenteerd waarom men gekozen had voor dat getal.

Ik vond dit een erg goede methode om weer een eenheid te creëren tussen de projectgroep en de klant. Het heeft de klant een goed inzicht gegeven in onze werkwijze en de afwegingen die wij gemaakt hebben tijdens de eerste sprint. Over het algemeen schatte de klant de complexiteit van featuresets minder zwaar in dan de projectleden. Wij als projectleden konden dan over de featureset aangeven waar wij de obstakels zagen. Meestal stemde de klant in met onze afwegingen.

Na het 'wegen' van de featuresets werd gekeken welke belangrijk waren om uit te voeren voor het prototype. Deze featuresets werden dan in de tweede sprint (zo mogelijk) uitgevoerd.



Afbeelding 5.2: Het uittekenen van de features voor het prototype

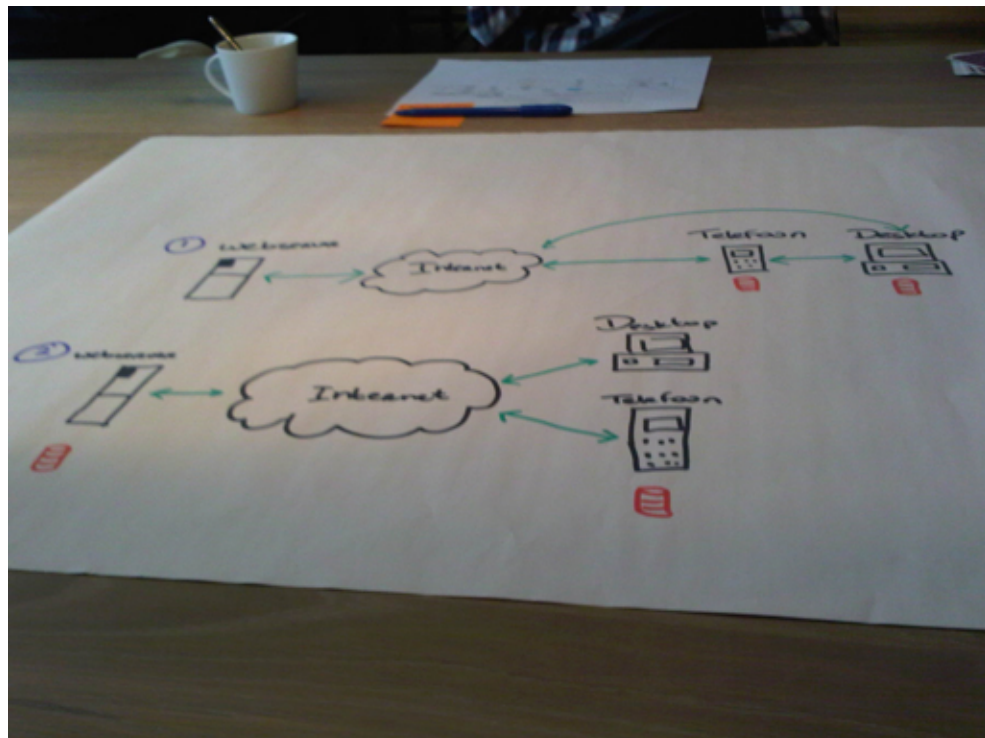
Naast het bepalen van de features voor het prototype wilde de klant ook een keuze maken betreffende de architectuur, waarop in de toekomst verder gebouwd kan worden. Er moest hiervoor buiten het prototype gedacht worden. De focus hierbij lag op de plek waar de patiëntgegevens opgeslagen moesten worden: op de backend server of op de smartphone zelf. Beide mogelijkheden leggen het zwaartepunt op een ander deel binnen de architectuur.

Wanneer men er voor kiest om alle patiëntgegevens op te slaan aan de serverkant, komt er een zeer grote hoeveelheid beveiligingseisen om de hoek kijken. Deze eisen zijn te vergelijken met die van een bank of het elektronische patiëntendossier. Een ander nadeel is de grote hoeveelheid dataverkeer die gestuurd moet worden over de lijn tussen de server en de smartphone.

Het grote voordeel van deze architectuur is dat de data beschikbaar is op elke smartphone die de applicatie draait.

Wanneer men er voor kiest de gegevens op te slaan op de smartphone, ligt de verantwoordelijkheid van de gegevens bij de eigenaar van de smartphone. Dit scheelt aanzienlijk in de beveiligingseisen en verantwoordelijkheid die bij de ontwikkelaar komt te liggen.

Een nadeel hiervan is dat de data maar op één smartphone staat. Wil men het op meerdere smartphones plaatsen, dan moeten er synchronisatie-oplossingen bedacht worden.

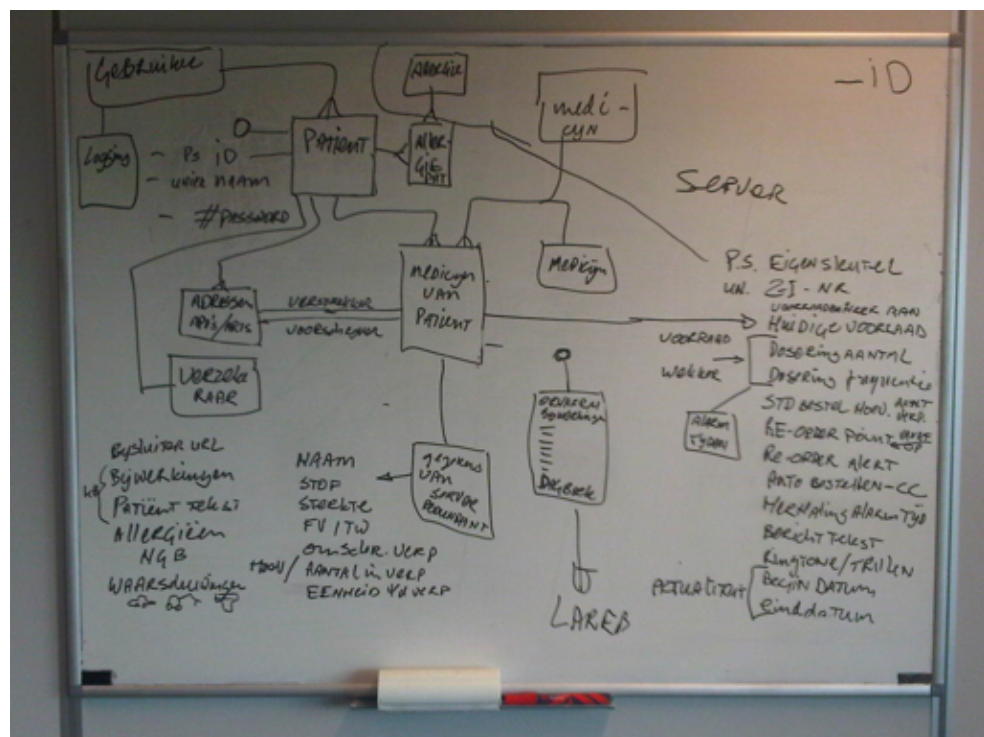


Afbeelding 5.3: De twee verschillende architecturen bij de opslag van patiëntgegevens

De klant had voor deze sessie de voorkeur voor de architectuur waarbij de patiëntgegevens opgeslagen worden aan de serverkant. Dit vanwege de mogelijkheid om het op elke smartphone en computer bereikbaar te laten zijn. Hij had zich echter niet gerealiseerd dat het een kostbare onderneming is om aan alle beveiligingseisen te voldoen. Uiteindelijk heeft hij er voor gekozen om de opslag van de patiëntgegevens aan de kant van de smartphone te plaatsen.

