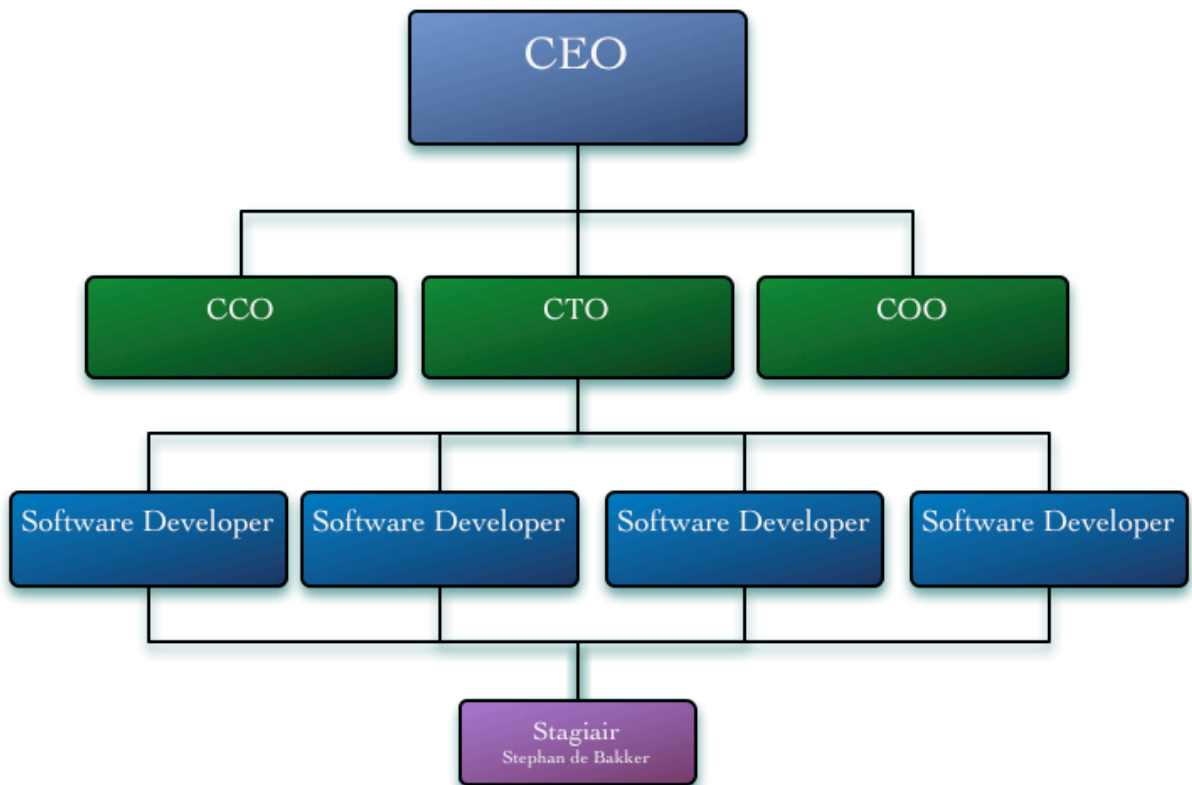


Bijlagen

Bijlage A

Solware



Afstudeeropdracht

Ontwikkel voor vastgoedbeheerders van A tot Z een App waarmee onderhoudswerkzaamheden door de klanten (bewoners) van de Vastgoedbeheerders zelf gemeld kunnen worden. Voor de afstudeeropdracht zal je werken naar de Solware werkwijze waar ook onderzoek, conceptontwikkeling en daadwerkelijke realisatie toebehoren.

Gedurende de afstudeeropdracht wordt je begeleid door professionals van Solware. Echter dient de basiskennis zoals iOS ontwikkeling d.m.v. zelfstudie te worden bijgebracht. Solware neemt deze basiskennis serieus en zal ook tijdens het afstuderen deze persoonlijke ontwikkeling toetsen.

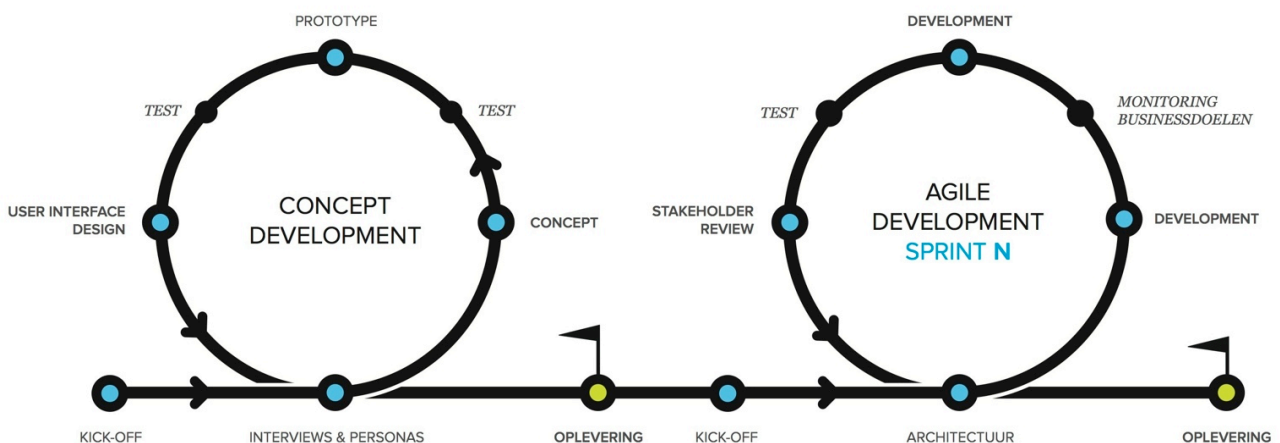
Als verantwoordelijke voor de ontwikkeling van deze App heb je ook contact met klanten en leveranciers van Solware.



Werkwijze Solware

Solware ontwikkelt voor de voorlopers van elke **High Impact Apps**. Onze web- en mobiele Apps hebben een grote impact op het primaire proces van onze klanten. Hierdoor verandert de manier waarop onze klanten met hun klanten en leveranciers samenwerken.

Solware hanteert een werkwijze waarmee op basis van **Business Value** wordt ontwikkeld. Deze werkwijze bestaat uit de fases **Concept Development** en **Agile Development**.



✓ Business doelstellingen bepalen	✓ Business doelstellingen behalen
✓ Stakeholder interviews & Persona's	✓ Team effort met betrokken stakeholders
✓ Story Map met User Stories	✓ Agile Development in sprints
✓ User Experience Design concept	✓ High Impact App

Om het juiste eindresultaat op te leveren voeren wij voorafgaand aan de daadwerkelijke realisatie een aantal korte stappen uit waarna een concept wordt opgeleverd.

Het **User Experience Design concept** bestaat uit visuals welke een goed beeld geven van het te verwachten eindresultaat. De hoofdlijnen van de App zijn hierdoor vele weken eerder inzichtelijk en u kunt vanaf dat moment al sturen op output! Daarnaast kunnen we met het concept de verwachtingen managen bij alle betrokken stakeholders.

Achtergrond en vraagstelling

Naar aanleiding van besprekingen tussen de Vastgoedbeheerder en Solware is de behoefte geuit een App te ontwikkelen waarmee bewoners hun onderhoud kunnen melden. Hierdoor hoeft de Vastgoedbeheerder deze werkzaamheden niet meer te verwerken en kunnen deze klussen direct worden ingepland.

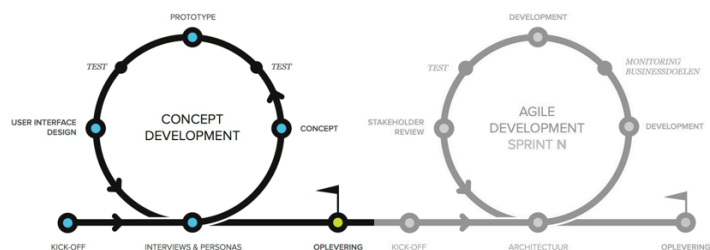
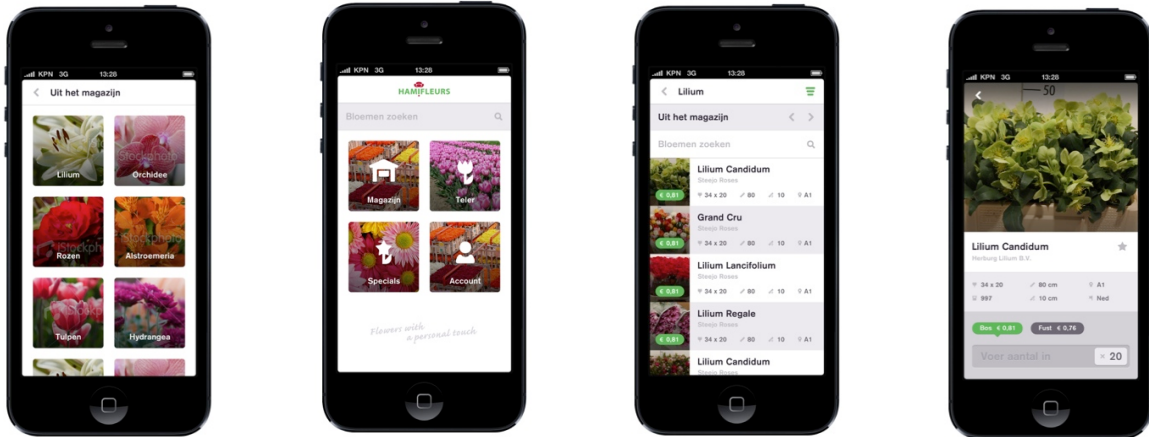


- Afbeelding: Voorbeeld vergelijkbare App -

- **Opstartscherm in huisstijl opdrachtgever,**
- **Eenmalig registreren bewoner,**
 - Registratie van naam, adres, telefoonnummer en e-mailadres,
 - Gegevens worden op de iPhone opgeslagen voor vervolg meldingen.
- **Selecteren ruimte van werkzaamheden,**
 - Ruimtes nader af te stemmen zoals: Woonkamer, badkamer, keuken, etc..
- **Selectie van categorie werkzaamheden,**
 - Te selecteren categorieën in de ruimte zoals toilet, wasbak, afzuigkap, etc..
- **Melding opsturen,**
 - Invoeren voorkeursdatum in komende drie dagen,
 - Eventueel competentie benodigd voor werkzaamheden zoals timmerman,
 - Vrij tekstveld om eventuele opmerkingen toe te voegen.
- **Melding bevestigen**
 - Duidelijke informatie over te verwachten vervolg communicatie.

De eerste fase (Concept Development)

Visuals waarin de eisen en wensen in visuele voorbeelden worden gepresenteerd zoals de applicatie er mogelijk uit zal komen te zien. Onderstaande voorbeeld van eerder ontwikkelde visuals van een native iOS Apps.



De Realisatie (Agile Development)

De realisatie van het concept tot en met de oplevering van uw **High Impact App** wordt door Solware gerealiseerd door toepassing van de kernwaarden van het Agile Manifesto:

- Individuen en interactie boven processen en gereedschappen.
- Werkende software boven uitgebreide documentatie.
- Samenwerking met de klant boven contractonderhandelingen.
- Inspelen op verandering boven het volgen van een plan.

Werken volgens deze kernwaarden betekent in de praktijk meer waarde en impact voor uw business dan louter waarde hechten aan specificaties en afspraken.

Ontwikkeltraject

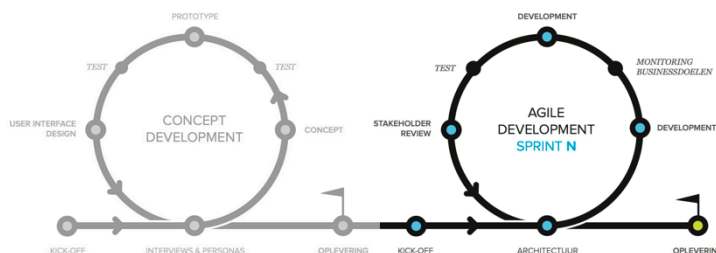
Met de Solware werkwijze kiest u voor het vastleggen van tijd en kosten. De reikwijdte van uw project zal tijdens het ontwikkeltraject continu worden aangescherpt waardoor u maximale toegevoegde waarde krijgt, altijd met focus op uw business doelstellingen.

Voor u als opdrachtgever is een actieve rol weggelegd waarin u in samenwerking met Solware de maximale toegevoegde waarde voor uw project ontwikkelt.



Tweewekelijkse iteraties

Uw project wordt gefaseerd en ontwikkeld in tijdvakken van twee weken: de opening, het middenspel en het eindspel. Tijdens de opening wordt een eerste concept van uw project opgezet. De functionele waarde wordt vervolgens tijdens het middenspel opgebouwd. In het eindspel wordt het product afgerond en komen zaken als presentatie, het verbeteren van prestaties en gebruiksgemak aan bod.



Bijlage C

Solware Enterprise Framework (SEF)

SEF is het door Solware in PHP¹ ontwikkelde web Framework dat gebruikt wordt om web applicaties en REST API's te bouwen. Dit Framework vereenvoudigt communicatie met een database door gebruik te maken van zogenaamde definities, deze definities beschrijven tabellen in de database. Deze definities kunnen gebruikt worden om nieuwe objecten aan te maken en weg te schrijven naar de daartoe behorende database. SEF zorgt ervoor dat andere benodigde bestanden aan de hand van deze definities worden aangemaakt, zoals een Model klasse en een Formulier klasse die speciaal voor het bewerken of toevoegen van dit object gemaakt is.

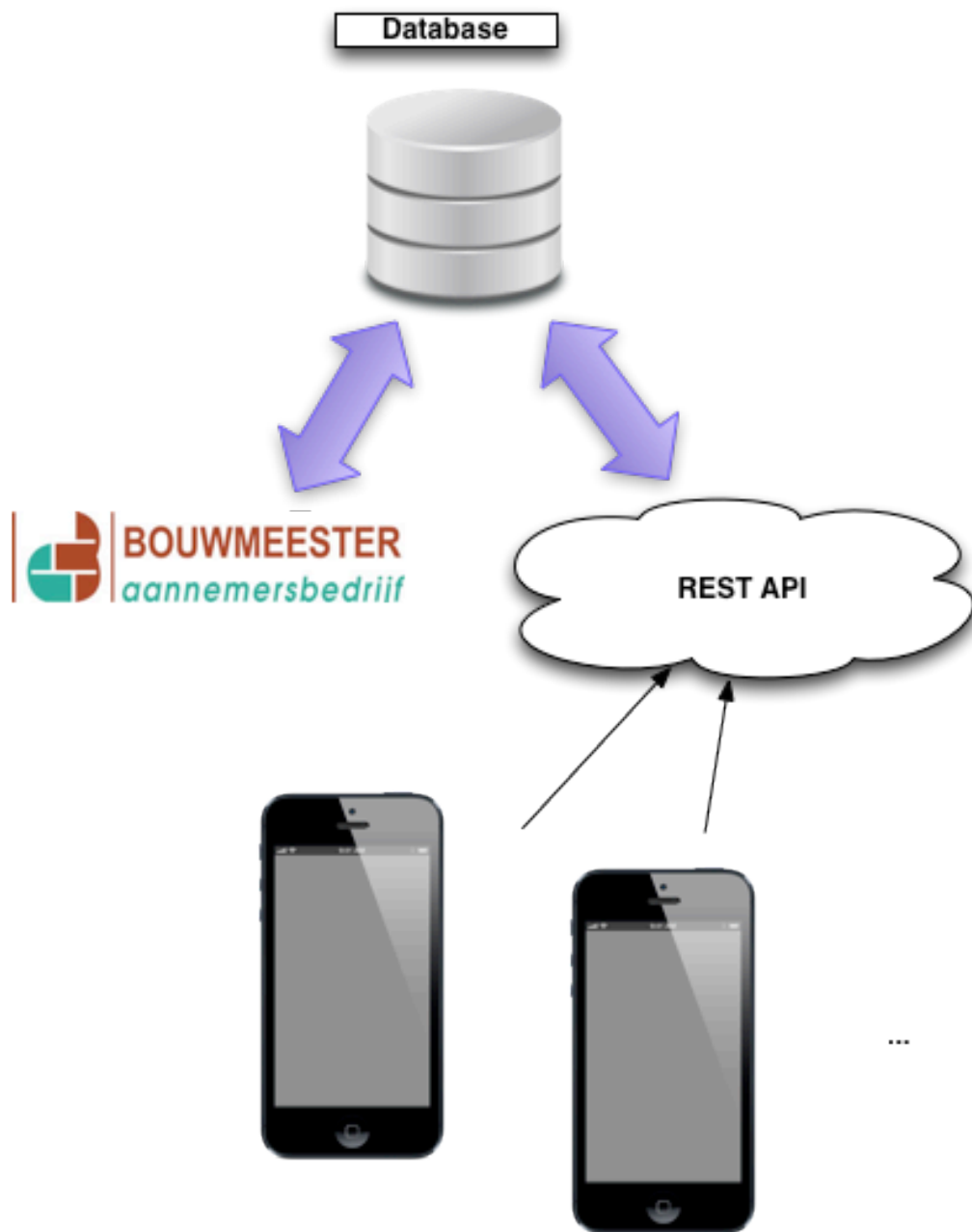
SEF maakt het ook eenvoudig om aan de hand van deze definities pagina's aan te maken waar bepaalde gebruikers informatie van deze objecten kunnen aanpassen, toevoegen of verwijderen. Op deze manier is het voor de ontwikkelaars erg eenvoudig om snel een compleet beheer systemen te ontwikkelen. Voor dit project zal de afstudeerder hier ook gebruik van maken.

Een laatst belangrijk onderdeel van SEF is de ondersteuning voor migratie naar nieuwe versie van de code. Om een migratie uit te voeren worden zogenaamde migratiescripts geschreven die alle benodigde wijzigingen aan de vorige versie van het systeem doorvoeren naar de nieuwe versie.

Source code van de in de achtergrond werkende code zal niet toegevoegd worden aan de bijlagen in verband met geheimhouding. Daarnaast zal wel code worden toegevoegd van door de afstudeerder toegevoegde objecten.