Sessie twee

Er is op aanvraag van de klant een tweede sessie gehouden. De patiëntgegevens moesten worden opgeslagen in een database op de smartphone en hij vond de doorlooptijd voor het ontwerpen en uitwerken van de database te lang ingeschat. Hij heeft zelf jarenlang databases ontwikkeld en wilde samen met de projectleden bedenken welke informatie er in de database moest komen. Aan de hand van deze informatie wilde hij samen met ons een databaseontwerp maken.



Afbeelding 5.4: Een eerste schets van het databaseontwerp

Deze sessie is enkele dagen later voortgezet. Nu was ook de projectleider aanwezig. Bij het verder uitwerken kwam er een conflict tussen de klant en de projectleider naar voren over hoe het ontwerp gemaakt moest worden. Dit leidde tot een vrij stroeve sessie.

Tijdens deze sessie is wel een eerste ontwerp van de data dictionary gemaakt.

Er is met de projectleider nog een derde sessie gehouden om het voorlopige ontwerp te normaliseren. Tijdens deze sessie zijn er tussentabellen toegevoegd in het ontwerp, waardoor een eerste versie van de database ontwikkeld kon worden. Ik heb daarna het volledige model en de data dictionary uitgewerkt. Aan de hand van deze informatie heb ik later de database opgezet. Hoe het ontwikkelen van de database verlopen is, komt later aan bod.

Aan de ene kant vond ik het goed om deze tweede sessie met de klant te hebben, maar aan de andere kant vond ik het ook nadelen hebben. Het voordeel kwam met name naar voren tijdens het eerste deel van deze sessie. Het globale ontwerp werd daar gemaakt door goed met elkaar te praten en te beargumenteren wat er wel en niet in moest komen. Dit zou voor mij een goede basis zijn geweest om verder te gaan met het ontwerp van de database. De klant wilde echter zelf ook verder werken aan het ontwerp in een tweede deel van deze sessie. Hierbij kwamen toch verschillen van inzicht naar voren waardoor deze sessie moeizaam verliep. Na dit deel van de sessie werd er nog een derde deel gehouden om de normalisatieslagen toe te passen. Deze laatste twee delen had ik zelf ook kunnen doen en dan mijn ontwerp voor kunnen leggen aan de klant en de projectleider. Die hadden daar hun feedback op kunnen geven. Dit is in mijn ogen een snellere manier van werken. Deze extra sessiedelen hebben veel tijd gekost. In totaal was de doorlooptijd van deze tweede sessie anderhalve week en namen deze drie werkdagen in beslag.

5.2 Backlog sprint 2

Doordat er enkele aanpassingen zijn geweest in de methodiek en er extra sessies zijn geweest met de klant, zijn er wijzigingen gekomen in de projectplanning en de sprint backlog. Door de extra tijd die de opstartsessies in beslag namen, kon er helaas minder opgeleverd worden dan van te voren was gepland.

Tijdens de tweede sprint werd het medicijnkastje gemaakt. In het medicijnkastje kan een patiënt opslaan welke medicijnen hij of zij slikt. Naast de medicijnsoort, kan men ook de frequentie en dosis van het toedienen van een medicijn aangeven. Dit wordt gebruikt voor het voorraadbeheer en voor een medicijnwekker. Deze wekker gaat af wanneer de patiënt het medicijn moet toedienen.

De medicijnkast heeft de vorm van een database die zich bevindt op de smartphone. Deze medicijnkast moet kunnen worden toegevoegd, verwijderd, getoond en gewijzigd.

De voor mij relevante onderdelen uit de sprint backlog:

- Systeemontwerp aanpassen met de medicijnkast(samen met Gunnar)
- Het ontwerpen en implementeren van de database op de smartphone
- Beveiligingsrapport bijwerken en advies geven over de implementatie

Daarnaast was ik verantwoordelijk voor het testen, documenteren en rapporteren van de door mij opgeleverde functionaliteiten.

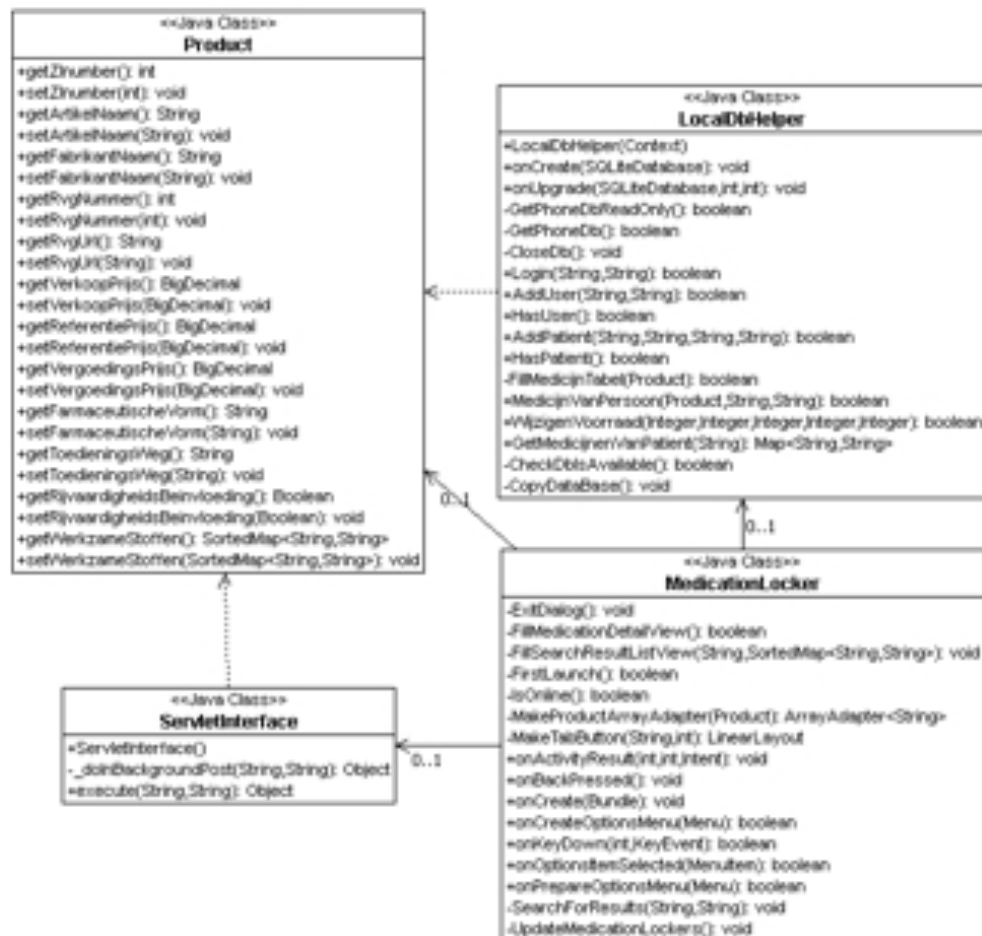
Omdat er extra tijd geïnvesteerd werd in overleg met de klant, was het plan om voor de beveiliging alleen een advies te geven aan de hand van onderzoek. Dit wordt beschreven in het beveiligingsplan. Dit verlicht de lasten voor een groot deel. Er wordt echter meer aandacht besteed aan de database. Deze bevatte niet alleen tabellen die bij sprint 2 hoorden, maar ook voor latere sprints. Dit omdat de klant graag de gehele database klaar wilde hebben aan het einde van deze sprint.

	Week	10	11	12	13	14	15	16	17	
Fase 2										
Demo										
Sessies met de klant										
Database medicijnkastje implementeren										
Beveiliging medicijnkastje										
Testen en rapporteren										
Verslag										

Tabel 5.2: De aangepaste planning

5.3 Systeemontwerp aanpassingen

Met de komst van het medicijnkastje moest ook het systeemontwerp worden aangepast. Er is een extra klasse gekomen die de verbinding met de database afhandelt. Meer over de werking van deze toegevoegde klasse is te lezen in het volgende hoofdstuk (5.5), waarin ik het ontwerpen en implementeren van de database bespreek.



Afbeelding 5.5: Systeemontwerp sprint 2

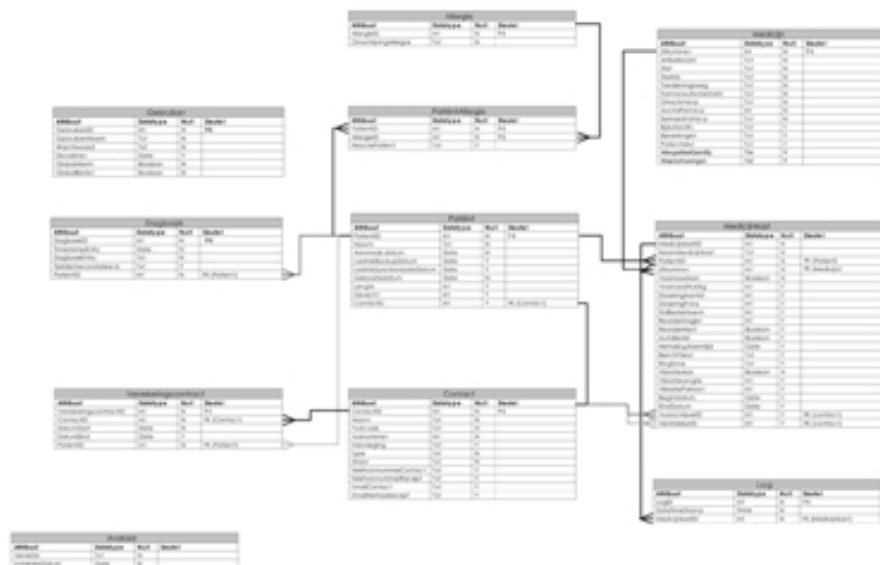
5.4 Database ontwerpen en implementeren

Ik heb bij het bespreken van de tweede sessie met de klant al aandacht geschonken aan de medicijnkast database die geïmplementeerd moest worden op de smartphone. In dit hoofdstuk wil ik verder gaan met de beschrijving van mijn aanpak.

Ik heb voor het ontwerpen van de database gebruik gemaakt van het boek 'Principes van databases' geschreven door Guy De Tré. Binnen de opleiding wordt gebruik gemaakt van dit boek en het biedt een goed inzicht in het ontwerpen van relationele databases.

Na de tweede sessie met de klant en alle projectleden heb ik het ontwerp van de database en data dictionary verder uitgewerkt. De klant wilde graag een database hebben die in latere sprints niet veel verder hoeft te worden uitgewerkt. Hierdoor is deze database veel groter geworden dan nodig is voor het prototype. Dit geldt uiteraard ook voor de data dictionary.

In het ontwerp zijn onder andere contactgegevens, een dagboek, verzekeringscontract, allergieën, logboek, bestelinformatie, etc opgenomen. Dit zijn onderdelen die mogelijk in toekomstige sprints geïmplementeerd zullen worden.



Afbeelding 5.6: De verschillende tabellen van de database. Het ontwerp op volledige grootte zit als bijlage toegevoegd.

Het grotere formaat van de database zorgde voor meer werk. Er moest veel meer nagedacht worden over het ontwerp van de database, dan wanneer er alleen een database moest worden ontworpen waarin een patiënt de medicijnen op kan slaan die toegediend moeten worden. Er was nu wel een basis voor een goede database, maar bij volgende sprints zullen er altijd aanpassingen gemaakt moeten worden. Ondanks de vele uren denkwerk die in het ontwerp van de database zitten, zullen er ongetwijfeld nog elementen over het hoofd gezien zijn. Hierdoor heb ik het gevoel dat ik veel overbodig werk heb gedaan.

Keuze databasemanagementsysteem

Nadat het databaseontwerp en de data dictionary waren gemaakt moest de database worden gebouwd. Om deze te kunnen bouwen moet er eerst een keuze gemaakt worden voor een databasemanagementsysteem.

Tijdens het inwerken in Android heb ik een tutorial gevolgd waarbij gebruik gemaakt werd van het databasemanagementsysteem SQLite. Ik heb onderzocht wat SQLite is en waarom er bij die tutorial gebruik gemaakt is van SQLite.

Wanneer men zoekt op SQLite in combinatie met Android komt men al snel op de developer pagina's van Android. Android zelf raadt het gebruik van SQLite aan. Verder valt op dat (bijna) elke applicatie, die gebruik maakt van een database, SQLite gebruikt. Ik heb vervolgens onderzocht wat de voor- en nadelen zijn van dit databasemanagementsysteem. Dit om te bepalen of het verstandig was dit systeem te gebruiken.

SQLite is een relatief klein databasemanagementsysteem dat zijn oorsprong vindt in de US Navy. Men wilde dat de software van een raketsturingssysteem niet gebruik hoefde te maken van een los databasemanagementsysteem. Men wilde deze geïntegreerd hebben binnen de software.

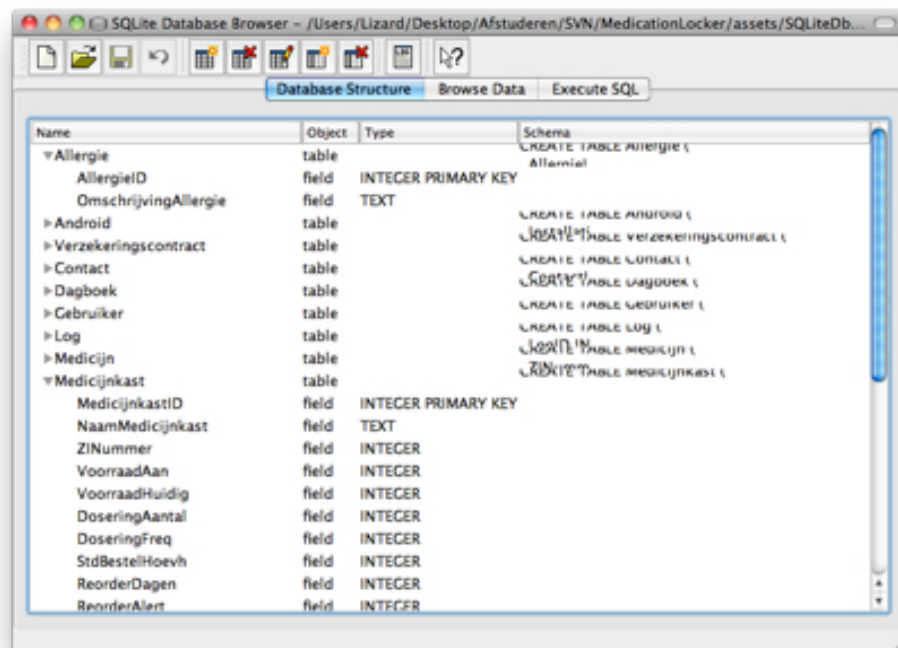
Hetzelfde principe kan ook gelden voor een database op een smartphone. Het is vrij lastig voor zowel de gebruiker als de ontwikkelaar om gebruik te maken van een los databasemanagementsysteem. Het kan zijn dat er verschillende databasemanagementsystemen worden ontwikkeld en verschillende applicaties verscheidene systemen gebruiken. De gebruiker heeft dan op een gegeven moment een aantal systemen op zijn smartphone draaien. Als ontwikkelaar moet er dan ook een keuze gemaakt worden voor een bepaald databasemanagementsysteem. Deze moet dan los geïnstalleerd worden naast de applicatie. Dit zijn allemaal problemen die opgelost kunnen worden, wanneer de database binnen de applicatie geïntegreerd is.