¹ <http://www.php.net>















Bijlage D



Sprint Backlog 2

Periode 10 jul 2013 t/m 23 jul 2013						
Realisatie in sprint 239:01 / 326:48 (73,14%)						
Titel	Actief	Status	Release	Begroot / realisatie		
BC-500: [OA] Lijst van Elementen [IOS]		Gereed	---	5:00 / 5:03	€	
BC-501: [OA] Lijst van ruimtes [IOS]		Gereed	---	5:00 / 3:00	€	
BC-502: [OA] Lijst van gebreken [IOS]		Gereed	---	5:00 / 2:09	€	
BC-503: [OA] Tonen van eigen agendapunten in de datumselectie.		Gereed	---	8:00 / 11:40	€	
BC-504: [OA] Verstuurde dagdeel en datum controleren p beschikbaarheid op platform.		In ontwikkeling	---	8:00 / 2:40	€	
BC-505: [OA] Documenteren nieuwe API functies.		Gereed	---	3:00 / 0:46	€	
BC-506: [OA] Afstudeerverslag bijwerken		Gereed	---	8:00 / 12:45	€	
BC-507: [OA] Opslaan van ontvangen elementen, ruimten en gebreken in CoreData. [IOS]		Gereed	---	13:00 / 12:47	€	
BC-508: [OA] Ontwikkelen van een CoreData model		Gereed	---	2:00 / 0:43	€	
BC-517: [OA] Volgorde verbeteren van de gebrek selectie flow.		Gereed	---	5:00 / 2:18	€	
BC-509: [OA] Updaten van bestaande CoreData objecten.		Gereed	---	5:00 / 4:30	€	
BC-519: [OA] Naam, email en telefoonnummers toevoegen aan een defect om te uploaden		Gereed	---	5:00 / 7:45	€	
BC-510: [OA] Afstudeerbezoek 18 Juli		Gereed	---	4:00 / ---	€	
Totaal begroot	76:00					
Totale realisatie	66:10					

Sprint Backlog 3

Periode		24 jul 2013 t/m 6 aug 2013				
Realisatie in sprint		168:55 / 367:52 (45,92%)				
Titel	Actief	Status	Release	Begroot / realisatie		
 BC-520: [OA] Foto toevoegen aan een defect.		Gereed	---	5:00 / 8:06	€	
 BC-527: [OA] Feedback verwerken review.		Gereed	---	3:00 / 2:31	€	
 BC-528: [OA] API omgooien naar de REST gedeelte van SEF		Gereed	---	5:00 / 16:34	€	
 BC-526: [OA] Foto uploaden en toevoegen in database		Gereed	---	8:00 / 9:41	€	
 BC-521: [OA] Adresgegevens gebruiker laten invullen		Gereed	---	5:00 / 9:55	€	
 BC-522: [OA] Gebruiker een woningcorporatie uit laten kiezen.		Gereed	---	5:00 / 3:37	€	
 BC-524: [OA] Notificaties naar de gebruiker, registreren van een gebruiker voor deze notificaties		In ontwikkeling	---	8:00 / 7:37	€	
 BC-536: [OA] Beschikbaarheid nummer tonen bij inplanning.		Gereed	---	2:00 / 1:19	€	
 BC-533: [OA] Versturen van notificaties naar de juiste gebruiker		Gereed voor oppakken	---	5:00 / 1:42	€	
 BC-525: [OA] Afstudeerverslag bijwerken		In ontwikkeling	---	13:00 / 11:13	€	
Totaal begroot		59:00				
Totale realisatie		72:17				

Sprint Backlog 4

Periode 7 aug 2013 t/m 20 aug 2013						
Realisatie in sprint 273:36 / 384:24 (71,18%)						
Titel	Actief	Status	Release	Begroot / realisatie		
BC-544: [OA] Afstudeerdossier afronden Sprint 3		Gereed	---	5:00 / 4:18		
BC-537: [OA] Bevestiging of afwijzing van de afspraak via de e-mail ontvangen.		Gereed	---	5:00 / 1:04		
BC-548: [OA] Online formulier voor het bewerken en toevoegen van beschikbaarheid van medewerkers.		Gereed	---	3:00 / 4:00		
BC-538: [OA] Op de BouwCloud een afspraak bevestigen of afwijzen.		Gereed	---	8:00 / 8:55		
BC-547: [OA] Localizeren van de teksten in de mobiele applicatie.		Gereed	---	3:00 / 2:16		
BC-540: [OA] Valideren van ontvangen data in de API.		Gereed	---	5:00 / 9:02		
BC-539: [OA] Bedenken uitgebreide methode voor ondersteuning meerdere iOS versies.		Gereed	---	3:00 / 2:31		
BC-542: [OA] Controleren of er voor iOS 6 onbekende methodes worden gebruikt en zonodig aanpassen.		Gereed	---	5:00 / 3:20		
BC-545: [OA] Afstudeerdossier begin maken aan Sprint 4.		Gereed	---	5:00 / 5:44		
BC-546: [OA] Wanneer je een afbeelding probeert te vervangen met een andere afbeelding blijft de status "processing" hangen.		Gereed	---	3:00 / 3:48		⚠
BC-543: [OA] Adresgegevens opzoeken op telefoon bug oplossen.		Gereed	---	5:00 / 0:46		⚠
BC-541: [OA] Implementeren van de User Interface per iOS versie die wordt ondersteund.		Gereed	---	13:00 / 14:34		
BC-552: [OA] Voorbereiden voor de demonstratie op 23 aug.		Gereed	---	8:00 / 14:11		
Totaal begroot 71:00						
Totale realisatie 74:34						

Sprint Backlog 5

Periode 21 aug 2013 t/m 3 sep 2013						
Realisatie in sprint 261:36 / 376:32 (69,48%)						
Titel	Actief	Status	Release	Begroot / realisatie		
BC-567: [OA] Mockups maken van de beheer systemen op de BouwCloud.		Afgerond	---	5:00 / 9:00		
BC-553: [OA] Meeting met John voor een review van de opdrachtgever app d.d. 21 augustus.		Gereed	---	8:00 / 5:24		
BC-556: [OA] Dagdelen van 1 uur toevoegen en de dagdelen beheersbaar maken.		Gereed	---	2:00 / 0:31		
BC-558: [OA] Een Excel import methode maken die alle beschikbaarheid van een maand / jaar automatisch verwerkt.		Gereed	---	8:00 / 6:24		
BC-555: [OA] Overzicht van reserveringen verbeteren en overzichtelijker maken.		Gereed	---	13:00 / 17:49		
BC-554: [OA] Woningcorporaties selecteerbaar maken om te tonen in de mobiele applicatie.		Gereed	---	2:00 / 1:03		
BC-564: [OA] Afbeelding bug oplossen en uiteindelijke afbeeldingen maken met een resolutie van 800 x 600.		Gereed	---	4:00 / 6:00		
BC-561: [OA] Op de mobiele applicatie een scherm tonen die de status van verzoek verzenden aangeeft.		Gereed	---	5:00 / 11:04		
BC-568: [OA] Een bewoner zijn reservering laten annuleren vanuit de lijst op de mobiele applicatie.		Gereed	---	5:00 / 3:45		
BC-557: [OA] In de ontvangen e-mail meer relevante informatie aangeven.		Gereed	---	3:00 / 5:58		
BC-565: [OA] De bewoner een lijst tonen van alle geuploade reserveringen.		Gereed	---	5:00 / 4:38		
BC-562: [OA] Mogelijkheid inbouwen in de mobiele applicatie dat meerdere defecten geupload kunnen worden		Gereed	---	5:00 / 3:08		
BC-566: [OA] Afstudeerdossier bijwerken en Tussentijds verslag schrijven / opsturen.		In ontwikkeling	---	8:00 / 1:24		
Totaal begroot 73:00						
Totale realisatie 76:13						

Sprint Backlog 6

Periode 4 sep 2013 t/m 17 sep 2013						
Realisatie in sprint 261:46 / 382:01 (68,53%)						
Titel	Actief	Status	Release	Begroot / realisatie		
BC-564: [OA] Afbeelding bug oplossen en uiteindelijke afbeeldingen maken met een resolutie van 800 x 600.		Gereed	---	4:00 / ---		
BC-578: [OA] Feedback review verwerken, het beheer gedeelte afstellen op de flow van andere reeds bestaande beheer onderdelen.		Gereed	---	5:00 / 4:13		
BC-582: [OA] Review feedback verwerken voor de mobiele applicatie.		Gereed	---	3:00 / 2:51		
BC-573: [OA] Afmaken van sprint verslag sprint 4, conclusie en demonstratie.		Gereed	---	3:00 / 0:58		
BC-579: [OA] Vanuit het beheer systeem een iPhone kunnen afsluiten om reparatieverzoeken te kunnen maken.		Nieuw	---	5:00 / ---		
BC-577: [OA] Communicatie over het ontwerp van de mobiele applicatie.		Nieuw	---	1:00 / ---		
BC-575: [OA] Schrijven van Sprint 5 verslag in het afstudeerverslag.		Gereed	---	8:00 / 6:41		
BC-559: [OA] Verificatie onderzoeken om zo Abuse tegen te gaan.		Afgerond	---	3:00 / 4:52		
BC-560: [OA] Implementeren van een verificatie methode om Abuse tegen te gaan.		Gereed	---	13:00 / 12:02		
BC-580: [OA] SSL gebruiken voor elke uitgaande communicatie vanaf de iPhone.		Gereed	---	8:00 / 4:27		
BC-576: [OA] Afstudeerverslag voorbereiden voor concept versie.		Gereed	---	5:00 / 8:46		
BC-581: [OA] Afbeeldingen tonen bij Locaties, Ruimten, Elementen en Defecten.		Gereed	---	13:00 / 15:57		
BC-583: [OA] Starten schrijven van Sprint 6 en SSL gedeelte / verificatie.		Gereed	---	8:00 / 6:18		
Totaal begroot	79:00					
Totale realisatie	67:09					

Sprint Backlog 7

Periode 18 sep 2013 t/m 1 okt 2013						
Realisatie in sprint 219:30 / 397:06 (55,27%)						
Titel	Actief	Status	Release	Begroot / realisatie		
BC-624: [OA] In het beheer systeem het archief bijwerken door een herstel knop te implementeren.		Gereed	---	3:00 / 0:40		
BC-625: [OA] Kleine review punten verwerken voor het beheer systeem.		Gereed	---	3:00 / 1:54		
BC-634: [OA] Afschrijven Sprint 6		Gereed	---	3:00 / 0:45		
BC-627: [OA] Bestaande review feedback op de mobiele applicatie verwerken.		Gereed	---	3:00 / 0:33		
BC-626: [OA] Validatie toevoegen binnen de REST API om te controleren of bepaalde gegevens bekeken mogen worden.		Gereed	---	3:00 / 0:59		
BC-628: [OA] Ervoor zorgen dat de gebruiker niet meerdere verzoeken kan toevoegen in een enkel reparatieverzoek.		Gereed	---	1:00 / 0:52		
BC-639: [OA] Afsudeerdossier feedback deels verwerken en meer op backlog richten.		Gereed	---	5:00 / 2:30		
BC-642: [OA] Communicatie over het ontwerp van de mobiele applicatie.		Gereed	---	1:00 / 1:49		
BC-645: [OA] Reageren op memory warnings om zo ongebruikt geheugen vrij te maken.		Gereed	---	5:00 / 2:55		
BC-629: [OA] Bij het invoeren van naam en adresgegevens op het toetsenbord een extra balk toevoegen die het navigeren makkelijk.		Gereed	---	8:00 / 6:11		
BC-643: [OA] Wanneer het het gebruikte token voor authenticatie niet meer geldig is bepaalde acties afhandelen.		Gereed	---	3:00 / 0:28		
BC-631: [OA] Tonen van het defect afbeelding in het overzicht per reparatieverzoek.		Gereed	---	1:00 / 2:24		
BC-632: [OA] Ontwikkelen van API request en response objecten die gebruikt worden door de mobiele applicatie.		Gereed	---	5:00 / 3:40		
BC-633: [OA] Herschrijven van de operatie klassen door gebruik te maken van notificaties van internet veranderingen.		Gereed	---	21:00 / 38:02		
BC-640: [OA] Start met schrijven van Sprint 7 in het afstudeerdossier.		Gereed voor oppakken	---	5:00 / ---		
Totaal begroot	70:00					
Totale realisatie	63:49					

Plan van Aanpak

Stephan de Bakker
Afstudeeropdracht Solware B.V.

Achtergrond	3
Opdrachtomschrijving	4
<i>Opdrachtgever en opdrachtnemer</i>	<i>4</i>
<i>Probleemstelling</i> 4	
<i>Doelstelling opdracht</i>	<i>4</i>
<i>Opdrachtbeschrijving</i>	<i>4</i>
<i>Op te leveren producten</i>	<i>4</i>
<i>Projectgrenzen</i> 4	
<i>Randvoorwaarden</i> 5	
Aanpak	6
<i>Methoden en technieken</i>	<i>6</i>
<i>Werkzaamheden</i> 6	
<i>Standaarden</i> 6	
<i>Planning</i> 7	
Projectinrichting	9
<i>Projectorganisatie</i> 9	
<i>Informatievoorziening</i>	<i>9</i>
<i>Faciliteiten</i> 9	
<i>Kwaliteitsborging</i> 9	
Risicofactoren	10
Kosten en Baten	11

Achtergrond

Het bedrijf Solware ontwikkelt en exploiteert high impact applicaties, Solware houdt zich vooral bezig met de ontwikkeling van Web- en mobiele applicaties. In totaal bestaat het software ontwikkelteam uit 5 personen die als doel hebben om applicaties voor opdrachtgevers te realiseren. Het project zal invloed hebben op deze software ontwikkel afdeling aangezien de opdracht te maken zal hebben met het ontwerpen en realiseren van software.

Het bedrijf heeft een opdracht van een woningcorporatie gekregen die later is uitgewerkt in het nu uit te voeren project. Dit project zal uitgevoerd worden als afstudeeropdracht onder begeleiding van het software ontwikkelteam van Solware.

Opdrachtomschrijving

Opdrachtgever en opdrachtnemer

De opdrachtgevende organisatie heeft de naam Bouwmeester, dit is een bedrijf die groot-, mutatie- en klachtenonderhoud, verbouwingen en renovaties aan woningen en kantoren biedt. Het bedrijf is gevestigd op de grens tussen Den Haag en het Westland. Voor meer informatie bezoek dan de site www.bouwmeester.org

Probleemstelling

De opdrachtgever wil de communicatie tussen de vastgoedbeheerders en bewoners optimaliseren. Afspraken voor reparaties in vastgoed worden telefonisch ingepland in overleg met de vastgoedbeheerder. Daarom wil de opdrachtgever een project starten die dit traject voor een groot deel automatiseert.

Doelstelling opdracht

Het doel van dit project is het optimaliseren en versnellen van de communicatie tussen een bewoner en de vastgoedbeheerder. Wat als gevolg heeft dat het inplannen van bijvoorbeeld een reparatieafpraak versneld en makkelijk is dan nu het geval, hierdoor worden bewoners sneller geholpen met problemen bij hun vastgoed.

Opdrachtbeschrijving

Het project zal opgelost worden door middel van de ontwikkeling van een mobiele applicatie. De applicatie kan worden gedownload door bewoners, door gebruik van deze applicatie kan de bewoner eenvoudig een reparatie afspraak inplannen met een vastgoedbeheerder. De applicatie moet voldoen aan de volgende eisen:

- De bewoner kan een probleem beschrijven en opsturen naar de vastgoedbeheerder
- De bewoner kan een foto toevoegen aan het bovenstaande probleem.
- De bewoner kan door middel van een kalender een afspraak inplannen voor reparatie.
- De bewoner zal door middel van notificaties op de hoogte worden gehouden van voortgang.
- De bewoner kan inloggen op een account en NAW gegevens invoeren of wijzigen.

Bovenstaande eisen zijn in grote richtlijnen beschreven, deze eisen zullen verder worden uitgeschreven en kan op ieder moment aangepast worden naar wens van de opdrachtgever of het team dat het project uitvoert.

Op te leveren producten

Uiteindelijk zal het gehele product opgeleverd worden, onder dit product behoren de volgende onderdelen:

- De mobiele applicatie (beschikbaar voor mobiele apparaten naar wens van de opdrachtgever)
- Source Code van de applicatie
- Afstudeerdossier

Projectgrenzen

De opdrachtnemer zorgt ervoor dat de applicatie gerealiseerd wordt en klaargemaakt wordt om in de winkel van een mobiel platform geplaatst te worden. Daarnaast zal de opdrachtnemer ervoor zorgen dat eventuele problemen die aan bod komen na de officiële release opgelost worden.

Randvoorwaarden

Het project zal door middel van Scrum gerealiseerd worden, deze ontwikkelmethode staat vast en is gekozen door het opdrachtnemende bedrijf. Verder zullen de te gebruiken middelen ook door het opdrachtnemende bedrijf worden vastgesteld.

Als laatste zal het opdrachtnemende bedrijf de hoofd functionele eisen uitwerken en in vooraf vastgestelde tijdsvakken aan het opdrachtgevende bedrijf tonen.

Aanpak

Methoden en technieken

Om dit project uit te voeren is gekozen voor de ontwikkeltechniek Scrum, deze methode is onderdeel van de Agile Development methode en wordt bij nagenoeg alle projecten toegepast bij het opdrachtnemende bedrijf. Door deze methode wordt om de twee weken een belangrijk functioneel onderdeel van het project opgeleverd aan de opdrachtgever zodat deze op zijn beurt een mening kan geven op het product dat tot dusver is opgeleverd.

Om de werking het systeem dat ontwikkelt wordt te omschrijven zullen flow-charts ontwikkeld worden. Klassediagrammen worden ontwikkeld om de softwarematige werking van het systeem te beschrijven en eventueel worden use-cases gemaakt om de interactie van een gebruiker met het systeem zo duidelijk mogelijk weer te geven. Alle diagrammen en schema's zullen hetzelfde programma gemaakt worden om zo een tekenstijl te houden.

Werkzaamheden

Door gebruik van de Scrum ontwikkelmethode wordt de planning opgedeeld in in dit geval twee-wekelijkse iteraties, die bij deze methode een Sprint wordt genoemd. Het eerste onderdeel dat voor alle Sprints wordt uitgevoerd is de *Release Planning Meeting*, in dit onderdeel wordt een zogenaamde *Product Backlog* opgesteld wat je kan vergelijken met functionele eisen van het systeem. Daarnaast wordt het doel van het project opgesteld en wordt besproken wat de beste manier is om tot dit doel te komen. Verder bevat elke Sprint de volgende werkzaamheden:

Sprint Planning Meeting

Bij de Sprint Planning Meeting wordt de iteratie die uitgevoerd gaat worden ingepland, tijdens deze planning wordt besproken hoe deze Sprint en wat er in de Sprint uitgevoerd zal worden.

Sprint Review

Aan het einde van een Sprint wordt besproken wat er allemaal tijdens deze Sprint is uitgevoerd en welke onderdelen ook daadwerkelijk klaar zijn (*Done*).