Een nadeel hiervan is dat het databasemanagementsysteem niet zo uitgebreid kan zijn als op een desktop pc. Dit is ook merkbaar bij SQLite.

Er is wel sprake van datatypes, primaire sleutels, vreemde sleutels en velden die niet 'null' mogen zijn. Alleen is er geen strikte invoercontrole over, iets waar bij grote databasemanagementsystemen altijd sprake van is. Om de integriteit van de database niet in gevaar te brengen moet er bij de invoer van data in de database controle uitgevoerd worden. Dit betekent dat er extra programmeercode gemaakt moet worden. Het is een vrij 'revolutionair' idee om bij een relationele database geen controle te hebben op de invoer. Toen ik vertelde over dit systeem aan de andere projectleden en de klant, keken zij verbaasd op dat zoiets bestond. Maar het voordeel van volledige integratie binnen de applicatie weegt zwaarder dan het ontbreken van de integriteitscontrole. Deze integriteitscontrole kan altijd zelf gedaan worden binnen de programmatuur.

Het bouwen van de database

Nadat de keuze gemaakt was voor het gebruik van SQLite als databasemanagementsysteem, kon de database gebouwd worden. SQLite maakt gebruik van de commandline en dat is een vrij tijdrovende klus. Men moet eerst handmatig een sql-bestand aanmaken waarin de gehele database geschreven is, zodat men deze kan importeren en de database kan aanmaken binnen SQLite. Omdat deze methode deels achterhaald is doordat grote databasemanagementsystemen dit via een grafische gebruikersinterface kunnen doen, ben ik op zoek gegaan naar een grafische gebruikersinterface voor SQLite. Er bleek een applicatie te bestaan, genaamd SQLite Browser. Dit is een open source applicatie. Helaas ondersteunt SQLite Browser niet alle functionaliteiten die SQLite biedt. Zo ondersteunt SQLite Browser niet alle datatypen, geen primaire en vreemde sleutels en kan er niet in aangegeven worden of een veld wel of niet 'null' mag zijn. Dit zijn wel basisfunctionaliteiten die in de database moeten komen. Ik heb als workaround in de SQLite Browser alle tabellen en velden aangemaakt. Deze heb ik als sql-bestand geëxporteerd en handmatig met de juiste datatypen, sleutels en 'null'-waarden aangepast. Daarna heb ik dit sql-bestand weer geïmporteerd binnen de SQLite Browser. Deze herkent nu de goede datatypen, sleutels en velden waar wel of geen 'null' ingevoerd mag worden.



Afbeelding 5.7: De SQLite Browser

Implementeren van de database

Toen de database klaar was kon deze toegevoegd worden aan het project. Daarna moest er code geïmplementeerd worden zodat de data uit de database bevroegd en gewijzigd kan worden.

De basis voor deze code heb ik gehaald uit de tutorial die ik heb doorlopen tijdens het inwerken in Android. Daarin staat hoe een database aangemaakt kan worden en geeft men een voorbeeld van een sql-statement die uitgevoerd wordt op de database. Als deze basis ligt, is het relatief makkelijk om een andere sql-statement uit te voeren op de database.

Wanneer de medicijnkast wordt aangemaakt door de gebruiker wordt de database gecreëerd op de smartphone. Wanneer men geen gebruik van de medicijnkast wenst te maken, zal deze niet geïnstalleerd worden. Op het moment van aanmaken haalt de applicatie het sql-bestand op waarin de database staat gedefinieerd en installeert deze op de smartphone.

De invoercontrole op de invoervelden om de integriteit van de database te kunnen bewaken staat in de grafische gebruikersinterface. Dit heb ik in overleg met Gunnar besloten, omdat er bij een foutieve invoer direct een melding gegeven kan worden dat er sprake is van een foutieve invoer.

5.5 Beveiligingsplan uitbreiden

Door extra sessies die zijn gehouden aan het begin van de tweede sprint, is er voor gekozen om minder tijd te besteden aan de beveiliging. Er is besloten om een advies op te stellen over de beveiliging van de data in de medijncast database. Daarnaast worden er enkele onderwerpen aangedragen waarin meer onderzoek gedaan moet worden tijdens toekomstige sprints.

Encryptie

Er wordt privacygevoelige informatie opgeslagen in de medijncast database op de smartphone. Deze privacygevoelige informatie moet met passende maatregelen beveiligd worden. Uit de risicoanalyse in het beveiligingsplan is geconcludeerd dat het betrouwbaarheidsaspect aanzienlijk verhoogd is. Om de privacygevoelige informatie vertrouwelijk en integer te houden, is het nuttig om gebruik te maken van encryptie.

Ik heb uitgezocht of het verstandig is om alleen de privacygevoelige velden of de gehele database te voorzien van encryptie. Aangezien encryptie en decryptie veel tijd kost en de database vele velden bevat, heb ik er voor gekozen om alleen de privacygevoelige velden te voorzien van encryptie. Ik heb uitgezocht welke velden privacygevoelige informatie bevatten en dit voorgelegd aan de projectleider en de klant voor feedback.

Omdat er geen tijd meer beschikbaar was om de encryptie te kunnen implementeren, heb ik in het beveiligingsplan een voorbeeld implementatie van encryptie en decryptie in Android toegevoegd. Dit kan een basis zijn voor een implementatie van encryptie in een van de volgende sprints.

Naast het gebruik van encryptie over de database is het verstandig om tijdens de verder ontwikkeling van de applicatie onderzoek te doen naar de volgende aspecten:

- Database overdracht tussen twee smartphones of naar een computer.
- Juridisch onderzoek naar de wetgeving omtrent het opslaan van persoonsgegevens.
- Onderzoek naar zwakheden in de gebruikte platformen (JBoss SEAM en Google Android).
- Onderzoek naar fysieke beveiliging.

Voor meer informatie wil ik u graag verwijzen naar het beveiligingsplan dat als bijlage toegevoegd zit.

5.6 Testen

Door tijdgebrek aan het einde van de tweede sprint, is er door Koos geen gebruikerstest uitgevoerd. Dit is iets wat tijdens de eerste sprint wel is gedaan.

Gunnar heeft wel de functionele testen uitgevoerd, welke positief uitgekapt zijn.

Ik heb technische testen gedaan met betrekking tot de medicijnkast database. De programmatuur die ik ontwikkeld heb om data te kunnen toevoegen, wijzigen en verwijderen in de database heb ik aan testen onderworpen. De resultaten uit deze testen waren positief. Doordat ik al veel testen heb uitgevoerd tijdens het ontwikkelen verliepen ook deze testen naar wens.

Test	Goed (%)	Fout (%)	Was fout, nu goed (%)	Was goed, nu fout(%)
Technisch run 3	100	0	0	0

Tabel 5.3: Testresultaten van de technische run in sprint 2

6 Afronding

Tijdens deze afstudeerperiode zijn er documenten, diagrammen, ontwerpen en programmeercode gemaakt. Deze zijn tijdens de laatste week volledig overgedragen aan Infoprofs. Dit is gedaan door alle laatste versies op de SVN te zetten.

Alle documenten, inclusief testdata, zijn beoordeeld door de projectleider. De meeste documenten zijn reeds tijdens het tot stand komen enkele malen beoordeeld en verbeterd.