Sprint Retrospective

In dit onderdeel wordt bekeken wat er allemaal verbeterd kan worden naar aanleiding van de zojuist uitgevoerde Sprint ten opzichte van de volgende uit te voeren Sprint. Zo kan bijvoorbeeld worden aangewezen welk deel van het proces verbeterd zou kunnen worden.

Daily Scrum

Alle bovenstaande werkzaamheden worden eens per Sprint uitgevoerd met uitzondering van de Daily Scrum, deze wordt namelijk elke dag uitgevoerd om duidelijk te maken wat een teamlid de vorige dag heeft uitgevoerd, wat hij vandaag gaat doen en welke obstakels dit teamlid tegen is gekomen.

Standaarden

Het project zal uitgevoerd worden door een persoon, namelijk de afstudeerder. De afstudeerder zal wel worden begeleid door het ontwikkelteam van Solware. Daarom zullen de volgende standaarden worden gehanteerd.

Software

Omdat de mobiele applicatie voor het platform iOS ontwikkeld zal worden moet er gebruik worden gemaakt van de applicatie Xcode. Deze applicatie wordt door het ontwikkelteam van Solware ook gebruikt voor de ontwikkeling en is daarom een goede standaard. Tesktdocumenten zullen worden opgeleverd in .doc formaat aangezien dit het meest gehanteerde formaat is ter wereld en dit ook binnen het bedrijf wordt gebruikt. Er zal gebruik worden gemaakt van een vaste stijl die in alle documenten terug te vinden is. Voor diagrammen zal gebruik worden gemaakt van het programma Omni-Graffle, diagrammen zullen geëxporteerd als pdf bestand worden opgeleverd.

Een belangrijke standaard voor de afstudeerder is dat er na elke werkdag in stukje over de werkzaamheden en beslissingen wordt geschreven in het logboek. Op deze manier zijn beslissingen altijd terug te vinden en te onderbouwen.

Planning

Zoals eerder al is aangegeven zal dit project uit een aantal tweewekelijkse iteraties (Sprints) bestaan, aangezien er 20 weken beschikbaar zijn kunnen deze makkelijk worden onderverdeelt in 10 Sprints. Deze zijn als volgt ingepland.

Sprint #	Startdatum	Reviewdatum	Beschrijving
1	24 jun. 2013	5 jul. 2013	Plan van Aanpak Product Backlog ¹ Walking Skeleton (v 0.1) User Stories ¹
2	8 jul. 2013	19 jul. 2013	Opleveren v 0.2 Defect aangeven bij een probleem. Week 4 Coordinator bezoek.
3	22 jul. 2013	2 aug. 2013	Opleveren v 0.3 Foto toevoegen bij het probleem
4	5 aug. 2013	16 aug. 2013	Opleveren v 0.4 Ondersteunen iOS 6.0
5	19 aug. 2013	30 aug. 2013	Opleveren v 0.5 Voortgangsverslag (uiterlijk 23 aug. 2013)
6	2 sep. 2013	13 sep. 2013	Opleveren v 0.6 Concept Afstudeerdossier (uiterlijk 13 sep. 2013)
7	16 sep. 2013	27 sep. 2013	Opleveren v 0.7
8	30 sep. 2013	11 okt. 2013	Opleveren v 0.8
9	14 okt. 2013	25 okt. 2013	Opleveren v 0.9
10	28 okt. 2013	8 nov. 2013	Opleveren Productie versie (v1.0)
n.v.t.	10 jan. 2014	n.v.t.	Inleveren Afstudeerdossier

¹ Dit is een levend document, dat betekent dat de inhoud van dit document tijdens de voortgang van het project kan veranderen.

Per Sprint wordt een gedeelte van de te ontwikkelen applicatie opgeleverd en wordt de versie van de applicatie verhoogd. Per Sprint moeten nog een aantal doelen worden bepaald, deze doelen worden voor een groot deel aangevuld in de planning van elke Sprint en de Retrospective van de zojuist afgeronde sprint.

Tijdens de Sprints bevinden zich ook een aantal oplever documenten voor bijvoorbeeld het voortgangsverslag of het concept van het afstudeerverslag. Deze documenten worden bij de begeleidend coördinator van de afstudeerder op de gegeven datum ingeleverd.

Projectinrichting

Projectorganisatie

Bij Scrum wordt onderscheidt gemaakt tussen de volgende rollen binnen het project:

Scrum Master

De Scrum Master is verantwoordelijk voor het team en zorgt ervoor dat dit team op de juiste manier Scrum toepast. Hij werkt ook samen met de klanten en zorgt ervoor dat een Product Owner wordt aangesteld voor een project. De Scrum Master is in dit geval Chi-Fai Lam van Lier.

Product Owner

De Product Owner is verantwoordelijk voor de Product Backlog zodat het gehele team hier aan kan werken. Ook zorgt hij ervoor dat het team weet welke onderdelen de hoogste prioriteit hebben. De Product Owner is in dit geval Michael van Lier.

Team

Het team zorgt ervoor dat het Product Backlog in incrementen wordt opgedeeld zodat dit op zijn beurt in Sprints kan worden opgedeeld.

Informatievoorziening

De communicatie binnen het bedrijf zal verlopen tussen de Product Owner en de Scrum Master, aangezien de Scrum Master zorgt voor de communicatie tussen het opdrachtgevende en opdrachtnemende bedrijf. Verder heeft de Product Owner verder contact met de rest van het ontwikkelteam van Solware.

Elke dag zal een Daily Scrum worden gehouden waarin het Team de voortgang bespreekt van het project. Communicatie zal wanneer nodig verder via e-mail worden onderhouden.

Faciliteiten

De faciliteiten die benodigd zijn voor dit project zijn de volgende:

- Een Mac computer
- Een test mobiel apparaat (iPhone / iPad)
- Ontwikkelsoftware (Xcode)
- Een Developer account bij Apple (benodigd voor de software en meer)
- Microsoft Office
- OmniGraffle

Kwaliteitsborging

Om aan te kunnen tonen dat het te ontwikkelen product voldoet aan de kwaliteitseisen zal in eerste instantie een review worden gedaan door de begeleidende ontwikkelaar van Solware. Deze begeleider is tijdens het gehele traject aanwezig geweest om zonodig tips te geven of hulp te bieden om de afstudeerder op weg te helpen.

Daarnaast zal het systeem door middel van Unit Tests en een testplan worden getest zodat het gehele systeem met zo min mogelijk bugs opgeleverd wordt. De applicatie zal binnen het bedrijf worden getest aangezien het niet mogelijk is om een applicatie in een winkel te plaatsen voor Beta-Testing of iets dergelijks. De productie versie zal daarom ook de eerste te gebruiken versie zijn voor de doelgroep.

Risicofactoren

Een risicofactor die invloed zou kunnen hebben op het slagen van dit project is de tijd die de afstudeerder in het project kan steken. In eerste instantie is namelijk 20 weken gepland voor het uitvoeren van deze opdracht, door goed inplannen en prioriteiten stellen tussen bepaalde functionele eisen is de kans dat deze factor echt een probleem wordt klein.

Kosten en Baten

In totaal zijn voor dit project per persoon $20 * 40 = 800$ uur ingecalculeerd. Aangezien het project alleen door de afstudeerder wordt uitgevoerd komt dat dus neer op totaal 800 uur voor het gehele project. Verder zijn er materiaal kosten zoals de computer en de software, en moet er ook betaald worden voor het ontwikkelaars account van Apple. Als laatste wordt verwacht dat de applicatie nog een aantal updates nodig heeft na de release, denk hierbij aan bugfixes en eventuele toevoeging van extra features.

De opbrengsten van de ontwikkeling van deze applicatie is versnelde communicatie tussen de bewoner en een vastgoedbeheerder door middel van de applicatie. De vastgoedbeheerder hoeft hierdoor geen telefoon meer bemand te laten omdat het inplannen digitaal gaat zonder direct contact tussen beide partijen, wat een kostenbesparing zou zijn.

Bijlage G

User Stories

Registreren / Inloggen

Als **bewoner** wil ik me kunnen aanmelden voor gebruik van de applicatie zodat ik in de toekomst werkzaamheden kan laten inplannen.

Als **bewoner** wil ik automatisch ingelogd in de applicatie blijven wanneer ik dat zeg voor de makkelijkheid, of niet wanneer ik dat instel zodat wanneer mijn telefoon weg is ik de applicatie niet openlaat voor iedereen.

Als **bewoner** wil ik mijn wachtwoord kunnen achterhalen wanneer ik hierom vraag zodat ik de applicatie alsnog kan gebruiken wanneer ik dit vergeet.

Als **bewoner** wil ik dat alle informatie die ik op een apparaat invoer ook op mijn andere iOS apparaten aanwezig is zodat ik op elk iOS apparaat de applicatie kan controleren.

Als **bewoner** wil ik dat ik alle informatie op zowel een iPhone als iPad kan weergeven.

Als **bewoner** wil ik mijn locatie automatisch laten invullen door *Location Services* waarna ik het valideer om zo het invulproces te vereenvoudigen en versnellen.

Als **bewoner** wil ik de tijd dat een sessie duurt aan kunnen passen naar mijn eigen wens.

Als **bewoner** wil ik de mogelijkheid om alle informatie van de telefoon te verwijderen.

Invoeren werkzaamheden

Als **bewoner** wil ik aan kunnen geven in welke ruimte het probleem zich plaatsvindt zodat ik onderscheid kan maken tussen verschillende problemen mochten deze aanwezig zijn.

Als **bewoner** wil ik een overige optie kunnen invoeren als ruimte mocht mijn optie niet aanwezig zijn.

Als **bewoner** wil ik aan kunnen geven wat de categorie van het probleem is om zo ook onderscheidt tussen eventuele meerdere categorieën te maken.

Als **bewoner** wil ik een overige optie kunnen invoeren als categorie mocht mijn optie niet aanwezig zijn.

Als **vastgoedbeheerder** wil ik de lijst van ruimten en categorieën online aan kunnen passen zodat elke gebruiker een nieuwe categorie toegevoegd krijgt op de telefoon.

Als **bewoner** wil ik een lijst kunnen weergeven van alle actieve en non-actieve werkzaamheden die ik heb aangemaakt om ze zo eenvoudig te kunnen wijzigen of te verzenden.

Als **bewoner** wil ik een extra beschrijving aan het probleem kunnen toevoeging om zo duidelijker weer te geven wat het probleem is aan de vastgoedbeheerder.

Als **bewoner** wil ik een foto kunnen toevoegen aan werkzaamheden zodat de vastgoedbeheerder een visueel beeld van het probleem heeft.

Als **bewoner** wil ik een competentie aan het probleem kunnen toevoegen zodat de vastgoedbeheerder weet welke expertise benodigd is.

Als **bewoner** wil ik een toegevoegde foto kunnen bewerken om zo op het probleem in te zoomen om het probleem te verduidelijken.

Als **bewoner** wil ik een reeds bestaand probleem kunnen bewerken.

Als **bewoner** wil ik reeds bestaande werkzaamheden kunnen verwijderen en annuleren.

Plannen werkzaamheden

Als **bewoner** wil ik een datum in kunnen voeren als voorkeur (max volgende 3 dagen) bij een probleem om zo snel te worden ingepland.

Als **bewoner** wil ik dat mijn eigen agenda wordt vertoond voordat ik een eventuele afspraak inplan.

Als **bewoner** wil ik meerdere data kunnen invoeren om zo meer flexibel te kunnen zijn voor de vastgoedbeheerder en een grotere kans te hebben om een datum ingepland te krijgen.

Als **bewoner** wil ik notificaties krijgen wanneer een ingevoerd probleem van status verandert, bijvoorbeeld wanneer een datum is ingepland.

Als **bewoner** wil ik een door de vastgoedbeheerder bevestigde datum zelf eerst bevestigen voordat de afspraak vaststaat.

OPTIONEEL

Administrator / Vastgoedbeheerder werknemer?

Als **werknemer** wil ik automatisch naar een lijst worden gebracht na inloggen die mij alle statussen van werkzaamheden vertoond.

Als **werknemer** wil ik de statussen van de verschillende werkzaamheden kunnen wijzigen om zo een probleem tot rekening te nemen.

Als **werknemer** wil ik een agenda kunnen laten inplannen en bekijken met alle werkzaamheden die ik zal moeten verrichten.

Bijlage H

In dit document is een inhoudsopgave te vinden die wijst naar alle code Header en Source bestanden met per Klasse een korte beschrijving. Vanuit het afstudeerverslag wordt verwezen naar een bepaald hoofdstuk dat overeenkomt met de Klasse. De code bestanden bevinden zich in de map “Code Bijlagen”. De code van de mobiele applicatie en de REST API is hier inbegrepen. Per klasse of bestand staat aangegeven in welke pagina’s de code te vinden is.

Belangrijke bestanden	7
<i>SAppDelegate (p 1 - 6)</i>	7
<i>main (p 7)</i>	7
<i>Localizable.strings (p 8 - 21)</i>	7
<i>InfoPlist.strings & BouwCloud-Info.plist (p 22 - 24)</i>	7
<i>BouwCloud-Prefix (p 25)</i>	7
Protocollen (Delegatie interface)	8
<i>SBCollectionControllerDelegate (p 26)</i>	8
<i>SBCompatabilityInterface (p 27)</i>	8
<i>SBCustomerControllerDelegate (p 28)</i>	8
<i>SBInternetCenterDelegate (p 29)</i>	8
<i>SBKeyboardToolbarDelegate (p 30)</i>	8
<i>SBMaintenanceInterface (p 31)</i>	8
<i>SBManagedObjectInterface (p 32)</i>	8
<i>SBOperationDelegate (p 33)</i>	9
<i>SBProcessorObserverDelegate (p 34)</i>	9
<i>SBReservationManagerDelegate (p 35)</i>	9
<i>SBURLOperationDelegate (p 36)</i>	9
Models	10
<i>SBAPIRequest (p 37 - 40)</i>	10
<i>SBAPIResponse (p 41 - 44)</i>	10
<i>SBAvailableDateDescriptor (p 45 - 47)</i>	10
<i>SBColorManager (p 52 - 53)</i>	10
<i>SBCustomerDescriptor (p 54 - 55)</i>	10
<i>SBDayPartDescriptor (p 56 - 58)</i>	10
<i>SBDefectImageDescriptor (p 59 - 62)</i>	10
<i>SBFetchedResultsControllerAnimation (p 63 - 64)</i>	11

<i>SINavigationBarDetailView (p 67 - 70)</i>	11
<i>SBRelationDescriptor (p 71 - 72)</i>	11
<i>SBReservationManager (p 73 - 80)</i>	11
<i>SBUUser (p 81 - 85)</i>	11
<i>SBVerificationManager (p 86 - 91)</i>	11
<i>SBCollectionViewCellViewLayout (p 48 - 51)</i>	12
<i>SBKeyboardToolbarLayout (p 65 - 66)</i>	12
NSManagedObjects	12
<i>Defect (p 92 - 93)</i>	12
<i>Element (p 94 - 95)</i>	12
<i>Location (p 96 - 97)</i>	12
<i>Proceeding (p 98 - 100)</i>	12
<i>Reservation (p 101 - 104)</i>	13
<i>Space (p 105 - 106)</i>	13
iOS Compatability (p 107 - 156)	13
Operation Management	13
<i>SBFileOperation (p 157 - 158)</i>	13
<i>SBFileThumbnailOperation (p 159 - 162)</i>	13
<i>SBImageFileSaveOperation (163 - 164)</i>	13
<i>SBInternetCenter (165 - 169)</i>	14
<i>SBOperation (170 - 174)</i>	14
<i>SBOperationManager (p 175 - 178)</i>	14
<i>SBProcessorObserver (p 179 - 182)</i>	14
<i>SBURLOperation (p 189 - 193)</i>	14
<i>SBURLReservationOperation (p 194 - 198)</i>	14
<i>SBURLSynchronizeOperation (p 199 - 205)</i>	14
UI Elementen	15

<i>SBCollectionController</i> (p 206 - 211)	15
<i>SBCollectionTableViewCellTitleView</i> (p 212 - 213)	15
<i>SBCollectionViewLocationCell</i> (p 214 - 215)	15
<i>SBElementCollectionViewCell</i> (p 216 - 218)	15
<i>SBErrorView</i> (p 219 - 221)	15
<i>SBHomeTableViewCell</i> (p 220 - 224)	15
<i>SBKeyboardToolbar</i> (p 227 - 229)	16
<i>SBNavigationBar_iOS_6_0</i> (p 230 - 231)	16
<i>SBNavigationBar_iOS_7_0</i> (p 232 - 233)	16
<i>SBNavigationBar</i> (p 234 - 237)	16
<i>SBNavigationControllerExtended</i> (p 238 - 239)	16
<i>SBReservationOverviewCell</i> (p 240 - 241)	16
<i>SBSelectableOptionTableViewCell</i> (p 244 - 245)	16
<i>SBTextField</i> (p 246 - 247)	16
<i>SBToolBar</i> (p 248 - 249)	17
<i>SBViewController</i> (p 250 - 252)	17

View Controllers 18

Huidige ViewControllers 18

<i>SBAAllReservationsViewController</i> (p 253 - 258)	18
<i>SBAAppointmentViewController</i> (p 259 - 266)	18
<i>SBDefectSelectionViewController</i> (p 267 - 271)	18
<i>SBElementSelectionViewController</i> (p 272 - 276)	18
<i>SBHomeViewController</i> (p 277 - 280)	18
<i>SBLocationSelectionViewController</i> (p 281 - 285)	18
<i>SBOptionalInformationViewController</i> (p 286 - 294)	18
<i>SBPersonalDataViewController</i> (p 295 - 307)	19
<i>SBReservationOverviewViewController</i> (p 370 - 377)	19
<i>SBRoomSelectionViewController</i> (p 378 - 382)	19

Oude ViewControllers	19
<i>SBPhoneCustomerViewController (p 308 - 313)</i>	19
<i>SBPhoneDateViewController (p 314 - 321)</i>	19
<i>SBPhoneDefectViewController (p 322 - 324)</i>	19
<i>SBPhoneDescriptionViewController (p 325 - 329)</i>	19
<i>SBPhoneElementViewController (p 330 - 332)</i>	20
<i>SBPhoneImageViewController (p 333 - 339)</i>	20
<i>SBPhoneLocationViewController (p 340 - 342)</i>	20
<i>SBPhoneLoginViewController (p 343 - 355)</i>	20
<i>SBPhoneOverviewViewController (p 356 - 366)</i>	20
<i>SBPhoneSpaceViewController (p 367 - 369)</i>	20
XML Interface bestanden	21
<i>PhoneStoryboard (p 383 - 393)</i>	21
<i>SBCollectionViewLocationCell (p 394)</i>	21
<i>SBHomeTableViewCell (p 395)</i>	21
<i>SBReservationOverviewCell (p 396)</i>	21
<i>SBSelectableOptionTableViewCell (397 - 398)</i>	21
REST API	22
API Methodes	22
<i>OAMobileAddMaintenanceAPIMethod (p 400 - 401)</i>	22
<i>OAMobileAddressLookUpAPIMethod (p 402)</i>	22
<i>OAMobileAddUserAPIMethod (p 403 - 405)</i>	22
<i>OAMobileAllCustomerAPIMethod (p 406)</i>	22
<i>OAMobileAuthenticationRequestAPIMethod (p 407)</i>	22
<i>OAMobileAvailableDatesAPIMethod (p 408 - 409)</i>	22
<i>OAMobileCancelReservationAPIMethod (p 410)</i>	22
<i>OAMobileCheckStateAPIMethod (p 411 - 412)</i>	23

<i>OAMobileImageAPIMethod (p 413)</i>	23
<i>OAMobileSynchronizeDefectsAPIMethod (p 414)</i>	23
Gebruikte objecten	23
<i>OM_AddressLookUp p 415 - 416)</i>	23
<i>OM_AddressLookUpLocation (p 417)</i>	23
<i>OM_DayPart (p 418)</i>	23
<i>OM_Defect (p 419)</i>	23
<i>OM_Element (p 420)</i>	24
<i>OM_Location (p 421)</i>	24
<i>OM_Maintenance (p 422)</i>	24
<i>OM_MaintenanceUser (p 423 - 424)</i>	24
<i>OM_Space (p 425)</i>	24
<i>OM_UuidAuthentication (p 426 - 427)</i>	24
<i>UuidAuthenticationProvider (p 428)</i>	24
Beheersysteem	25
<i>OM_AvailabilityController (p 429 - 431)</i>	25
<i>OM_AvailabilityForm (p 432)</i>	25
<i>OM_AvailabilityOverviewForm (p 433 - 434)</i>	25
<i>OM_DayPartController (p 435)</i>	25
<i>OM_DayPartOverviewForm (p 436)</i>	25
<i>OM_MaintenanceController (p 437 - 443)</i>	25
<i>OM_MaintenanceUserController (p 444 - 446)</i>	25
<i>OM_MaintenanceUserForm (p 447)</i>	25
<i>OM_MaintenanceUserMaintenanceOverviewForm (p 448)</i>	25
<i>OM_MaintenanceUserStateHistoryOverviewForm (p 449)</i>	25

Belangrijke bestanden

De volgende bestanden worden gebruikt om de applicatie op te starten en om andere belangrijke iOS instellingen klaar te maken.

SAppDelegate (p 1 - 6)

De applicatie wordt gestart vanuit dit object door een aanroep van de methode “application: didFinishLaunchingWithOptions:”. CoreData wordt geïnitieerd etc.

“/Andere Bestanden/[SAppDelegate.h.pdf](#)”

“/Andere Bestanden/[SAppDelegate.m.pdf](#)”

main (p 7)

De methode die UIApplication main aanroepert waardoor de SAppDelegate gecreëerd wordt.

“/Andere Bestanden/[main.m.pdf](#)”

Localizable.strings (p 8 - 21)

Dit bestand bevat alle vertalingen, per taal is een apart van dit bestand gebruikt.

“/Andere Bestanden/[Localizable.strings.pdf](#)”

InfoPlist.strings & BouwCloud-Info.plist (p 22 - 24)

Bevat instellingen voor de applicatie.

“/Andere Bestanden/[InfoPlist.strings.pdf](#)”

“/Andere Bestanden/[BouwCloud-Info.plist.pdf](#)”

BouwCloud-Prefix (p 25)

Bevat de header bestanden die overal inbegrepen moeten worden.

“/Andere bestanden/[BouwCloud-Prefix.pch.pdf](#)”

Protocollen (Delegatie interface)

De volgende protocol bestanden dienen als interfaces voor objecten om te kunnen reageren op bepaalde methodes die aangeroepen worden door een “Worker” object. De bestanden zijn te vinden in de map “Protocols”.

SBCollectionControllerDelegate (p 26)

Is gebruikt voor een UICollectionView Controller om notificaties te sturen naar een delegatie object wanneer een object geselecteerd is.

“/Protocols/[SBCollectionControllerDelegate.h.pdf](#)”

SBCompatibilityInterface (p 27)

Bevat standaard methodes die geïmplementeerd moeten worden door UIViewController objecten om zo te worden geprepareerd voor de juiste iOS versie.

“/Protocols/[SBCompatibilityInterface.h.pdf](#)”

SBCustomerControllerDelegate (p 28)

Werd gebruikt om aan te geven dat een gebruiker in de Woningcorporatie controller een bepaald selectie heeft gemaakt.

“/Protocols/[SBCustomerControllerDelegate.h.pdf](#)”

SBInternetCenterDelegate (p 29)

Stuurt berichten door aan luisterende objecten wanneer internet beschikbaarheid aanpast naar beter of slechtere connectie.

“/Protocols/[SBInternetCenterDelegate.h.pdf](#)”

SBKeyboardToolbarDelegate (p 30)

Wordt gebruikt om berichten door te sturen aan een UIViewController wanneer de gebruiker een bepaalde knop heeft ingedrukt op de UIToolBar van een Keyboard.

“/Protocols/[SBKeyboardToolbarDelegate.h.pdf](#)”

SBMaintenanceInterface (p 31)

Wordt gebruikt om het centrale reparatieverzoek object “SBReservationManager” notificaties te sturen wanneer bepaalde waarden van een nieuw verzoek worden aangepast.

“/Protocols/[SBMaintenanceInterface.h.pdf](#)”

SBManagedObjectInterface (p 32)

Wordt gebruikt om door middel van dezelfde methode naam een bepaalde beschrijving te krijgen van zowel een Defect, Element, Location of Room object.

“/Protocols/[SBManagedObjectInterface.h.pdf](#)”

SBOperationDelegate (p 33)

Wordt gebruikt om notificaties door te sturen over de voortgang van een SBOperation object.

“/Protocols/[SBOperationDelegate.h.pdf](#)”

SBProcessorObserverDelegate (p 34)**SBReservationManagerDelegate (p 35)**

Wordt gebruikt om aan te tonen dat de reservering manager bezig is met het versturen van een nieuw verzoek of verversen van de bestaande verzoeken.

“/Protocols/[SBReservationManagerDelegate.h.pdf](#)”

SBURLOperationDelegate (p 36)

Wordt gebruikt om aan te tonen hoeveel informatie op een bepaald moment is gedownload of wanneer het resultaat beschikbaar is van een URL verzoek.

“/Protocols/[SBURLOperationDelegate.h.pdf](#)”

Models

De volgende bestanden worden gezien als Model objecten, deze bestanden bevinden zich in de “Models” map.

SBAPIRequest (p 37 - 40)

Wordt gebruikt als object om uit te voeren door SBURLOperation objecten zodat altijd de Solware API wordt aangeroepen en eenvoudig POST waardes toevoegen.

“/Models/SBAPIRequest.h.pdf”

“/Models/SBAPIRequest.m.pdf”

SBAPIResponse (p 41 - 44)

Dit object wordt teruggestuurd door een SBURLOperation wanneer de connectie succesvol is uitgevoerd, dit object bevat de resultaten.

“/Models/SBAPIResponse.h.pdf”

“/Models/SBAPIResponse.m.pdf”

SBAvailableDateDescriptor (p 45 - 47)

Dit object dient als container voor een beschikbare datum die gekozen kan worden in het datum selectie scherm.

“/Models/SBAvailableDateDescriptor.h.pdf”

“/Models/SBAvailableDateDescriptor.m.pdf”

SBColorManager (p 52 - 53)

Wordt gebruikt als centraal aanspreekpunt voor alle kleuren die gebruikt worden in de applicatie zoals de achtergrond van een UIViewController.

“/Models/SBColorManager.h.pdf”

“/Models/SBColorManager.m.pdf”

SBCustomerDescriptor (p 54 - 55)

Dient als container voor een woningcorporatie en bevat een identifier en beschrijving hiervan.

“/Models/SBCustomerDescriptor.h.pdf”

“/Models/SBCustomerDescriptor.m.pdf”

SBDayPartDescriptor (p 56 - 58)

Dient als container voor een dagdeel die te kiezen is voor een bepaalde geselecteerde datum.

“/Models/SBDayPartDescriptor.h.pdf”

“/Models/SBDayPartDescriptor.m.pdf”

SBDefectImageDescriptor (p 59 - 62)

Dient als container voor een foto die gekozen is om toe te voegen aan een reparatieverzoek, dit object zorgt ervoor dat het bestand wordt verkleind etc.

[“/Models/SBDefectImageDescriptor.h.pdf”](#)

[“/Models/SBDefectImageDescriptor.m.pdf”](#)

SBFetchResultsControllerAnimation (p 63 - 64)

Dient als animatie container voor de SBCollectionController objecten om ervoor te zorgen dat wanneer objecten veranderen deze animaties tegelijk worden uitgevoerd.

[“/Models/SBFetchResultsControllerAnimation.h.pdf”](#)

[“/Models/SBFetchResultsControllerAnimation.m.pdf”](#)

SBNavigationBarDetailView (p 67 - 70)

Dit object wordt gebruikt om afbeeldingen, titels en beschrijving teksten toe te voegen aan de navigatie balk en zorgt ervoor dat deze objecten juist uitgelijnd zijn.

[“/Models/SBNavigationBarDetailView.h.pdf”](#)

[“/Models/SBNavigationBarDetailView.m.pdf”](#)

SBRelationDescriptor (p 71 - 72)

Dit object wordt gebruikt om een relatie tussen managed objecten te beschrijven die tijdens het synchroniseren nog toegevoegd moeten worden. Dit is belangrijk aangezien eerst alle objecten worden weggeschreven waarna al deze relaties worden uitgevoerd.

[“/Models/SBRelationDescriptor.h.pdf”](#)

[“/Models/SBRelationDescriptor.m.pdf”](#)

SBReservationManager (p 73 - 80)

Dit object beheert alle reparatieverzoeken en stuurt berichten naar luisterende objecten om zo up to date te blijven van alle reparatieverzoeken. Nieuwe objecten worden toegevoegd, gewijzigd of verwijderd aan de hand van dit object.

[“/Models/SBReservationManager.h.pdf”](#)

[“/Models/SBReservationManager.m.pdf”](#)

SBUUser (p 81 - 85)

Dit is een centraal aanspreek punt voor alle wijzigingen die aan het huidige adres en andere persoonlijke gegevens worden uitgevoerd. De informatie wordt gebruikt om reparatieverzoeken te versturen.

[“/Models/SBUUser.h.pdf”](#)

[“/Models/SBUUser.m.pdf”](#)

SBVerificationManager (p 86 - 91)

Dit object wordt gebruikt om ervoor te zorgen dat de applicatie kan communiceren met de REST API door middel van authenticatie, daarnaast worden SSL certificaten gecontroleerd van de server waar de connectie mee wordt gemaakt.

[“/Models/SBVerificationManager.h.pdf”](#)

[“/Models/SBVerificationManager.m.pdf”](#)

SBCollectionViewCellViewLayout (p 48 - 51)

Wordt gebruikt om de layout van een UICollectionViewCell te beschrijven zodat deze goed uitgelijnd worden op het scherm en de juiste vormgeving hebben.

“/Models/SBCollectionViewCellViewLayout.h.pdf”

“/Models/SBCollectionViewCellViewLayout.m.pdf”

SBKeyboardToolbarLayout (p 65 - 66)

Wordt gebruikt om te beschrijven welke knoppen in de Toolbar moeten worden getoond boven een toetsenbord.

“/Models/SBKeyboardToolbarLayout.h.pdf”

“/Models/SBKeyboardToolbarLayout.m.pdf”

NSManagedObjects

De volgende objecten worden beheert door CoreData en zijn permanent opgeslagen op de telefoon. De header en source bestanden bevinden zich in de map “/Models/NSManagedObjects”.

Defect (p 92 - 93)

Dit object beschrijft een defect dat gekozen kan worden door de gebruiker. Dit is de laatste stap van de defect keuze.

“/Models/NSManagedObjects/Defect.h.pdf”

“/Models/NSManagedObjects/Defect.m.pdf”

Element (p 94 - 95)

Dit object beschrijft het element dat leidt naar een defect. Is de derde stap in het defect keuze scherm.

“/Models/NSManagedObjects/Element.h.pdf”

“/Models/NSManagedObjects/Element.m.pdf”

Location (p 96 - 97)

Dit is de eerste stap in het defect keuze scherm en beschrijft de locatie waar het defect plaatsvindt.

“/Models/NSManagedObjects/Location.h.pdf”

“/Models/NSManagedObjects/Location.m.pdf”

Proceeding (p 98 - 100)

Dit object beschrijft een volledige defect beschrijving inclusief afbeelding en opmerking. Een Reservation object kan op deze manier meer van deze objecten bevatten voor eventuele uitbreiding.

“/Models/NSManagedObjects/Proceeding.h.pdf”

“/Models/NSManagedObjects/Proceeding.m.pdf”

Reservation (p 101 - 104)

Dit is het uiteindelijke reparatieverzoek object dat alle Proceeding objecten en de gebruikers informatie bevat. Daarnaast wordt de status van het verzoek hier bijgehouden.

“/Models/NSManagedObjects/[Reservation.h.pdf](#)”

“/Models/NSManagedObjects/[Reservation.m.pdf](#)”

Space (p 105 - 106)

Dit is de tweede stap in de defect keuze en beschrijft de ruimte waar het defect zich plaatsvindt.

“/Models/NSManagedObjects/[Space.h.pdf](#)”

“/Models/NSManagedObjects/[Space.m.pdf](#)”

iOS Compatability (p 107 - 156)

In de map “/Models/iOS Compatability” bevinden zich alle bestanden die gebruikt worden om voor een bepaalde iOS versie de interface te configureren. De bestandsnamen komen nagenoeg overeen met de ViewController objecten die gebruikt worden in de applicatie, vooral de oudere view controllers maken hier gebruik van.

Operation Management

Hier worden de verschillende operatie management objecten beschreven die gebruikt worden om operaties in te plannen, uit te voeren en te beheren. De bestanden bevinden zich in de map “/Models/Operation Management”.

SBFileOperation (p 157 - 158)

Dit object wordt gebruikt als abstractie voor onderliggende bestand actie objecten, zoals het wegschrijven van een afbeelding naar de juiste locatie op het bestandssysteem.

“/Models/Operation Management/[SBFileOperation.h.pdf](#)”

“/Models/Operation Management/[SBFileOperation.m.pdf](#)”

SBFileThumbnailOperation (p 159 - 162)

Dit object wordt gebruikt om van een bestaande afbeelding een verkleining te maken die geschikt is om te worden verstuurd over het internet.

“/Models/Operation Management/[SBFileThumbnailOperation.h.pdf](#)”

“/Models/Operation Management/[SBFileThumbnailOperation.m.pdf](#)”

SBIImageFileSaveOperation (163 - 164)

Dit object schrijft een afbeelding weg naar de daartoe gewezen locatie op het bestandssysteem.

“/Models/Operation Management/[SBIImageFileSaveOperation.h.pdf](#)”

“/Models/Operation Management/[SBIImageFileSaveOperation.m.pdf](#)”

SBIInternetCenter (165 - 169)

Dit object beheert de beschikbaarheid van het internet en het aantal objecten dat op dit moment gebruik maken van het internet.

“/Models/Operation Management/[SBIInternetCenter.h.pdf](#)”

“/Models/Operation Management/[SBIInternetCenter.m.pdf](#)”

SBOperation (170 - 174)

Dit is de abstracte klasse waar alle operatie object van overerven en bevat de standaard functionaliteit om een operatie in de achtergrond uit te voeren.

“/Models/Operation Management/[SBOperation.h.pdf](#)”

“/Models/Operation Management/[SBOperation.m.pdf](#)”

SBOperationManager (p 175 - 178)

Dit object beheert alle operatie objecten en start ze wanneer nodig, daarnaast zorgt dit object ervoor dat de operaties worden uitgevoerd aan de hand van de processor.

“/Models/Operation Management/[SBOperationManager.h.pdf](#)”

“/Models/Operation Management/[SBOperationManager.m.pdf](#)”

SBProcessorObserver (p 179 - 182)

Dit object wordt gebruikt in het Proof Of Concept om te controleren wat de huidige processor activiteit is en om notificaties hiervan door te sturen.

“/Models/Operation Management/[SBProcessorObserver.h.pdf](#)”

“/Models/Operation Management/[SBProcessorObserver.m.pdf](#)”

SBURLOperation (p 189 - 193)

Dit is een object dat gebruikt kan worden om informatie van de REST API server te downloaden door middel van SBAPIRequest objecten. Het dient ook als abstracte klasse voor objecten die meer acties benodigen om als URLOperation te voldoen.

“/Models/Operation Management/[SBURLOperation.h.pdf](#)”

“/Models/Operation Management/[SBURLOperation.m.pdf](#)”

SBURLReservationOperation (p 194 - 198)

Dit is een subklasse van de SBURLOperation en verwerkt een volledig Reservation object naar de REST API.

“/Models/Operation Management/[SBURLreservationOperation.h.pdf](#)”

“/Models/Operation Management/[SBURLreservationOperation.m.pdf](#)”

SBURLSynchronizeOperation (p 199 - 205)

Dit object synchroniseert alle informatie van defect keuzes die op de REST API te vinden zijn met de informatie die lokaal staat op de applicatie.

“/Models/Operation Management/[SBURLSynchronizeOperation.h.pdf](#)”

“/Models/Operation Management/[SBURLSynchronizeOperation.m.pdf](#)”

UI Elementen

Dit hoofdstuk bevat alle customized interface elementen die ontwikkeld zijn voor de mobiele applicatie. De bestanden bevinden zich in de map “/UI Elementen/”.

SBCollectionController (p 206 - 211)

Dit object werd in een oudere versie gebruikt om de verschillende keuzes te tonen van bijvoorbeeld de Locatie of Defect objecten. Het zorgt ervoor dat de keuzes netjes op het scherm wordt getoond.

“/UI Elementen/[SBCollectionController.h.pdf](#)”

“/UI Elementen/[SBCollectionController.m.pdf](#)”

SBCollectionTableViewCellTitleView (p 212 - 213)

Dit object beschrijft de layout en titel die gebruikt wordt voor een bepaalde UICollectionViewCell die het bovenstaande object bevat.

“/UI Elementen/[SBCollectionTableViewCellTitleView.h.pdf](#)”

“/UI Elementen/[SBCollectionTableViewCellTitleView.m.pdf](#)”

SBCollectionViewLocationCell (p 214 - 215)

Dit zijn speciale UICollectionViewCell objecten die gebruikt worden in de nieuwste versie om de eerste vier locaties te tonen op het scherm.