De code is door senior software engineer Andries gecontroleerd. Alleen de data dictionary heb ik op zijn aanraden een laatste update gegeven.

De code werd goed beoordeeld en aan de hand van de code hebben Gunnar en ik API-documentatie gegenereerd in de vorm van een JavaDoc. Deze JavaDoc is in de vorm van html pagina's. De Javadoc beschrijft welke packages en klassen er zijn. Ook beschrijft deze de methoden en variabelen die aanwezig zijn in de applicatie en wat deze doen.

Door het gebruik van de JavaDoc kan er bij eventuele verdere ontwikkeling snel worden ingelezen door de nieuwe ontwikkelaar. De overdracht verliep zonder problemen.

Er is ook nog een tweede demo gegeven voor de klant, opdrachtgever en sponsors. Bij deze demo lieten wij als projectgroep het uiteindelijke resultaat van het project zien. De klant vertelde dat hij het gevoel heeft dat hij met het opgeleverde resultaat naar mogelijke nieuwe sponsors kan stappen om geld te verkrijgen voor verdere ontwikkeling van de applicatie. Ook de opdrachtgever en de sponsors waren tevreden met het geboekte resultaat. Na de demo zijn de klant, opdrachtgever en sponsors met elkaar aan tafel gaan zitten om te praten over hoe het beste verder gegaan kan worden met dit project. Dit is uiteindelijk toch het doel van de opdracht. Het geeft een bevredigend gevoel als de mensen waarvoor je de opdracht uitvoert tevreden zijn.



Afbeelding 6.1: Het voorbereiden van de videodemo

7. Evaluatie

In dit hoofdstuk kijk ik terug op de afgelopen stageperiode. Ik evalueer wat goed en wat minder goed ging. Daarbij wil ik ook kijken naar wat eventueel beter zou kunnen. Tijdens deze evaluatie wil ik als eerste reflecteren op het proces dat ik doorlopen heb. Hierbij blik ik terug op de gebruikte methoden en technieken. Daarna wil ik de door mij opgeleverde producten evalueren. Hierbij geef ik mijn mening over het tot stand komen van het product. Waarom heb ik het op deze wijze aangepakt en wat kan ik in de toekomst anders doen.

Vervolgens evalueer ik de beroepstaken/competenties die ik gekozen heb voor de start van de afstudeerperiode. Ik wil daarbij kijken hoe ik deze beroepstaken/competenties heb uitgevoerd en hoe ik dit eventueel beter had kunnen doen.

Tot slot wil ik nog een algemene conclusie trekken betreffende het project. Hierin laat ik naar voren komen hoe ik de afstudeerperiode heb ervaren.

7.1 Proces

In dit hoofdstuk evalueer ik het proces dat ik doorlopen heb tijdens de afstudeerperiode. Hierin komen de methoden en technieken die ik gebruikt heb aan de orde.

Scrum

Het gebruik van de op Scrum gebaseerde agile methode vond ik erg fijn werken.

Deze methode is erg geschikt voor het werken aan een kleinschalig project. Er komt niet te veel bureaucratie kijken bij het ontwikkelen. Dit is in mijn ogen ook niet nodig bij het ontwikkelen van een prototype applicatie. De nadruk lag vooral op het goed documenteren en de correctheid van de code. Men wilde dit prototype zo ver mogelijk ontwikkelen en met de methode Scrum is dit mijns inziens goed gelukt. In mijn ogen zou een methode als Rational Unified Proces (RUP), dat op school aan bod gekomen is, te uitgebreid zijn voor een kleinschalig project als dit. Er is maar een beperkt aantal mensen dat aan de applicatie werkt. Wanneer er problemen ontstaan tijdens de ontwikkeling kan er op individueel niveau gecommuniceerd worden om het probleem op te lossen. Het is daarom ook belangrijker om de gemaakte programmeercode overdraagbaar te maken voor de toekomst.

Qua communicatie lijkt het werken met Scrum in een kleinschalig project erg op de projecten die ik op school doorlopen heb. Deze projectgroepen bestaan meestal uit vier tot zes personen. Hierbij is ook de dagelijkse communicatie, zoals een 'Daily Scrum', erg belangrijk.

Door de uitgebreide ervaring op school met het werken volgens deze methode was het voor mij vrij gemakkelijk om op de op Scrum gebaseerde methode over te stappen. Dit gold overigens niet alleen voor mij, maar ook voor de overige leden van de projectgroep. Er was een goede samenwerking en communicatie tussen de projectleden. De motivatie om zo ver mogelijk te komen, iets wat belangrijk is bij Scrum, was erg goed. Er is hard gewerkt om zoveel mogelijk doelstellingen te behalen.

Het was wel erg jammer dat door de aanpassingen die gemaakt zijn op de officiële Scrum methode, niet het optimale rendement uit het project gehaald is. Doordat de klant een beeld geschetst wilde hebben van extra functionaliteiten die toegevoegd kunnen worden aan de applicatie, is er veel tijd verloren gegaan. Ik had deze tijd graag gebruikt om enkele functionaliteiten uit het beveiligingsplan te implementeren.

JTrac

Het werken met JTrac (issue tracker) was erg fijn. Door de kleinschaligheid van het project is de JTrac niet onoverzichtelijk geworden. Het is erg handig voor het melden van bugs en om de status van de voortgang van een ander projectlid te kunnen bekijken. Het was wel zo dat er vaak bij het voltooien van een (deel van) een feature persoonlijk gecommuniceerd werd. Daarbij werd dan vaak een beknopte uitleg gegeven. Hierdoor schoot het bijwerken van JTrac er soms bij in. In het vervolg wil ik strikter omgaan met het werken met een dergelijke 'issue tracker'. Wanneer iedereen consequent werkt met het systeem, kan elk projectlid goed de status van het project bijhouden. Men blijft goed op de hoogte van de ontwikkeling en de problemen die zich voordoen. Dit is daardoor ook goed gedocumenteerd. Daardoor kan er bij het ontstaan van problemen gemakkelijk worden bepaald waar deze zijn ontstaan. Dit kan worden meegenomen in het vervolg van het project.

Subversiesysteem

Ik ben erg te spreken over wijze van werken met een subversiesysteem. Tijdens mijn opleiding heb ik bij verscheidene projecten gewerkt met een dergelijk systeem. Het is ook een onderwerp dat aan bod komt tijdens een van de projecten. Aan het begin van dat project moest er een subversiesysteem op een server worden geïnstalleerd. Deze moest gebruikt worden tijdens het project. Deze techniek heb ik ook tijdens latere projecten gebruikt. Ook bij het afstudeerproject verliep het gebruik van het subversiesysteem naar behoren. Het is incidenteel gebeurd dat er een niet werkende versie van de programmatuur op de server gecommit was. Dit kan met een subversiesysteem gemakkelijk ongedaan gemaakt worden door een oudere versie van het corrupte bestand terug te zetten.

Het is jammer dat het systeem af en toe problemen had met het in

een keer vernieuwen van alle subversiemappen. Het gebeurde soms dat het systeem op één bestand vastliep en daardoor de overige mappen niet meer werden geüpdate. Wanneer het subversieprogramma deze fout aangaf, moest elke map apart nog een keer geüpdate worden om er verzekerd van te zijn de laatste versie te bezitten.