“/UI Elementen/[SBCollectionViewLocationCell.h.pdf](#)”

“/UI Elementen/[SBCollectionViewLocationCell.m.pdf](#)”

SBElementCollectionViewCell (p 216 - 218)

Dit is een UICollectionViewCell die in oudere versies van de code werd gebruikt in de SBCollectionController om de keuzes te representeren.

“/UI Elementen/[SBElementCollectionViewCell.h.pdf](#)”

“/UI Elementen/[SBElementCollectionViewCell.m.pdf](#)”

SBErrorView (p 219 - 221)

Dit UIView object maakt het mogelijk om een foutmelding, laad melding inclusief activiteit indicatie of waarschuwing te tonen op het scherm. Het zorgt ervoor dat de content netjes is uitgelijnd.

“/UI Elementen/[SBErrorView.h.pdf](#)”

“/UI Elementen/[SBErrorView.m.pdf](#)”

SBHomeTableViewCell (p 220 - 224)

In de nieuwste versie van de mobiele applicatie wordt dit object gebruikt om de opties te tonen in het hoofdscherm.

“/UI Elementen/[SBHomeTableViewCell.h.pdf](#)”

“/UI Elementen/[SBHomeTableViewCell.m.pdf](#)”

SKeyboardToolbar (p 227 - 229)

Dit object is de balk die getoond wordt boven een toetsenbord om extra keuzes mogelijk te maken.

“/UI Elementen/[SKeyboardToolbar.h.pdf](#)”

“/UI Elementen/[SKeyboardToolbar.m.pdf](#)”

SNavigationBar_iOS_6_0 (p 230 - 231)

Dit object wordt gebruikt voor de navigatie balk in de iOS 6 versie van de mobiele applicatie en overerft van het SNavigationBar object.

“/UI Elementen/[SNavigationBar_iOS_6_0.h.pdf](#)”

“/UI Elementen/[SNavigationBar_iOS_6_0.m.pdf](#)”

SNavigationBar_iOS_7_0 (p 232 - 233)

Dit object wordt gebruikt voor de navigatie balk in de iOS 7 versie van de mobiele applicatie en overerft van het SNavigationBar object.

“/UI Elementen/[SNavigationBar_iOS_7_0.h.pdf](#)”

“/UI Elementen/[SNavigationBar_iOS_7_0.m.pdf](#)”

SNavigationBar (p 234 - 237)

Dit abstracte object bevat de basis functionaliteit van de navigatie balk die benodigd is voor elke iOS versie.

“/UI Elementen/[SNavigationBar.h.pdf](#)”

“/UI Elementen/[SNavigationBar.m.pdf](#)”

SNavigationControllerExtended (p 238 - 239)

Dit object werd in een oude versie gebruikt om ervoor te zorgen dat een actie kan worden opgevangen en geannuleerd zodat een nette animatie kan worden uitgevoerd.

“/UI Elementen/[SNavigationControllerExtended.h.pdf](#)”

“/UI Elementen/[SNavigationControllerExtended.m.pdf](#)”

SReservationOverviewCell (p 240 - 241)

Dit object wordt gebruikt als overzicht tabel van alle reparatieverzoeken in de nieuwste versie.

“/UI Elementen/[SReservationOverviewCell.h.pdf](#)”

“/UI Elementen/[SReservationOverviewCell.m.pdf](#)”

SSelectableOptionTableViewCell (p 244 - 245)

Dit object wordt gebruikt in de defect keuze stappen om ervoor te zorgen dat de laatste drie stappen dezelfde cell vormgeving gebruiken maar met andere inhoud.

“/UI Elementen/[SSelectableOptionTableViewCell.h.pdf](#)”

“/UI Elementen/[SSelectableOptionTableViewCell.m.pdf](#)”

STextField (p 246 - 247)

Dit is een overerfde klasse van de standaard UITextField klasse en voegt extra functionaliteit toe om een textveld te identificeren wanneer het zich in een tabel bevindt.

“/UI Elementen/[SBTextField.h.pdf](#)”
“/UI Elementen/[SBTextField.m.pdf](#)”

SSToolBar (p 248 - 249)

Dit object wordt alleen gebruikt in de iOS 6 versie van de applicatie en kleurt de onderstaande balk de juiste kleur.

“/UI Elementen/[SSToolBar.h.pdf](#)”
“/UI Elementen/[SSToolBar.m.pdf](#)”

SViewController (p 250 - 252)

Dit object bevat functionaliteit die alle view controller objecten zullen bevatten om ervoor te zorgen dat bijvoorbeeld de achtergrond kleur overal hetzelfde is etc.

“/UI Elementen/[SViewController.h.pdf](#)”
“/UI Elementen/[SViewController.m.pdf](#)”

View Controllers

In dit hoofdstuk worden alle nieuwe en oude UIViewController objecten die we gebruiken voor de schermen in de mobiele applicatie beschreven. De bestanden zijn te vinden in de map “/ViewController”.

Huidige ViewControllers

SBAIReservationsViewController (p 253 - 258)

Dit object is deel van de vormgegeven applicatie en toont een lijst van alle reparatieverzoeken die de gebruiker heeft verstuurd.

“/ViewControllers/SBAIReservationsViewController.h.pdf”

“/ViewControllers/SBAIReservationsViewController.m.pdf”

SBAAppointmentViewController (p 259 - 266)

Dit object is deel van de vormgegeven applicatie en maakt het de gebruiker mogelijk om een datum en dagdeel combinatie uit te kiezen voor de afspraak.

“/ViewControllers/SBAAppointmentViewController.h.pdf”

“/ViewControllers/SBAAppointmentViewController.m.pdf”

SBDefectSelectionViewController (p 267 - 271)

Deel van de nieuwe vormgeving en laat de gebruiker een keuze maken tussen defecten.

“/ViewControllers/SBDefectSelectionViewController.h.pdf”

“/ViewControllers/SBDefectSelectionViewController.m.pdf”

SBElementSelectionViewController (p 272 - 276)

Deel van de nieuwe vormgeving en laat de gebruiker een keuze maken tussen elementen.

“/ViewControllers/SBElementSelectionViewController.h.pdf”

“/ViewControllers/SBElementSelectionViewController.m.pdf”

SBHomeViewController (p 277 - 280)

Dit is in de nieuwe vormgeving het hoofdscherm waaruit de gebruiker verschillende acties kan uitvoeren zoals toevoegen en beheren van reparatieverzoeken.

“/ViewControllers/SBHomeViewController.h.pdf”

“/ViewControllers/SBHomeViewController.m.pdf”

SBLocationSelectionViewController (p 281 - 285)

Deel van de nieuwe vormgeving en laat de gebruiker een keuze maken tussen locaties.

“/ViewControllers/SBLocationSelectionViewController.h.pdf”

“/ViewControllers/SBLocationSelectionViewController.m.pdf”

SBOptionalInformationViewController (p 286 - 294)

Deel van de nieuwe vormgeving en laat de gebruiker optionele informatie opgeven, namelijk een opmerking en/of afbeelding.

["/ViewControllers/SBOptionalInformationViewController.h.pdf"](#)
["/ViewControllers/SBOptionalInformationViewController.m.pdf"](#)

SBPersonalDataViewController (p 295 - 307)

In dit scherm kan de gebruiker alle benodigde informatie toevoegen zoals emailadres en andere adresgegevens. Dit scherm is ook deel van de nieuwe vormgeving.

["/ViewControllers/SBPersonalDataViewController.h.pdf"](#)
["/ViewControllers/SBPersonalDataViewController.m.pdf"](#)

SBReservationOverviewViewController (p 370 - 377)

Deze viewcontroller is ook deel van de nieuwe vormgeving en laat de gebruiker een overzicht zien van het nieuwe of bestaande verzoek.

["/ViewControllers/SBReservationOverviewViewController.h.pdf"](#)
["/ViewControllers/SBReservationOverviewViewController.m.pdf"](#)

SBRoomSelectionViewController (p 378 - 382)

Deel van de nieuwe vormgeving en laat de gebruiker een keuze maken tussen ruimtes.

["/ViewControllers/SBRoomSelectionViewController.h.pdf"](#)
["/ViewControllers/SBRoomSelectionViewController.m.pdf"](#)

Oude ViewControllers

SBPhoneCustomerViewController (p 308 - 313)

Dit is een oude view controller die individueel gebruikt werd om een woningcorporatie te selecteren voor de gebruikers gegevens.

["/ViewControllers/SBPhoneCustomerViewController.h.pdf"](#)
["/ViewControllers/SBPhoneCustomerViewController.m.pdf"](#)

SBPhoneDateViewController (p 314 - 321)

In deze oude view controller was het mogelijk om in tabel vorm met uitklappende animaties een datum en dagdeel combinatie te selecteren om de afspraak op in te plannen.

["/ViewControllers/SBPhoneDateViewController.h.pdf"](#)
["/ViewControllers/SBPhoneDateViewController.m.pdf"](#)

SBPhoneDefectViewController (p 322 - 324)

In dit oude viewcontroller object was het voor de gebruiker mogelijk om via selecteerbare tegels een defect te kiezen.

["/ViewControllers/SBPhoneDefectViewController.h.pdf"](#)
["/ViewControllers/SBPhoneDefectViewController.m.pdf"](#)

SBPhoneDescriptionViewController (p 325 - 329)

In deze view controller was het in de oude versie mogelijk om een aantal opmerkingen toe te voegen aan het verzoek.

["/ViewControllers/SBPhoneDescriptionViewController.h.pdf"](#)

["/ViewControllers/SBPhoneDescriptionViewController.m.pdf"](#)

SBPhoneElementViewController (p 330 - 332)

In dit oude viewcontroller object was het voor de gebruiker mogelijk om via selecteerbare tegels een element te kiezen.

["/ViewControllers/SBPhoneElementViewController.h.pdf"](#)

["/ViewControllers/SBPhoneElementViewController.m.pdf"](#)

SBPhoneImageViewController (p 333 - 339)

Dit scherm is in de nieuwe versie samengevoegd met de opmerking toevoegen scherm en maakt het mogelijk om een afbeelding te kiezen en toe te voegen aan het verzoek.

["/ViewControllers/SBPhoneImageViewController.h.pdf"](#)

["/ViewControllers/SBPhoneImageViewController.m.pdf"](#)

SBPhoneLocationViewController (p 340 - 342)

In dit oude viewcontroller object was het voor de gebruiker mogelijk om via selecteerbare tegels een locatie te kiezen.

["/ViewControllers/SBPhoneLocationViewController.h.pdf"](#)

["/ViewControllers/SBViewController.m.pdf"](#)

SBPhoneLoginViewController (p 343 - 355)

Dit is het hoofdscherm in de oude versie, in deze versie werd gelijk gevraagd om de persoonlijke gegevens. Dit is nu verplaatst naar een andere viewcontroller.

["/ViewControllers/SBPhoneLoginViewController.h.pdf"](#)

["/ViewControllers/SBPhoneLoginViewController.m.pdf"](#)

SBPhoneOverviewViewController (p 356 - 366)

In dit scherm was het in een oudere versie mogelijk om een overzicht van de reparatieverzoeken te bekijken.

["/ViewControllers/SBPhoneOverviewViewController.h.pdf"](#)

["/ViewControllers/SBPhoneOverviewViewController.m.pdf"](#)

SBPhoneSpaceViewController (p 367 - 369)

In dit oude viewcontroller object was het voor de gebruiker mogelijk om via selecteerbare tegels een ruimte te kiezen.

["/ViewControllers/SBPhoneSpaceViewController.h.pdf"](#)

["/ViewControllers/SBPhoneSpaceViewController.m.pdf"](#)

XML Interface bestanden

De volgende bestanden zijn XML representaties van bestanden die het vereenvoudigen om interfaces te ontwikkelen op het iOS platform. Door middel van het programma Xcode wordt een XML bestand gegenereerd. De bestanden zijn te vinden in de map “/XML Interface”.

PhoneStoryboard (p 383 - 393)

In dit bestand bevindt zich de gehele flow van de mobiele applicatie, het beschrijft van welk scherm naar een ander scherm kan worden genavigeerd en hou deze schermen in elkaar steken kwa vormgeving en uitlijning etc.

“/XML Interface/[PhoneStoryboard.storyboard.pdf](#)”

“/XML Interface/[PhoneStoryboard.storyboard.pdf](#)”

SBCollectionViewLocationCell (p 394)

Dit bestand is een vormgevende beschrijven van een tabel cell voor een SBCollectionController object.

“/XML Interface/[SBCollectionViewLocationCell.xib.pdf](#)”

“/XML Interface/[SBCollectionViewLocationCell.xib.pdf](#)”

SBHomeTableViewCell (p 395)

Dit is de vormgeving voor de hoofdscherm cellen.

“/XML Interface/[SBHomeTableViewCell.xib.pdf](#)”

“/XML Interface/[SBHomeTableViewCell.xib.pdf](#)”

SBReservationOverviewCell (p 396)

Dit is de vormgeving die gebruikt wordt voor reparatieverzoek overzicht tabel cellen.

“/XML Interface/[SBReservationOverviewCell.xib.pdf](#)”

“/XML Interface/[SBReservationOverviewCell.xib.pdf](#)”

SBSelectableOptionTableViewCell (397 - 398)

dit is de vormgeving die wordt gebruikt in de keuze tabellen voor Space, Element en Defect objecten.

“/XML Interface/[SBSelectableOptionTableViewCell.xib.pdf](#)”

“/XML Interface/[SBSelectableOptionTableViewCell.xib.pdf](#)”

REST API

In dit hoofdstuk worden de gebruikte objecten voor de werking van de REST API beschreven. Dit bestaat uit twee onderdelen namelijk de methodes die aangeroepen kunnen worden op de API en de objecten die de API gebruikt om informatie te verwerken en terug te sturen.

API Methodes

OAMobileAddMaintenanceAPIMethod (p 400 - 401)

Deze methode maakt het mogelijk om een nieuw defect aan een reparatieverzoek toe te voegen. Dat wil zeggen dat er al een reparatieverzoek is aangemaakt voordat deze methode wordt aangeroepen, per aanroep wordt een nieuw defect toegevoegd.

“/REST API/[OAMobileAddMaintenanceAPIMethod.php.pdf](#)”

OAMobileAddressLookUpAPIMethod (p 402)

Deze methode ontvangt postcode en huisnummer gegevens om de adresgegevens verder van op te zoeken en te valideren of het daadwerkelijk bestaat.

“/REST API/[OAMobileAddressLookUpAPIMethod.php.pdf](#)”

OAMobileAddUserAPIMethod (p 403 - 405)

Deze methode maakt in eerste instantie het reparatieverzoek aan met alle persoonlijke gegevens en adresgegevens. Hier wordt later door middel van de bovenste API methode content aan toegevoegd voor defecten.

“/REST API/[OAMobileAddUserAPIMethod.php.pdf](#)”

OAMobileAllCustomerAPIMethod (p 406)

Deze methode stuurt alle mogelijke woningcorporaties terug.

“/REST API/[OAMobileAllCustomerAPIMethod.php.pdf](#)”

OAMobileAuthenticationRequestAPIMethod (p 407)

Deze methode maakt nieuwe authenticatie gegevens aan voor een nieuwe telefoon die wil communiceren met de applicatie.

“/REST API/[OAMobileAuthenticationRequestAPIMethod.php.pdf](#)”

OAMobileAvailableDatesAPIMethod (p 408 - 409)

Deze methode stuurt alle mogelijke datums en dagdelen terug die de gebruiker kan reserveren.

“/REST API/[OAMobileAvailableDatesAPIMethod.php.pdf](#)”

OAMobileCancelReservationAPIMethod (p 410)

Deze methode annuleert een reeds toegevoegd reparatieverzoek en stuurt een mailtje naar de gebruiker met het succes bericht.

“/REST API/[OAMobileCancelReservationAPIMethod.php.pdf](#)”

OAMobileCheckStateAPIMethod (p 411 - 412)

Deze methode ontvangt een lijst met reparatieverzoek nummers waarvan de status gesynchroniseerd moet worden. De methode stuurt de huidige status van de verzoeken terug.

“/REST API/[OAMobileCheckStateAPIMethod.php.pdf](#)”

OAMobileImageAPIMethod (p 413)

Deze methode wordt niet meer gebruikt door de nieuwe versie van de mobiele applicatie aangezien er geen foto's worden getoond. Deze methode diende ervoor om een afbeelding terug te sturen voor een bepaalde Locatie, Space, Element of Defect.

“/REST API/[OAMobileImageAPIMethod.php.pdf](#)”

OAMobileSynchronizeDefectsAPIMethod (p 414)

Deze methode stuurt alle huidige Locaties, Spaces, Elementen en Defecten terug die op dat moment in de database staan wanneer de mobiele applicatie deze dient te synchroniseren.

“/REST API/[OAMobileSynchronizeDefectsAPIMethod.php.pdf](#)”

Gebruikte objecten

OM_AddressLookUp p 415 - 416)

Dit object wordt gebruikt om de connectie te openen met de postcode database om het verkregen postcode huisnummer combinatie te controleren op juistheid.

“/REST API/[OM_AddressLookUp.php.pdf](#)”

OM_AddressLookUpLocation (p 417)

Dit object dient als cache en wrapper voor de verkregen informatie van de postcode database. Dit object wordt opgeslagen in onze database om te dienen als cache zodat niet voor dezelfde adressen de postcode database benaderd hoeft te worden.

“/REST API/[OM_AddressLookUpLocation.php.pdf](#)”

OM_DayPart (p 418)

Dit object is opgeslagen in de database en beschrijft alle mogelijke dagdelen die geselecteerd kunnen worden. Op moment van schrijven zijn dit er 8.

“/REST API/[OM_DayPart.php.pdf](#)”

OM_Defect (p 419)

Dit object wordt ook opgeslagen in de database en beschrijft een mogelijk defect die geselecteerd kan worden in de database.

“/REST API/[OM_Defect.php.pdf](#)”

OM_Element (p 420)

Dit object wordt opgeslagen in de database en beschrijft een mogelijk element die geselecteerd kan worden in de database.

“/REST API/[OM_Element.php.pdf](#)”

OM_Location (p 421)

Dit object wordt opgeslagen in de database en beschrijft een mogelijke locatie die geselecteerd kan worden in de database.

“/REST API/[OM_Location.php.pdf](#)”

OM_Maintenance (p 422)

Dit is een object dat het Proceeding in de mobiele applicatie voorstelt, dat betekent dat het de gekozen defect bevat naast opmerkingen en een afbeelding.

“/REST API/[OM_Maintenance.php.pdf](#)”

OM_MaintenanceUser (p 423 - 424)

Dit object is een reparatieverzoek dat alle informatie van de gebruiker bevat en mogelijk een of meerdere OM_Maintenance objecten.

“/REST API/[OM_MaintenanceUser.php.pdf](#)”

OM_Space (p 425)

Dit object wordt opgeslagen in de database en beschrijft een mogelijke ruimte die geselecteerd kan worden in de database.

“/REST API/[OM_Space.php.pdf](#)”

OM_UuidAuthentication (p 426 - 427)

Dit object wordt opgeslagen in de database en beschikt over authenticatie gegevens die gebruikt kunnen worden in de mobiele applicatie. Dit object is gekoppeld aan een bepaalde gebruiker om ervoor te kunnen zorgen dat alleen de maker van het object er toegang tot heeft.

“/REST API/[OM_UuidAuthentication.php.pdf](#)”

UuidAuthenticationProvider (p 428)

Dit object wordt gebruikt om in de database authenticatie gegevens te valideren en te kijken of de gebruiker toegang heeft tot een bepaalde methode.

“/REST API/[UuidAuthenticationProvider.php.pdf](#)”

Beheersysteem

OM_AvailabilityController (p 429 - 431)

Dit object beheert de opties die uit te voeren zijn met beschikbaarheid objecten.

“/Beheer/[OM_AvailabilityController.php.pdf](#)”

OM_AvailabilityForm (p 432)

Dit formulier maakt het mogelijk beschikbaarheid te wijzigen.

“/Beheer/[OM_AvailabilityForm.php.pdf](#)”

OM_AvailabilityOverviewForm (p 433 - 434)

Dit formulier maakt het mogelijk om een overzicht van beschikbaarheid te tonen.

“/Beheer/[OM_AvailabilityOverviewForm.php.pdf](#)”

OM_DayPartController (p 435)

Dit object beheert alle opties die met dagdelen uitgevoerd kunnen worden.

“/Beheer/[OM_DayPartController.php.pdf](#)”

OM_DayPartOverviewForm (p 436)

Dit formulier toont een overzicht van dagdelen.

“/Beheer/[OM_DayPartOverviewForm.php.pdf](#)”

OM_MaintenanceController (p 437 - 443)

Dit object beheert de opties die uit te voeren zijn met specifieke reparatie objecten.

“/Beheer/[OM_MaintenanceController.php.pdf](#)”

OM_MaintenanceUserController (p 444 - 446)

Dit object beheert de opties die uit te voeren zijn met reparatieverzoek objecten.

“/Beheer/[OM_MaintenanceUserController.php.pdf](#)”

OM_MaintenanceUserForm (p 447)

Dit formulier maakt het mogelijk om reparatieverzoeken te wijzigen.

“/Beheer/[OM_MaintenanceUserForm.php.pdf](#)”

OM_MaintenanceUserMaintenanceOverviewForm (p 448)

Dit formulier toont alle specifieke reparaties onder een reparatieverzoek.

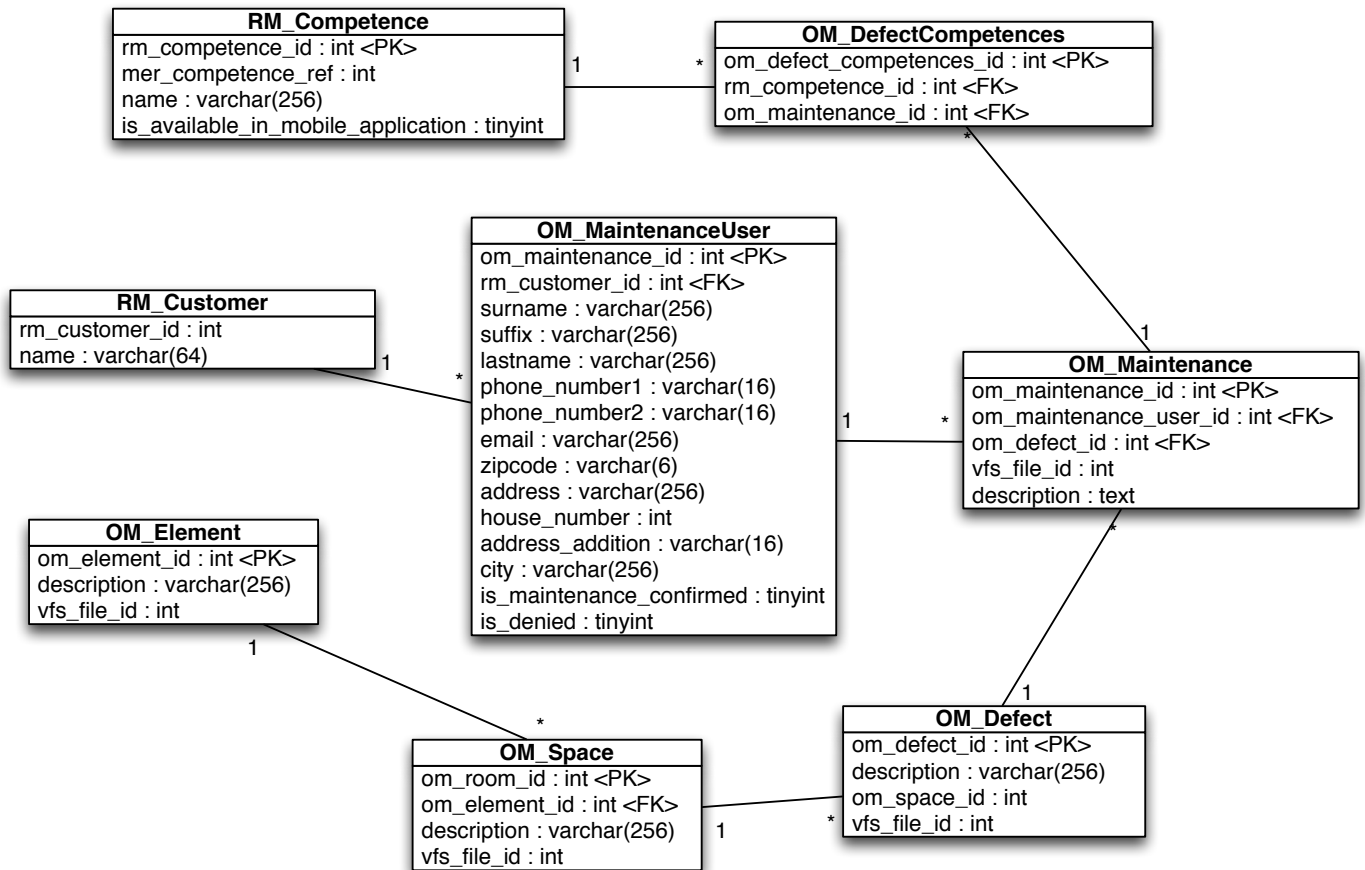
“/Beheer/[OM_MaintenanceUserMaintenanceOverviewForm.php.pdf](#)”

OM_MaintenanceUserStateHistoryOverviewForm (p 449)

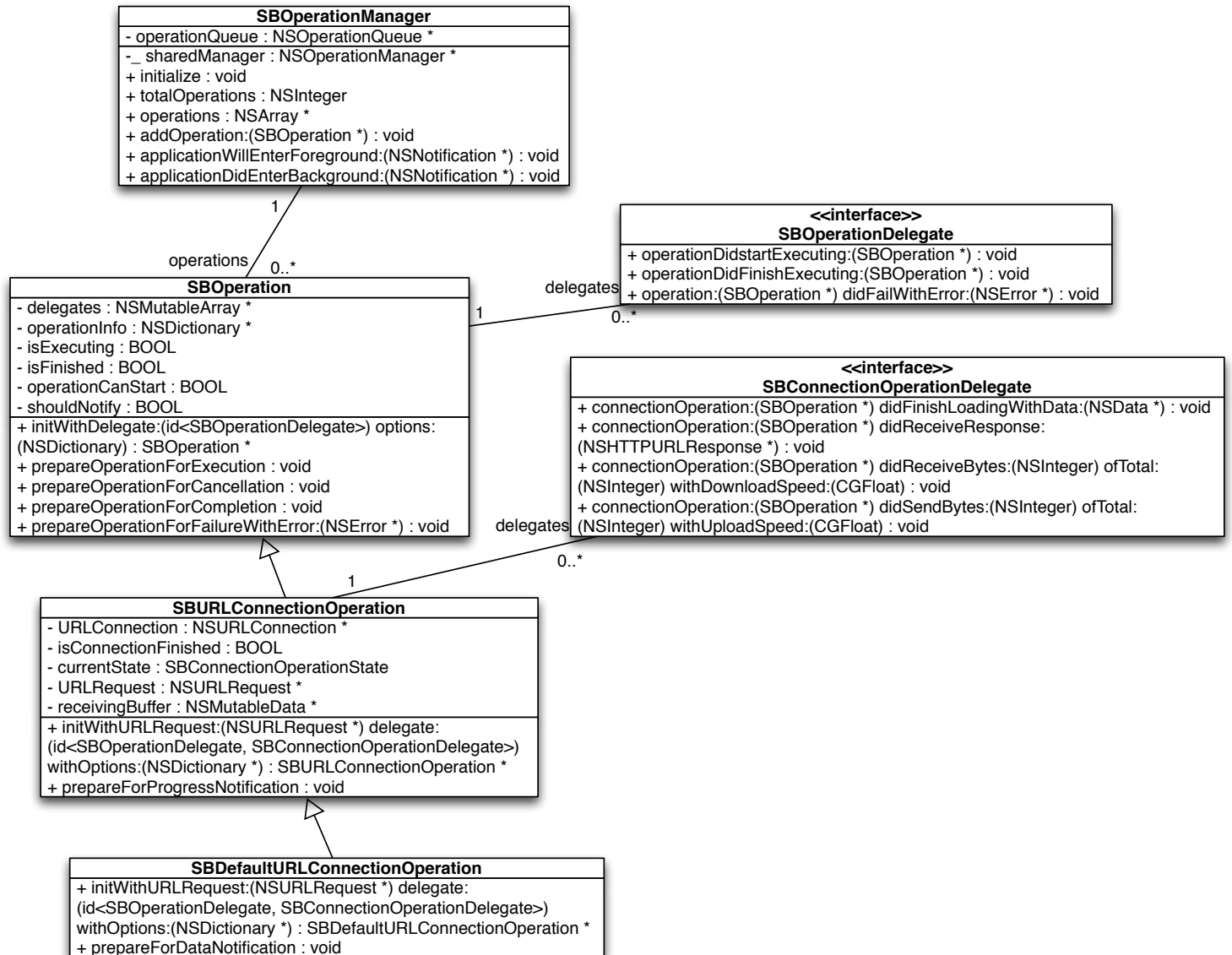
Dit formulier toont alle status wijzigingen van een specifiek reparatieverzoek.

“/Beheer/OM_MaintenanceUserStateHistoryOverviewForm.php.pdf”

Bijlage I



Bijlage J



Bijlage K

Standaard Objective-C guide

In deze bijlage zal een meer gedetailleerde maar korte introductie op het Objective-C en Cocoa Touch platform worden gegeven. Een aantal fundamentele ontwikkel strategieën zullen behandeld worden die het vereenvoudigen om de code te begrijpen en de wijze van programmeren. Aangezien de taal voor een aantal van deze strategieën geoptimaliseerd is zal je in alle code veel gebruik van deze strategieën zien. Zie de onderstaande inhoudsopgave om snel naar bepaalde onderdelen en strategieën te gaan.

Xcode	3
<i>Objective-C</i>	3
<i>Cocoa Touch</i>	3
Fundamentele objecten	4
<i>NSObject</i>	4
<i>NSRunLoop</i>	4
<i>UIViewController</i>	4
<i>NSString</i>	4
<i>NSArray</i>	5
Core Strategieën	5
<i>Model-View-Controller</i>	5
<i>Key-Value Observing</i>	5
<i>Delegatie</i>	6
CoreData	6

Xcode

Xcode¹ is het programma dat gebruikt wordt om applicaties te ontwikkelen voor zowel het Mac OSX als het iOS platform (het platform voor dit project). Onder iOS vallen onder andere iPhones, iPads en iPod Touches. Op dit moment is er geen andere manier van ontwikkelen beschikbaar die alle functies ondersteund zoals Xcode dat doet voor deze platformen. Het programma kan via een ontwikkelaar account van Apple gedownload worden, voor dit account zal een bepaald bedrag per jaar moeten worden. Voor dit bedrag krijg je de mogelijkheid om geschreven applicaties te versturen naar Apple, de applicatie zal gecontroleerd worden en wanneer goedgekeurd in de legale App Store geplaatst worden. Als de applicatie beschikbaar is in de App Store kunnen gebruikers de applicatie downloaden en ook daadwerkelijk gebruiken.

Objective-C

Objective-C² is een programmeer taal die gebruikt dient te worden om applicaties voor OSX en iOS te ontwikkelen. De taal is een zogenaamde “superset” van de taal C, dat betekent dat tussen Objective-C ook normale C code geschreven kan worden. Een groot verschil tussen de twee talen is het feit dat Objective-C object georiënteerd is waarnaast C dat niet is.

Om door middel van Objective-C een methode te definiëren wordt dat op de volgende wijze gedaan. In dit voorbeeld wordt een methode geschreven die als argument een *NSString** object nodig heeft en void als return waarde heeft. Onder de methode wordt een voorbeeld beschreven die aantoont hoe deze methode op het *self* (Objective-C versie van *this*) object word aangeroepen.

```
- (void) prepareString:(NSString *string)
{
    // do something with string
}

NSString *lang = @"Objective-C";
[self prepareString: [NSString stringWithFormat: @"Dit is een %@ string", lang]];
```

Wanneer objecten met de prefix “*NS*” worden gedefinieerd betekent dit dat een object wordt gebruikt dat in de standaard SDK is inbegrepen, deze SDK is genaamd Cocoa Touch en zal verder in dit document behandeld worden. Voor een gedetailleerde uitleg over nagenoeg alle functies van Objective-C verwijs ik je naar de tweede voetnoot van deze pagina.

Cocoa Touch

¹ <https://developer.apple.com/xcode/>

² <https://developer.apple.com/library/mac/documentation/Cocoa/Conceptual/ProgrammingWithObjectiveC/Introduction/Introduction.html>

Cocoa touch³ is de naam van het Framework dat gebruikt wordt om applicaties te ontwikkelen voor iOS. Dit Framework is een gestripte versie van het originele Framework Cocoa⁴ dat gebruikt wordt voor desktop OSX applicaties. Een aantal functies zijn uit dit Framework gehaald en toegevoegd om het specifiek geschikt voor iOS mobiele applicaties te maken. Objecten zoals *NSString* en *NSArray* maken een Objective-C aanpak van programmeren mogelijk zonder al deze functionaliteit zelf te schrijven. Een aantal objecten die gebruikt worden in het project zijn zowel in Cocoa als in Cocoa touch beschikbaar.

Fundamentele objecten

In dit hoofdstuk worden de meest fundamentele objecten van het Cocoa Touch Framework beschreven, deze objecten zullen het meest gebruikt worden tijdens het ontwikkelen en zorgen voor de unieke Objective-C programmeer wijze.

NSObject

Het root object *NSObject*⁵ dat door ieder object overerft wordt in het Cocoa Touch Framework, dit object bevat de basis functionaliteit van elk object zoals vergelijking en klasse identificatie.

NSRunLoop

De *NSRunLoop*⁶ is een thread specifiek object dat input verwerkt van verschillende bronnen. Deze bronnen kunnen bijvoorbeeld timers zijn of wanneer de gebruiker op het scherm een MultiTouch beweging heeft gemaakt. Dit object zorgt er op zijn beurt voor dat de thread deze bron verwerkt door de daartoe bestemde methode aan te roepen. Een andere belangrijke functionaliteit van dit object is het zorgen dat de applicatie in een infinite while loop wordt geplaatst. Per iteratie van deze loop wordt gekeken of een nieuwe bron kan worden verwerkt.

UIViewController

Een *UIViewController*⁷ is een fundamenteel object dat het Model-View-Controller ontwerp van Objective-C mogelijk maakt. Het object is gekoppeld aan een *UIView* object waar alle user interface objecten in staan gedefinieerd. Door dit object te koppelen met een zelf bedacht Model kan aan dit patroon voldaan worden. Meer informatie over dit patroon volgt verder in dit document.

NSString

Het Objective-C standaard string object is *NSString*⁸ waar een groot aantal verschillende types strings kan worden opgeslagen zoals UTF-8 of UTF-16. Met dit object kunnen ook

³ <https://developer.apple.com/technologies/ios/cocoa-touch.html>

⁴ <https://developer.apple.com/technologies/mac/cocoa.html>

⁵ https://developer.apple.com/library/ios/documentation/cocoa/reference/foundation/Classes/NSObject_Class/Reference/Reference.html

⁶ https://developer.apple.com/library/mac/documentation/cocoa/reference/foundation/classes/NSRunLoop_Class/Reference/Reference.html

⁷ https://developer.apple.com/library/ios/documentation/uikit/reference/UIViewController_Class/Reference/Reference.html

⁸ https://developer.apple.com/library/ios/documentation/cocoa/reference/foundation/Classes/NSString_Class/Reference/NSString.html

andere versies van de string worden uitgelezen en verwerkt worden. Dit is een van de objecten die het meeste gebruikt zal worden door de gehele applicatie.

NSArray

Dit is de Objective-C versie van een normale C array, dit object is alleen geschikt voor objecten die overerft zijn van het *NSObject* root object. Standaard C integers of andere typen kunnen niet worden opgeslagen in een array.

Core Strategieën

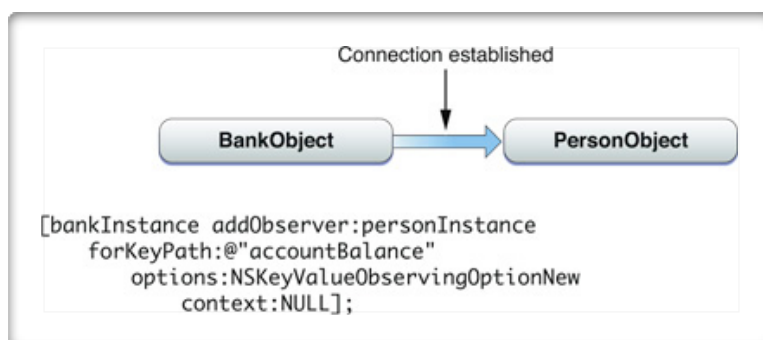
Objective-C in combinatie met Cocoa Touch bevat een waslijst aan core strategieën, in dit hoofdstuk zullen een aantal van deze behandeld worden, daarnaast is het mogelijk om in de Apple documentatie informatie van alle strategieën te lezen⁹.