Android

Ik heb gemengde gevoelens over het werken met Android. Het inwerken in Android ging prima. Doordat Android gebaseerd is op Java, een taal die tijdens de opleiding vaak gebruikt wordt, was het vrij eenvoudig om te werken met Android. Het grote nadeel van Android is de instabiliteit van het platform. Ik heb de indruk dat Google veel te snel een besturingssysteem voor een smartphone op de markt heeft willen brengen. Niet elke functionaliteit kan direct goedwerkend geïmplementeerd worden. Ik had hier vooral last van bij het gebruiken van zelfgetekende certificaten bij SSL. De SSL implementatie die bestaat binnen Android geldt alleen voor officieel uitgegeven certificaten. Zelfgetekende certificaten gaven een foutmelding, waardoor ik zelf een implementatie moest gaan uitzoeken en schrijven. Gunnar is met de grafische user interface tegen dezelfde problemen aangelopen. In het 'bug report' van Android staan bugs die vele jaren geleden gemeld zijn en nog steeds niet verholpen zijn. Vergeleken met de Java en .NET omgevingen waarmee ik op school gewerkt heb, is dit een behoorlijk verschil. De hiervoor genoemde platformen zijn degelijk ontwikkeld en hebben deze problemen in veel mindere mate. Het is ook jammer dat de Android markt erg versplinterd is. Een ontwikkelaar van een smartphone kan een eigen variant op Android implementeren op smartphones. Hierdoor kan het gebeuren dat het gebruik van intents (iets wat ik juist als een voordeel zie) niet meer goed functioneert. Het direct aanspreken van bijvoorbeeld een agendafunctie kan verschillen op smartphones van verschillende ontwikkelaars.

Ik zou graag nog eens willen kijken hoe er voor de iPhone, BlackBerry of Windows Mobile applicaties ontwikkeld moeten worden. Dan kan ik zien of deze problemen specifiek zijn voor Android of voor het merendeel van de besturingssystemen voor de smartphone.

Eclipse

Ik ben tevreden over de keuze voor Eclipse als IDE. Het is erg gemakkelijk om te werken met Eclipse. Ik was vrij snel bekend met de belangrijkste functionaliteiten binnen de omgeving. Eclipse in combinatie met de plugin voor Android is ideaal. De Android SDK kan erg gemakkelijk binnen de omgeving geïntegreerd worden. Vooral het direct gebruik kunnen maken van de emulator van het Android besturingssysteem was erg prettig. Gemaakte programmeercodes kunnen zo direct gedraaid worden zonder dat er een fysieke smartphone met Android aanwezig hoeft te zijn. Dit scheelt aanzienlijk in de aanschaf van smartphones.

Samen met Visual Studio.NET vind ik dit een van de betere IDE's om in te werken.

TestFrame

Het werken met TestFrame verliep naar wens. TestFrame is niet zo uitgebreid als bijvoorbeeld TMap. Het is wel erg praktisch ingericht. Het was vrij eenvoudig om te leren werken met TestFrame. De projectleider, die een voorkeur had voor het werken met TestFrame, heeft de andere projectleden een kleine cursus gegeven. Zelfs voor Koos, die niets af wist van het testen van software, was het een begrijpelijke techniek. Bij dit project paste TestFrame prima, maar voor een groter project waarbij meer testen nodig zijn is een techniek als TMap beter. Er is op een praktische manier gewerkt met TestFrame. Wanneer er met grote systemen gewerkt wordt moet de mate van testen vele malen zorgvuldiger zijn dan op deze praktische manier.

TestFrame gaf voor dit project een goed overzicht van de kwaliteit van de software. De projectleider heeft door de resultaten uit TestFrame en de beoordeling van de programmeercode door de senior software engineer, voldoende inzicht gekregen in de kwaliteit van de applicatie.

7.2 Producten

In dit hoofdstuk evalueer ik de door mij opgeleverde producten. Hierbij kijk ik naar de doelstelling betreffende het product en het uiteindelijke resultaat.

Planning

Bij de start van het project is een planning opgesteld waarvan we als projectgroep dachten dat deze haalbaar was. Uiteindelijk is deze initiële planning niet geheel uitgevoerd, maar op enkele punten aangepast. De planning voor de eerste sprint verliep zoals verwacht. De grote problemen ontstonden bij het uitvoeren van de tweede sprint. Doordat de klant extra sessies verlangde om meer betrokken te zijn bij het project is er tijd verloren gegaan. Hierdoor konden niet alle geplande functionaliteiten uit de backlog door mij uitgevoerd worden. Er is toen in overleg met de klant bekeken welke functionaliteiten hij het belangrijkste vond. Als projectgroep konden wij inschatten of deze functionaliteiten haalbaar waren in de resterende tijd. In mijn geval is er tijdens deze tweede sprint voor gekozen om de nadruk te leggen op de ontwikkeling van de database voor het medicijnkastje. Doordat dit medicijnkastje uitgebreider moest worden dan gepland, is besloten om voor deze sprint minder uitgebreid te werken aan de beveiliging. Er is voor gekozen een advies te geven over de beveiliging voor het medicijnkastje. De implementatie van het advies kwam te vervallen door tijdgebrek. Dit project heeft mij duidelijk gemaakt dat een planning maken erg lastig is en dat er vaak obstakels opduiken die

er voor zorgen dat een projectplanning aangepast moet worden. Het heeft wel aangetoond dat het maken van een goede planning erg nuttig is. Vooral het scoren van complexiteit voor het ontwikkelen van functionaliteiten met behulp van spelkaarten vond ik een erg goede manier. Hiermee kan een goede inschatting gemaakt worden van de doorlooptijd van de ontwikkeling van functionaliteiten. Dit is een techniek die ik, mocht dat nodig zijn, in het vervolg wil gebruiken.

Systeemontwerp

Het maken van het systeemontwerp verliep prima. Ik heb dit ontwerp gemaakt en geüpdate samen met Gunnar. Ik heb in het verleden vele projecten samen met hem uitgevoerd, dus er was geen sprake van communicatieve problemen.

We hebben beiden onze ervaringen met het inwerken in Android gedeeld en aan de hand daarvan het eerste ontwerp opgesteld. Tijdens de tweede sprint moest er een keuze gemaakt worden in de architectuur. Dit is gebeurd in overleg met de klant. Wij hebben de mogelijke architecturen aan hem voorgelegd en daarbij argumenten geleverd. Na de keuze van de klant voor de architectuur, waarbij de database van het medicijnkastje op de smartphone komt, hebben wij ons ontwerp aangepast.

Verbinding tussen de smartphone en de backend server

Ik ben trots op de verbinding die ik heb kunnen leggen tussen de smartphone en de backend server. Het benaderen van de servlet met een smartphone is iets waar weinig documentatie over te vinden is. Dit is zeker niet het geval wanneer er gebruik gemaakt moet worden van SSL met zelfgetekende certificaten. Voor mezelf heb ik hierbij mijn grenzen weten te verleggen.

Ik heb voor het opzetten een goede communicatie gehad met Andries en Gunnar. Beide hadden een aanzienlijk belang in de verbinding die ik opgesteld heb. Met Andries heb ik goed kunnen overleggen hoe de verbinding opgezet moest worden. Hij had nagenoeg hetzelfde idee en heeft mij ook nog kunnen helpen bij het opzetten van de verbinding. Met Gunnar heb ik goed kunnen overleggen over hoe hij het ontvangen en verzenden van data het liefst aangeleverd ziet voor de grafische user interface.