Model-View-Controller

Dit¹⁰ is een veel voorkomend model dat gebruikt kan worden in applicaties die de user interface van de daadwerkelijke content willen scheiden. Het model bestaat uit een door de programmeur gedefinieerd Model, daarnaast een *UIViewController* object dat de content opvraagt en aan het *UIView* doorstuurt om zo op het scherm aan de gebruiker te tonen. Door gebruik te maken van dit patroon wordt overzichtelijke code gegenereerd waarbij de afhankelijkheden zijn afgescheiden van elkaar. Elk object doet de taak waar het voor gespecialiseerd is.

Key-Value Observing

Met Key-Value Observing¹¹ wordt bedoeld dat een object van een ander object een notificatie wil ontvangen wanneer een gespecificeerde waarde is veranderd. Hierdoor kan het luisterende object up to date blijven over de waarden die een ander object heeft. In onderstaande afbeelding is een voorbeeld uitgebeeld dat een connectie maakt tussen twee verschillende objecten.



Figuur 1: Key-Value Observing connectie maken overzicht¹²

⁹ <https://developer.apple.com/library/mac/documentation/general/conceptual/devpedia-cocoacore>

¹⁰ <https://developer.apple.com/library/mac/documentation/general/conceptual/devpedia-cocoacore/MVC.html>

¹¹ <https://developer.apple.com/library/mac/documentation/general/conceptual/devpedia-cocoacore/KVO.html>

¹² <https://developer.apple.com/library/mac/documentation/Cocoa/Conceptual/KeyValueObserving/KeyValueObserving.html>

Delegatie

Delegatie¹³ laat objecten beslissingen nemen voor andere objecten of op de hoogte blijven van bepaalde status veranderingen etc. Zo kan een downloader bijvoorbeeld aan een delegatie object laten weten dat een bepaald percentage van de download compleet is. Het delegatie object kan dit weer vertalen naar de gebruiker in de user interface. Om delegatie toe te passen wordt in Objective-C gebruikt gemaakt van zogenaamde Protocollen¹⁴, een protocol dient als interface voor objecten om bepaalde methodes te implementeren die aangeroepen kunnen worden door een object op het delegerende object.

CoreData

CoreData¹⁵ is een deel van het Cocoa touch Framework dat gebruikt kan worden om informatie op te slaan op de harde schijf van een mobiele telefoon. Door een model te definiëren die een SQLite database beschrijft kan door middel van de CoreData objecten informatie definitief worden opgeslagen. Het meest belangrijke object in dit Framework is het *NSManagedObject*¹⁶ object, dit is het basis object voor elk ander object dat opgeslagen moet worden in de database en overerft daarom ook hiervan. CoreData is een object georiënteerde opslag methode, er wordt niet gebruik gemaakt van Foreign Keys om relaties tussen objecten te definiëren. Dit wordt allemaal door CoreData in de achtergrond geregeld, de relaties kunnen wel door de programmeur aangegeven worden.

Een ander belangrijk object is het *NSManagedObjectContext*¹⁷, dit object is het aanspreekpunt om objecten toe te voegen of uit te lezen uit de SQLite database. Dit wordt door middel van een *NSPersistentStoreRequest*¹⁸ uitgevoerd. Aan de hand van dit abstracte object kunnen objecten opgeslagen of uitgelezen worden.

CoreData wordt als een moeilijk te implementeren onderdeel van een mobiele applicatie gezien aangezien de meeste activiteit naar de achtergrond threads zal moeten worden verplaatst om de main thread niet te blokkeren. Wanneer dit gebeurt kan de gebruiker van de mobiele applicatie geen acties uitvoeren. Wanneer CoreData in de achtergrond wordt uitgevoerd zullen veel controles moeten worden uitgevoerd of er geen informatie op een foute manier tussen threads wordt uitgewisseld.

¹³ <https://developer.apple.com/library/mac/documentation/general/conceptual/devpedia-cocoacore/Delegation.html>

¹⁴ <https://developer.apple.com/library/mac/documentation/general/conceptual/devpedia-cocoacore/Protocol.html>

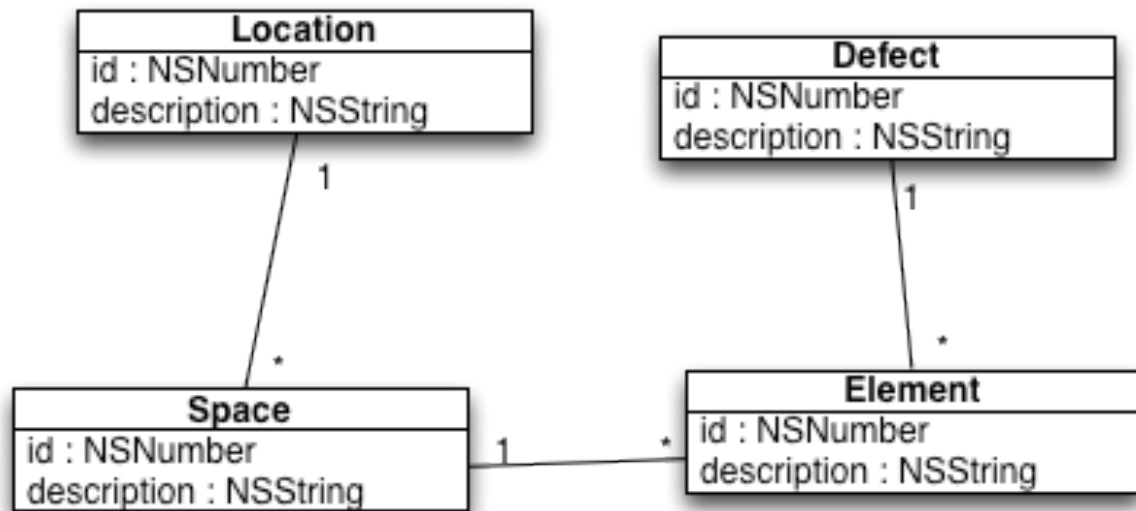
¹⁵ <https://developer.apple.com/library/mac/documentation/cocoa/Conceptual/CoreData/cdProgrammingGuide.html>

¹⁶ https://developer.apple.com/library/ios/documentation/cocoa/reference/CoreDataFramework/Classes/NSManagedObject_Class/Reference/NSManagedObject.html

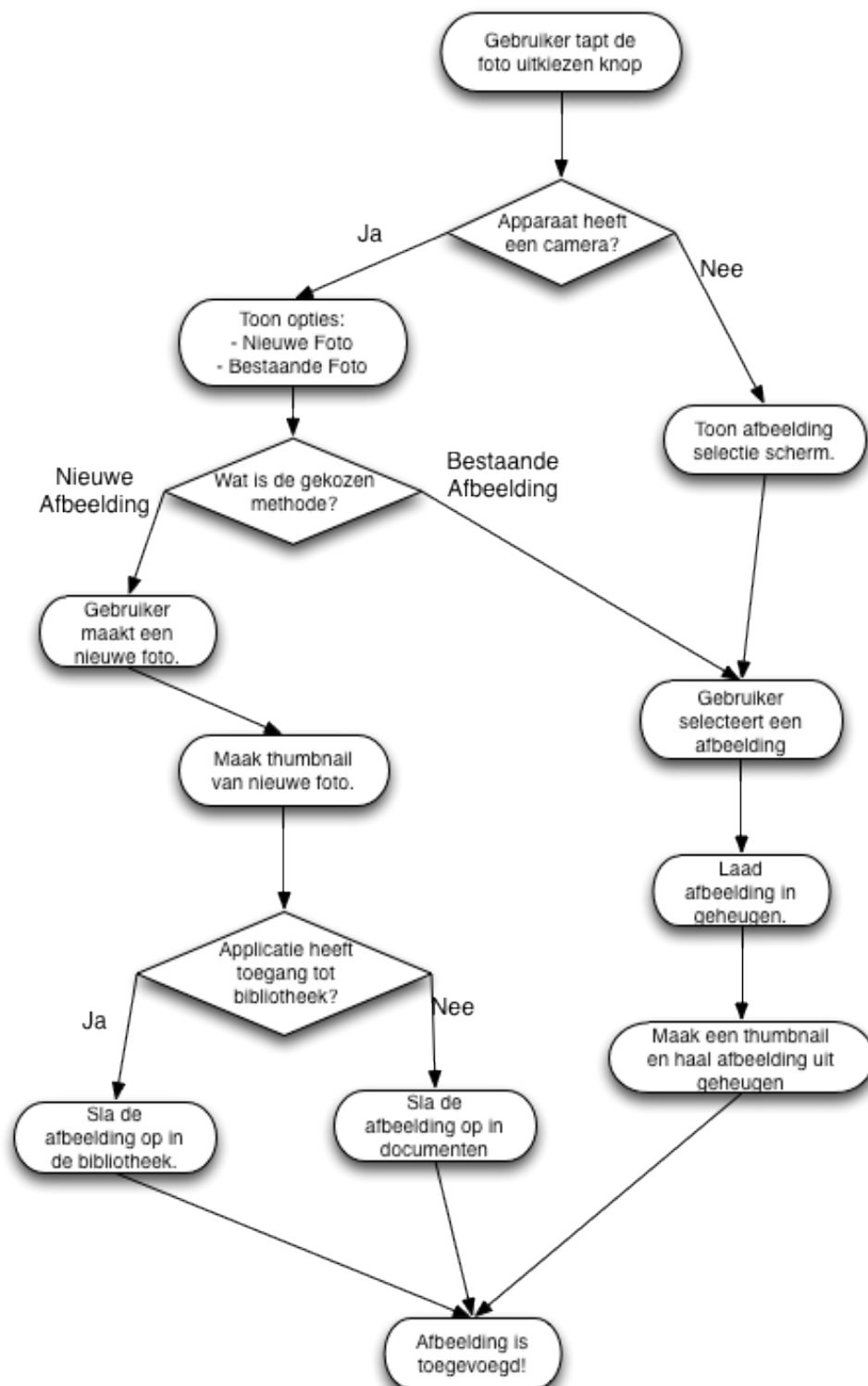
¹⁷ https://developer.apple.com/library/ios/documentation/cocoa/reference/CoreDataFramework/Classes/NSManagedObjectContext_Class/NSManagedObjectContext.html

¹⁸ https://developer.apple.com/library/ios/documentation/CoreData/Reference/NSPersistentStoreRequest_Class/NSPersistentStoreRequest.html