Ik vond het wel erg jammer dat bij de aanvang van mijn afstudeerperiode Andries nog niet klaar was met het opzetten van de basis van de backend. Ik heb daarom zelf een miniatuurweergave gemaakt van deze omgeving. Dit heeft veel extra tijd gekost. Het voordeel hiervan is wel dat ik een beter inzicht had in de gehele omgeving. Mijn miniaturomgeving verschilde niet veel met de omgeving die door Andries is opgezet, waardoor ik uiteindelijk de omgeving toch nog makkelijk kon benaderen. Ik moest alleen enkele kleine aanpassingen maken voor de benadering van de servlet. Wanneer ik weer in een dergelijke situatie terecht zou komen ga ik wel meer communiceren met de persoon die de backend maakt, zodat er eventueel minder 'dubbel' werk

gedaan hoeft te worden.

Ik vond het fijn dat ik Andries heb kunnen helpen met de implementatie van SSL aan de serverkant. Hij had hier nog geen ervaring mee en aangezien ik me tijdens dit project in de implementatie van SSL heb verdiept, heb ik ook gekeken naar de serverkant. Met Andries heb ik SSL aan de serverkant werkend gekregen. Hij stond erg open voor mijn kennis en was ook bereid om van Gunnar en mij te leren. Dit gaf mij het gevoel dat ik na de opleiding van de afgelopen jaren voldoende kennis heb opgebouwd om professioneel te kunnen werken. Wanneer mijn kennis niet toereikend zou zijn, zou Andries waarschijnlijk minder open hebben gestaan voor mijn kennis.

Database op de smartphone

Ik ben erg trots op het resultaat van de database voor de smartphone. Door de wens van de klant dat de database ook voor toekomstige sprints geschikt moet zijn, was er meer werk nodig dan gepland. Dit vond ik een mooie uitdaging. Er moest erg goed nagedacht worden over welke velden en tabellen er in de database aanwezig moeten zijn. Ik was erg blij met feedback en hulp van de andere projectleden. Het merendeel heeft ervaring met het ontwikkelen van databases, wat de keuze voor velden en tabellen eenvoudiger maakte. Het nadeel van de vele 'experts' op het gebied van databases waren kleine conflicten die ontstonden in inzichten. Dit was voornamelijk het geval met de klant. Deze heeft vele jaren ervaring in het ontwikkelen in databases, maar dit conflicteerde soms met de door mij geleerde theorie over het opzetten van relationele databases. Ook andere projectleden, waaronder de projectleider, hadden dit verschil in inzicht met de klant. Uiteindelijk is dit opgelost door als projectgroep nog een normalisatieslag uit te voeren over de database. Dit heeft er voor gezorgd dat met behulp van tussentabellen er niet onnodig data dubbel opgeslagen moeten worden.

Ik was blij dat de projectleider dezelfde ideeën had over de opzet van een database, want ik vind het moeilijk als student in te gaan tegen iemand die beweert jarenlange ervaring te hebben op dat terrein. De projectleider heeft mij daarbij erg geholpen om de database zo te ontwerpen dat deze overeenstemt met de door mij geleerde theorie.

Beveiligingsplan en de implementatie van de resultaten

De beoordelingen van de projectleider en twee senior software engineers betreffende het beveiligingsplan waren erg positief. Er was op enkele kleine punten kritiek, maar als totaal vonden ze het een goed en duidelijk opgezet document. Het was begrijpelijk en de argumentatie voor keuzes vonden ze erg goed. Ik was blij om dit te horen van mensen die ervaring hebben binnen dit vakgebied. Het is een grote blijk van waardering voor het door mij geleverde werk.

De punten van kritiek die zij gaven waren gebaseerd op goede argumenten. Ik heb vervolgens de nodige aanpassingen aangebracht. Dit waren kleine punten, zoals het vollediger uitwerken van een argumentatie.

7.3 Beroepstaken/competenties

Bij aanvang van het afstuderen moest er een aantal beroepstaken/competenties gekozen worden, waar ik extra aandacht aan zou geven tijdens het doorlopen van het project. Hieronder staan de gekozen beroepstaken/competenties genoemd. Onder het kopje van een beroepstaak/competentie staat eerst, in cursief, beschreven hoe ik deze beroepstaak/competentie dacht aan te pakken. Daaronder beschrijf ik hoe het uitvoeren van deze beroepstaak/competentie verlopen is.

1.1 Selecteren methoden, technieken en tools

De organisatie heeft nog nooit met Android gewerkt of ontwikkelt, daarom is het aan de projectleden om de omgeving en tools te selecteren, waarmee deze opdracht wordt gerealiseerd.

In overleg met de projectleider en de overige projectleden is er voor gekozen om een op Scrum gebaseerde agile methodiek te gebruiken. Voor een kleinschalig project als dit is het een zeer goede keuze geweest. Problemen konden onder andere met behulp van de JTrac issue tracker en de goede communicatie snel opgelost worden. De keuze voor de technieken van de issue tracker JTrac en het gebruik van een subversiesysteem is ook gemaakt in overleg met de projectleden. Hierbij waren de projectleden eensgezind. Ik heb er voor gekozen, deels samen met Gunnar, om gebruik te maken van Unified Modelling Language (UML) voor het maken van bijvoorbeeld het systeemontwerp. Dit is een techniek die ik op school geleerd heb en waarmee de projectleider, de senior software engineer en de andere junior software engineer bekend waren. De keuze voor Eclipse als IDE was, zoals al in hoofdstuk 7.1 staat genoemd, erg goed. Het was een ideale en overzichtelijke omgeving om in te ontwikkelen voor Android.

1.4 Uitvoeren analyse door definitie van requirements

Er wordt een prioriteitenlijst gemaakt met functionele en niet-functionele requirements.

Koos heeft onderzoek gedaan naar wensen die mogelijke gebruikers hebben. Dit heeft hij op aanraden van Gunnar en mij omgezet in use case diagrammen. Naast het onderzoek van Koos bij mogelijke gebruikers zijn er ook gesprekken geweest met de klant. In deze gesprekken hebben wij als projectgroep geprobeerd te achterhalen welke wensen de klant voor ogen had.

Bij de gesprekken met de klant heb ik samen met de andere projectleden door middel van het stellen van vragen, de wensen van de klant proberen duidelijk te krijgen. Op basis van deze lijst van wensen zijn functionele en niet-functionele requirements opgesteld. Deze zijn later gebruikt bij het opstellen van het systeemontwerp en de keuze voor architectuur.

2.2 Ontwerpen,bouwen en bevragen van een database

De G-standaard wordt geïmplementeerd op een server, zodat deze bereikbaar is via een smartphone (client). Ook zal er lokaal een database voor het zogenoemde 'medicijnkastje' moeten worden aangemaakt en bevroegd worden. Hierin komt de gebruikersspecifieke informatie in te staan.

Het ontwerpen van de database werd complexer dan ik had verwacht. De oorzaak hiervan was het feit dat de klant halverwege het project extra eisen stelde. Hij wilde de database zo ontwerpen, dat deze ook geschikt is voor toekomstige sprints. Dit zorgde er voor dat er extra tijd in het ontwerpen gestoken moest worden. Ik was wel blij dat de klant en de overige projectleden mij geholpen hebben bij het ontwerpen van de database. Hierbij moet vooral gedacht worden aan welke velden en tabellen er wel en niet in de database moeten komen. Na hier een helder beeld van te hebben gekregen, heb ik een databaseontwerp gemaakt en die laten beoordelen door de projectleider en de klant. Dit ontwerp heb ik gemaakt met behulp van de theorie over relationele databases uit het boek 'Principes van relationele database' van Guy De Tré. Toen het ontwerp goed beoordeeld was ben ik dit om gaan zetten in een sql-lite database.

De implementatie van de programmeercode voor het bevragen en bewerken van de database verliep naar wens. Er was goede documentatie te vinden over het gebruik van SQLite binnen Android.

Doordat ik gebruik heb gemaakt van normalisatie tijdens het ontwerpen van de database, het maken van een technisch database model en het maken van een data dictionary heb ik aangetoond dat deze database op een professionele manier is ontworpen.

3.1 Ontwerpen van software architectuur

Van het gehele softwaresysteem wordt een architectuur gemaakt. Deze bevat frontend (Android applicatie), backend (Java Servlets) en de databases.

Nadat de requirements opgesteld waren, aan de hand van interviews met de klant en eventuele gebruikers (zie competentie/beroepstaak 1.4), heb ik samen met Gunnar een softwarearchitectuur opgezet. Dit hebben we gedaan met behulp van UML.

Tijdens de tweede sprint hebben wij, op aandringen van de klant, een definitieve keuze gemaakt voor de gehele architectuur van de omgeving. Hierbij is er voor gekozen om de verantwoordelijkheid voor privacygevoelige data bij de gebruiker neer te leggen. Dit hebben we gedaan door de mogelijke architecturen uit te werken en hierbij argumenten voor en tegen de architectuur te geven. De klant heeft daarna gekozen voor een van de architecturen.

3.2 Ontwerpen systeemdeel

Na het globale ontwerp wordt het systeem in modules opgedeeld, de relevante modules zijn: de communicatie tussen front-end (Android) en de backend (Webserver) en de backend van het 'medicijnkastje' (oa. de lokale database).

Na het maken van het systeemontwerp heb ik de door mij te ontwikkelen modules ontworpen met behulp van sequence diagrammen (UML). Hierin heb ik bijvoorbeeld weergegeven hoe de communicatie tussen de backend server en de smartphone verloopt. Van deze diagrammen bij het ontwerpen van een systeemonderdeel heb ik ook gebruik gemaakt tijdens projecten op school. Hier heb ik ook lessen in gehad en het gebruik van deze diagrammen werd aangeraden door docenten.

3.3 Bouwen applicatie

Nadat er een module is ontworpen wordt deze geïmplementeerd in iteratieve stappen door de ontwikkelaar.

Nadat het ontwerp van een module gemaakt was, ben ik deze gaan ontwikkelen. Ik heb rekening gehouden met het feit dat mijn programmeercode samen moet werken met de programmeercode die ontwikkeld werd door Andries en Gunnar. De code is ook enkele keren aangepast. Een voorbeeld hiervan is de implementatie van SSL. De 'onbeveiligde' verbinding die ik reeds ontwikkeld had moest dusdanig aangepast worden dat er gebruik gemaakt kon worden van SSL. Dit is een voorbeeld van een iteratieve stap die ik heb doorlopen binnen mijn programmeercode.

Ik heb tijdens het bouwen van de applicatie rekening gehouden met eventuele overdracht van de programmeercode. Ik heb gekozen voor logische benamingen voor variabelen, methoden en klassen. Daarnaast heb ik de code voorzien van beschrijvend commentaar. Bij de overdracht van de programmeercode aan Infoprofs is dit goedgekeurd door de senior software designer.

3.5 Uitvoeren en rapporteren over het testproces

Nadat een module is ontwikkeld, zullen er functionele testen worden uitgevoerd aan de hand van het aangeleverde testrapport.

Het door de projectleider gemaakte testplan heb ik uitgevoerd en gerapporteerd. Dit heb ik gedaan met behulp van TestFrame. Dit was, in overleg met de overige projectleden, gekozen door de projectleider. Na het uitvoeren en het rapporteren van de testen is deze gecontroleerd door de projectleider.

Ik vond TestFrame erg fijn werken. Zeker voor een kleinschalig project als deze is het prima te gebruiken. Het is een niet te uitgebreide methode van testen. TestFrame geeft ook een duidelijk overzicht in percentages en diagrammen over de kwaliteit van de software. Iets waardoor de projectleider snel de kwaliteit van de opgeleverde software kon zien.

7.4 Conclusie

Terugkijkend op de gehele afstudeerperiode komt een aantal positieve en negatieve punten naar voren.

Positieve punten:

- Het is gelukt om een verbinding tussen een smartphone en backendserver op te zetten met beveiliging in de vorm van SSL.
- Ik heb veel kennis kunnen opdoen in het ontwikkelen voor Android en de smartphone in het algemeen.
- Ik heb een professionele database voor de smartphone ontworpen, gebouwd en geïmplementeerd.

Negatieve punten:

- Ik heb tijdens de tweede sprint meer tijd willen besteden aan de beveiliging. Dit is echter niet gelukt door tijdgebrek.
- Android is te versplinterd en niet stabiel genoeg.

De opdrachtgever, sponsors en projectleider waren blij met het opgeleverde resultaat. Ook de klant was tevreden en heeft aangegeven dat hij denkt dat hij nieuwe sponsors en eventuele andere partijen kan benaderen voor een samenwerking.

Ik was blij om dit te horen, net als de andere projectleden. Ik heb veel geleerd tijdens deze afstudeerperiode en kijk met een fijn gevoel terug met de samenwerking met Infoprofs en de klant Rick Dekker.



Afbeelding 7.1: De applicatie werkend op een Android smartphone

Literatuurlijst

Boeken:

Jos Warmer & Anneke Kleppe (2006), Praktisch UML (NL) Pearson Education Benelux b.v. ; ISBN-13: 9789043008129

Guy de Tré (2007), Principes van databases (NL) Pearson Education Benelux b.v. ; ISBN: 9789043013024

P.L. Overbeek (2006), Informatiebeveiliging onder controle (NL) Pearson Education Benelux b.v. ; ISBN: 9789043006927

Internet:

<http://developer.android.com/>
De door Google uitgegeven informatie over de Android omgeving.

<http://www.hackaday.com/>
Bevat tutorials over Android voor beginners.

<http://www.google.com/events/io/2010/sessions.html>
Video's over Android die zijn gemaakt tijdens het IO 2010 event.

<http://www.androidsnippets.com/>
Voor het gebruik van voorbeelden van implementaties van programmeercode.

Verklarende woordenlijst

ADT	Android Development Toolkit in Eclipse
Agile methode	Een groep ontwikkelingsmethodieken met snelle iteratieslagen
Android	Een besturingssysteem voor de smartphone
Backlog	Een lijst met functionaliteiten die ontwikkelt kunnen worden
Eclipse	Een voor Java gemaakte IDE
Encryptie	Het coderen van data gebaseerd op een bepaald algoritme
Feature driven development	Een agile methode om software te ontwikkelen
G-standaard	Een database met medicijninformatie van alle medicijnen die verkrijgbaar zijn op de Nederlandse markt
IDE	Een Integrated Development Environment. Een omgeving om software in te kunnen ontwikkelen
Intent	Een event gebruikt in het Android systeem
Java	Een programmeertaal en platform die op meerdere hardware- en besturingssystemen werkt
JAVADOC	Documentatie over programmeercode
JTrac	Een issue tracking systeem.
Netbeans	Een IDE voor Javas
Scrum	Een methodiek voor projectmanagement
SDK	Software development kit. Wordt gebruikt om software te ontwikkelen
Serializable	Wanneer data serializable is, kan deze via een netwerkprotocol overgestuurd worden
Smartphone	Een telefoon die applicaties kan uitvoeren
Sql-code	Taal die ontwikkeld is voor het werken met databases. Deze kan databases aanmaken en bewerken.
SQLite	Een SQL database
SVN	Subversiebeheer. Wordt gebruikt voor versiebeheer tijdens projecten
TestFrame	Een test rapportage raamwerk
Unified Modelling Language (UML)	Een methode om diagrammen te modelleren
Z-index	Een organisatie die de G-standaard database bijhoudt en uitgeeft