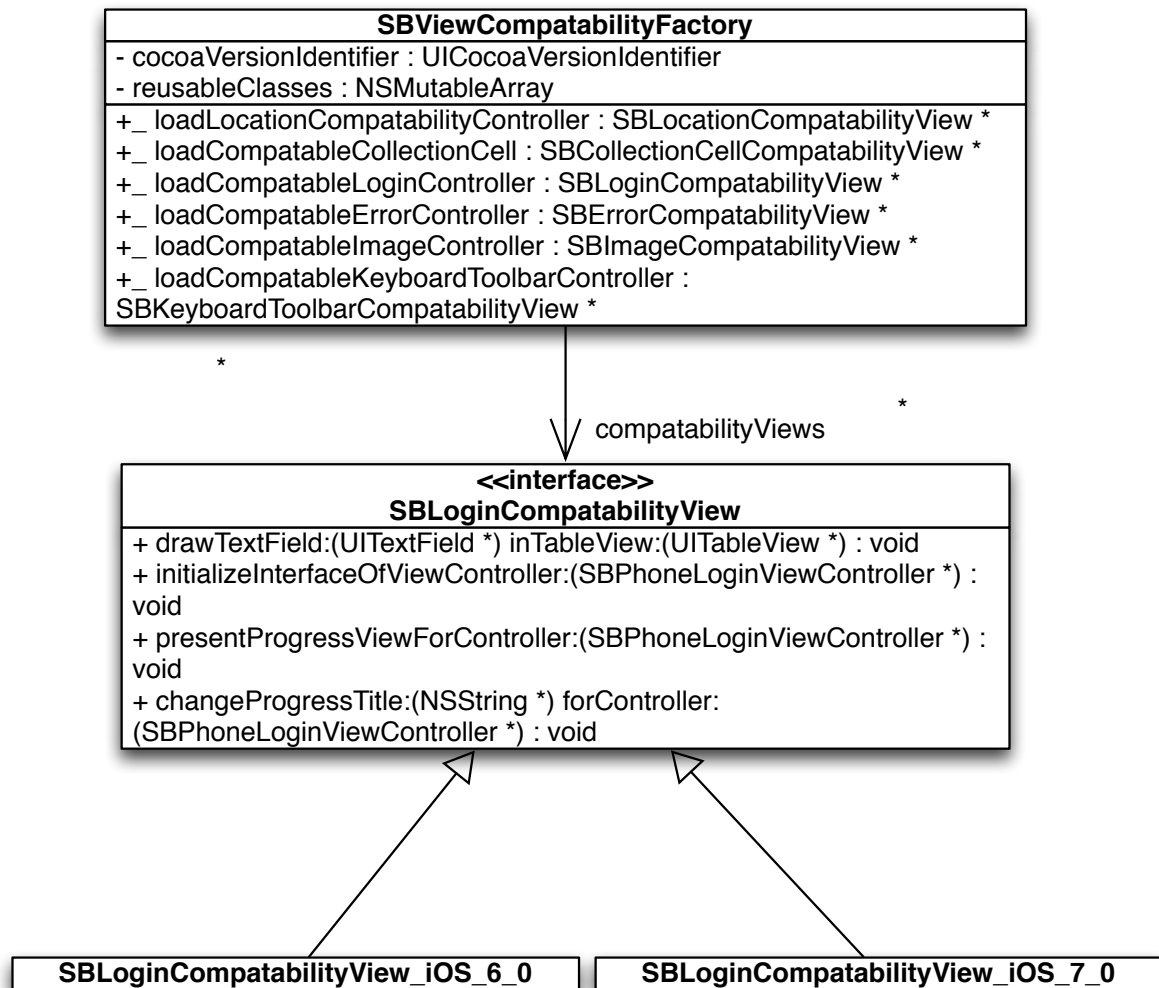
Bijlage L



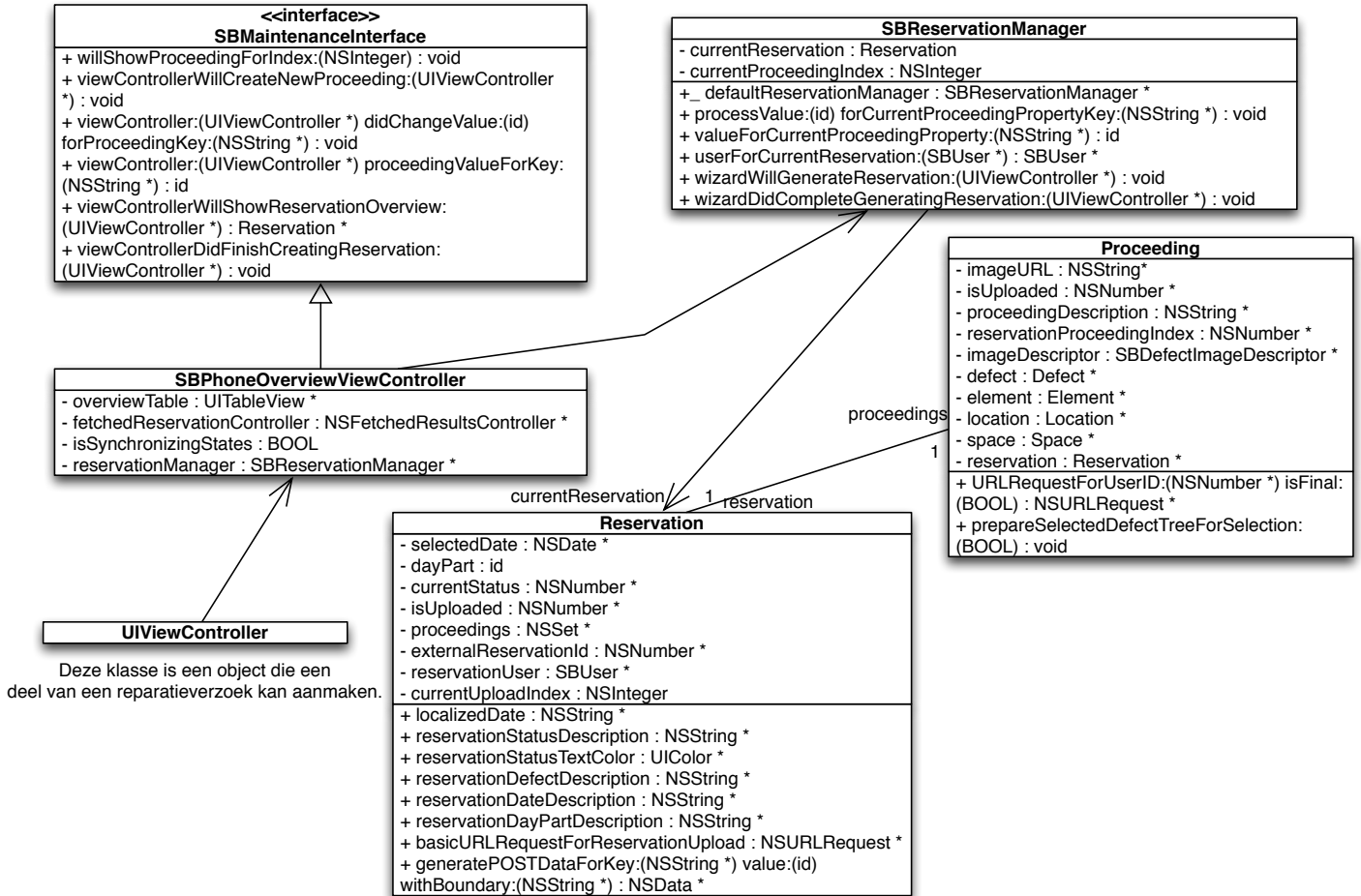
Bijlage M



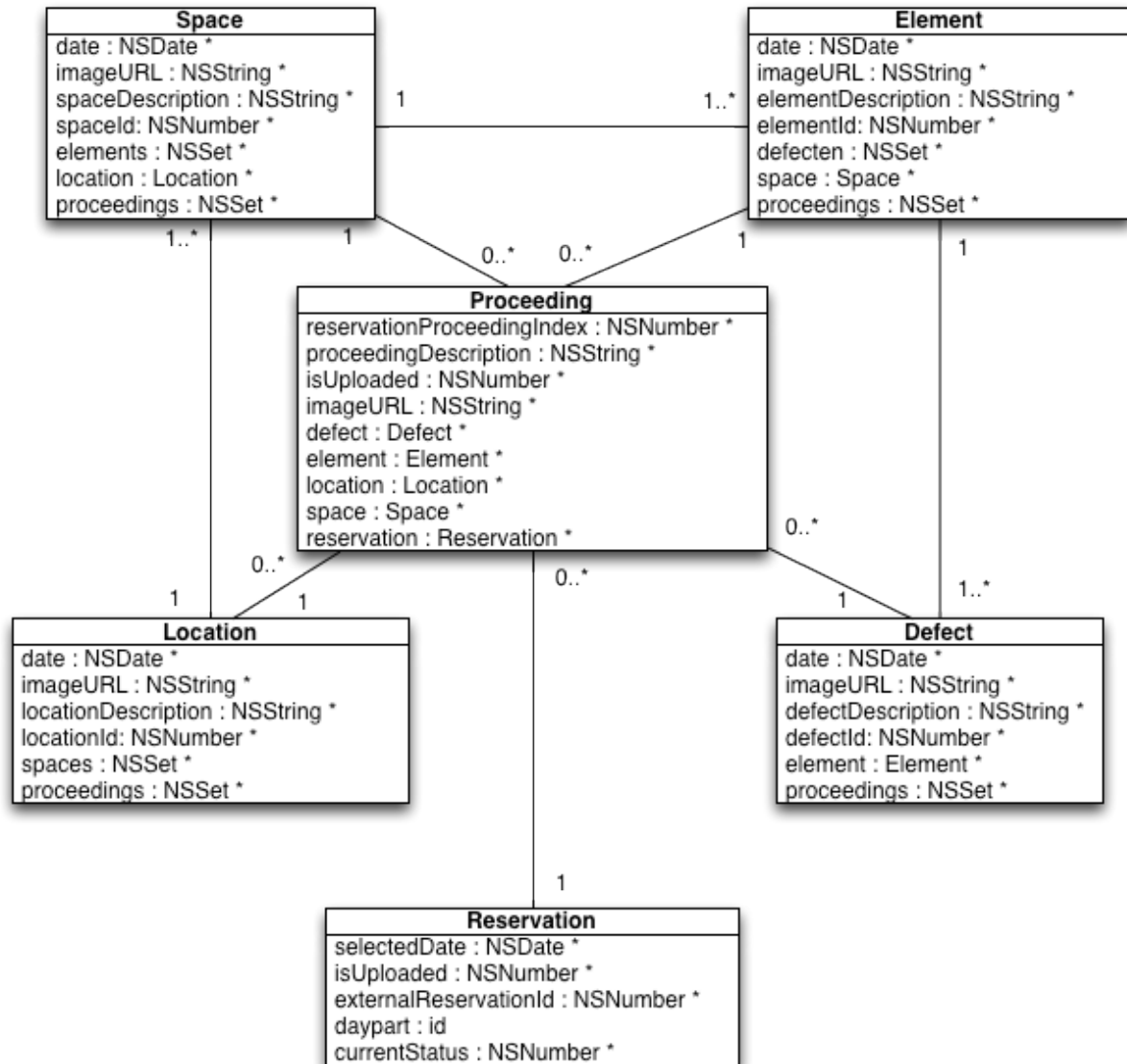
Bijlage N



Bijlage O



Bijlage P



Bijlage Q

API documentatie BouwCloud

Standaard

Voor de API methodes die gebruikt worden voor de BouwCloud mobiele applicatie zal gebruikt gemaakt worden van een JSON response. Alle content dat als antwoord verstuurd wordt zal in deze vorm zijn op afbeeldingen en andere binaire bestanden na. Voor een JSON response wordt de JSend¹ standaard aangehouden, deze standaard zorgt ervoor dat een standaard formaat van response wordt aangeleverd aan de client om makkelijk een foutmeldingen in te lezen of een correct response.

In een JSend response zal altijd het “status” veld gevuld zijn met een “**success**”, “**error**” of “**fail**”. Aan de hand van deze status weet je dat een bepaald verzoek succesvol is uitgevoerd of dat er een fout is opgetreden. Het data veld zal in de meeste gevallen gevuld zijn bij een “**success**” of “**fail**” status. In dit veld zal zich de dynamische resultaten van het verzoek bevinden. Wanneer de status “**error**” bevat zal een extra veld “**message**” gevuld zijn met informatie waarom de fout is opgetreden, daarnaast is het soms ook mogelijk dat een “**code**” en/of “**data**” veld in dit geval gevuld is.

Bij een geauthoriseerd verzoek wordt bedoelt dat inloggegevens moeten worden gebruikt die door middel van de “authenticationRequest” API methode zijn opgevraagd. Mocht deze info niet zijn meegegeven wordt een authenticatie foutmelding teruggestuurd voor verzoeken waar dit benodigd is. Voor elke API methode behalve de “authenticationRequest” en “getImage” methode is authenticatie vereist.

Spceifieke foutmeldingen zullen per methode beschreven worden, naast deze foutmeldingen kunnen ook exceptions optreden waarvan de foutmeldingen variabel zullen zijn. Deze meldingen zullen in het “**message**” veld worden ingevuld.

¹ <http://labs.omniti.com/labs/jsend>

POST */api/v1/addMaintenance/*

Maak een nieuwe reparatieverzoek aan met de verkregen gegevens, de informatie wordt gekoppeld aan het *user_id* wanneer de geauthenticeerde toegang heeft om informatie toe te voegen aan de gekoppelde user.

Parameter	Description
description <i>optioneel</i>	De beschrijving van het probleem, is optioneel en wordt gelimiteerd op 256 karakters, wordt automatisch verkort naar dit aantal karakters wanneer nodig.
defect_id <i>verplicht</i>	Het unieke nummer van de defect, dit is het uiteindelijke defect die de gebruiker heeft gekozen uit de locatie, ruimte en element.
userfile <i>optioneel</i>	Optioneel een afbeelding die kan worden toegevoegd voor een betere visuele beschrijving van het probleem.
user_id <i>verplicht</i>	Het gebruiker nummer waar het verzoek aan gekoppeld zal worden.
is_final_maintenance <i>optioneel</i> <i>values = 'true' / 'false' (String)</i>	Wanneer deze waarde 'true' is wordt aangegeven dat alle verzoeken voor de gebruiker zijn toegevoegd. De API zal er nu voor zorgen dat een email wordt verstuurd met alle informatie van het reparatieverzoek naar de gebruiker.

Data

Wanneer alle informatie succesvol is verwerkt zullen geen data velden worden teruggestuurd, alleen het status veld zal de waarde "success" bevatten waaraan te zien is dat het verzoek is toegevoegd.

Specifieke foutmeldingen

oa.maintenance.image.failed_upload

Wanneer deze foutmelding wordt teruggestuurd betekent dit dat er een fout is opgetreden tijdens het verwerken van de afbeelding.

oa.maintenance.validation.description.invalid

Deze foutmelding wordt teruggestuurd wanneer een foute beschrijving is toegevoegd, dit betekent dat de lengte van deze string langer is dan 256 karakters. Er wordt geen controle uitgevoerd op karakters, alleen een escape methode valideert deze informatie.

oa.maintenance.validation.missing_values

Wordt gegenereerd wanneer niet alle bovenstaande verplichte waarden zijn opgegeven.

POST /api/v1/addUser

Deze methode voegt een nieuwe gebruiker aan het systeem toe met de volgende parameters.

Parameter	Beschrijving
name <i>verplicht</i>	De volledige naam van de gebruiker, inclusief de achternaam en een eventueel tussenvoegsel. Deze waarde is verplicht om een gebruiker te kunnen toevoegen.
email <i>verplicht</i>	Het emailadres van de gebruiker, dit adres zal worden gevalideerd op de juiste vorm voordat het zal worden toegevoegd. Wanneer het adres niet wordt goedgekeurd zal een foutmelding worden teruggestuurd.
phone_number <i>verplicht</i>	Het verplichte telefoonnummer van de gebruiker, zal worden gevalideerd op een juist mobiel telefoonnummer of een juist nederlands net nummer. Kan een foutmelding genereren.
phone_number_optional <i>optioneel</i>	Het optionele telefoonnummer, wanneer deze waarde wordt meegegeven ook al is het leeg zal het op de manier worden gevalideerd als het verplichte telefoonnummer. Ook wanneer de telefoonnummers hetzelfde zijn zal een foutmelding worden gegenereerd.
zipcode <i>verplicht</i>	Een nederlands postcode in de vorm 1234AB of 1234 AB (niet hoofdlettergevoelig). Wanneer het adres niet bestaat of niet aan de voorwaarden voldoet zal een foutmelding worden teruggestuurd.
house_number <i>verplicht</i>	Een verplicht huisnummer, de postcode zal in combinatie met het huisnummer worden gevalideerd zodat het zeker weten bestaat.
number_addition <i>optioneel</i>	De optionele toevoeging aan het huisnummer, zal niet worden gevalideerd aangezien dit van alles kan zijn.
customer_id <i>verplicht</i>	Een koppeling met de customer (woningcorporatie) tabel. Wanneer het id niet bestaat zal een foutmelding worden meegestuurd.
date <i>verplicht</i>	Een UNIX timestamp in seconden na 1 januari 1970 die aangeeft op welke datum de gebruiker het reparatieverzoek wil plannen.
daypart_id <i>verplicht</i>	Het dagdeel identifier nummer dat aangeeft op welk dagdeel van bovenstaande datum de gebruiker het reparatieverzoek wil plannen.

Parameter	Beschrijving
notification_token <i>optioneel</i>	Een token van de telefoon om zo push notifications te kunnen verzenden, is niet verplicht.

Wanneer een van de bovenstaande waarden een foutmelding genereert zal de gebruiker niet worden toegevoegd.

Data

Als alle informatie succesvol is verwerkt en het reparatieverzoek is toegevoegd aan de database zal de volgende data teruggestuurd worden in het data veld.

Data Key	Opmerkingen
user_id	Bevat het nummer van het zojuist toegevoegde reparatieverzoek. Gebruik dit nummer om specifieke defecten aan het reparatieverzoek toe te voegen door middel van de addMaintenance API methode.

Specifieke foutmeldingen

oa.maintenance_user.insert.failed

Door een interne systeem fout kon de het nieuwe verzoek niet aangemaakt worden en weggeschreven worden in de database.

oa.maintenance_user.address.invalid

De opgegeven adres gegevens zijn niet geldig volgens de postcode database.

oa.maintenance_user.date.invalid

De datum waar het reparatieverzoek op gepland zal worden volgens de gebruiker is niet geldig of heeft al een maximaal aantal verzoeken gereserveerd.

oa.maintenance_user.missing.values

Geeft aan dat niet alle bovenstaande verplichte POST waarden zijn meegegeven.

oa.maintenance.user.validation.zipcode.invalid

Geeft aan dat de structuur van de postcode niet door de reguliere expressie controle is gekomen.

oa.maintenance.user.validation.housenumber.invalid

Geeft aan dat het opgegeven huisnummer geen correcte vorm heeft of groter is dan het maximale aantal 99999.

oa.maintenance.user.validation.email.invalid

Het opgegeven email adres kon niet door de reguliere expressie controle komen. Het formaat van dit email adres is dat niet geldig.

oa.maintenance.user.validation.name.invalid

Wanneer de volledige naam van de gebruiker niet wordt goedgekeurd wordt deze foutmeldingen gegenereerd.

oa.maintenance.user.phonenumber.invalid

Wanneer het verplichte telefoonnummer niet correct gevalideerd kon worden wordt deze waarde geretourneerd.

oa.maintenance.user.validation.phonenumber.optional.invalid

Wanneer een tweede telefoonnummer is opgegeven en de controle van dit telefoonnummer mislukt dan wordt deze foutmelding gegenereerd.

POST /api/v1/lookUpAddress/

Valideert de verstuurde postcode / huisnummer combinatie met de postcode database. Bij een normaal verzoek met juiste authenticatie gegevens worden geen specifieke foutmeldingen verwacht.

POST Parameter	Opmerkingen
zipcode <i>verplicht</i>	De postcode om te valideren. Kan geschreven worden met of zonder een spatie tussen de getallen en karakters.
house_number <i>verplicht</i>	Het huisnummer waar op gecontroleerd moet worden, moet een nummer zijn van maximaal 99999.
addition <i>optioneel</i>	Het optionele nummer toevoeging, zoals 'a' of 'boven'. Kan weggelaten worden.

Data

Als het adres succesvol is gevonden in de API database zal de volgende informatie worden teruggestuurd met specifieke informatie over het opgegeven adres.

Data Key	Opmerkingen
zipcode	De postcode die opgegeven is zonder spaties.
house_number	Het huisnummer zonder spaties of andere niet toegestane karakters.
house_number_addition	Het optionele toevoeging veld, dit veld wordt altijd teruggestuurd maar bevat niet altijd een toevoeging.
street	De straatnaam die hoort bij de opgezochte postcode.
city	De woonplaats die hoort bij de opgezochte postcode.
latitude	De geo latitude locatie van de postcode en huisnummer combinatie.
longitude	De geo longitude locatie van de postcode huisnummer combinatie.

Specifieke foutmeldingen

oa.address.validation.invalid_data

Wanneer de postcode, huisnummer of toevoeging niet door de reguliere expressie controle kon komen wordt deze foutmeldingen gegenereerd.

oa.address.validation.missing_values

Wanneer niet alle waardes benodigd voor de controle zijn meegestuurd wordt deze foutmeldingen teruggestuurd.

GET /api/v1/allCustomers/

Stuurt een lijst van woningcorporaties terug die te selecteren zijn in de mobiele applicatie. Voor deze API methode hoeven geen parameters meegestuurd te worden, wel moeten logingegevens worden meegegeven aangezien er gebruik gemaakt wordt van Basic HTTP authenticatie.

Data

In het data veld wordt een array van woningcorporaties teruggestuurd welke per item de volgende waardes bevat.

Data Key	Opmerkingen
id	Het unieke nummer van de woningcorporatie, gebruik dit nummer om een nieuw reparatieverzoek toe te voegen voor een woningcorporatie.
description	Dit veld bevat de naam van de woningcorporatie.

POST /api/v1/authenticationRequest/

Genereert een gebruikersnaam en wachtwoord die gebruikt kunnen worden voor de mobiele applicatie om alle andere API methodes aan te roepen.

POST Parameter	Opmerkingen
UUID <i>verplicht</i>	Een 36 karakter lange string die gebruikt wordt om een mobiel apparaat te herkennen. Een iPhone gebruikt een vooraf gegenereerd UUID om authenticatie gegevens van de API op te vragen.

Data

Wanneer de informatie succesvol verwerkt kon worden en nieuwe gegevens zijn aangemaakt worden de volgende keys gevuld in het "data" veld.

Data veld	
user	Bevat een string van 64 karakters die gebruikt kan worden als username voor HTTP authenticatie bij opvolgende verzoeken.
token	Bevat een string van 64 karakters die gebruikt kan worden als wachtwoord voor HTTP authenticatie bij opvolgende verzoeken.

Specifieke Foutmeldingen

om.uuid.validation.create.failed

Wordt teruggestuurd wanneer de API door een interne fout geen login gegevens kon genereren.

om.uuid.validation.invalid.uuid

Wanneer het meegegeven UUID niet voldoet aan de precieze 36 karakter lengte wordt deze waarde teruggestuurd.

GET /api/v1/getAvailableDates/

Stuurt een lijst van beschikbare datums en de daarbij behorende beschikbare dagdelen terug voor reparatieverzoeken. De gebruiker kan een van deze dagdelen op datum selecteren om het reparatieverzoek op te reserveren. De datum objecten die geretourneerd worden zullen starten op 3 dagen na de huidige server datum, waarna in totaal tot en met 31 dagen van die datum in de toekomst objecten worden geselecteerd. Al deze objecten hebben minimaal een vrije reservering plek.

Data

De volgende lijst zal in het datum veld worden ingevuld wanneer het verzoek succesvol is verwerkt. Deze objecten kunnen deze 1 of meerdere keren voorkomen.

Data key	Opmerkingen
date	Een UNIX timestamp vanaf 1 januari 1970 die de datum van een mogelijke planning representeert.

Onder elk “date” key bevindt zich een andere array met alle beschikbare dagdelen, de volgende tabel toont de waardes die in deze lijst gezet kunnen worden.

Data key	Opmerkingen
id	Het unieke nummer van het dagdeel. Gebruik dit nummer om het daadwerkelijke reparatieverzoek te versturen.
description	De beschrijving van het tijdsvak van dit dagdeel, dit is een string.
start	Een string met de starttijd van het tijdsvak.
end	Een string met de eindtijd van het tijdsvak.

Specifieke foutmeldingen

api.rest.available_dates.none

Wordt teruggestuurd wanneer een geen datum en dagdeel objecten gevonden kunnen worden die vrij zijn. Dat betekent dat er in de komende maand geen vrije datums zijn.

POST /api/v1/cancelReservation/

Deze methode maakt het mogelijk om een reparatieverzoek te annuleren. Wanneer de geauthenticeerde gebruiker toegang heeft om het reparatieverzoek te annuleren wordt de status aangepast en een email verzonden naar de gebruiker met een bevestiging van de annulering.

Er wordt verwacht dat de volgende POST waarde meegegeven is.

POST key	Opmerkingen
reservation_id	Het unieke nummer van het reparatieverzoek dat geannuleerd moet worden.

Data

Als het reparatieverzoek succesvol geannuleerd is en het email bericht is ook verzonden naar de gebruiker zal alleen het unieke nummer van het reparatieverzoek dat geannuleerd is teruggestuurd worden.

Data key	Opmerkingen
id	Het unieke nummer van het reparatieverzoek dat geannuleerd is.

Specifieke foutmeldingen

oa.maintenance.validation.access.denied

Als de gebruiker geen toegang heeft om het reparatieverzoek te annuleren wordt deze foutmelding teruggestuurd.

oa.reservation.cancel.invalid.id

Als door een onbekende fout het reparatieverzoek voor het meegegeven unieke nummer niet gevonden kon worden zal deze foutmelding worden teruggestuurd.

oa.reservation.cancel.missing.id

Als het unieke nummer van het reparatieverzoek niet is meegegeven aan het verzoek zal deze foutmelding meegegeven worden.

POST /api/v1/checkState/

Deze API methode retourneert de status van alle reparatieverzoeken die in een lijst van unieke nummers zijn verstuurd in het verzoek. Je kan deze status opvragen en bijvoorbeeld een nieuwe status “cancelled” terugkrijgen. In onderstaande tabel zie je de vereiste POST parameter die nodig is voor dit verzoek. Wanneer de geauthenticeerde gebruiker geen toegang heeft tot een bepaald reparatieverzoek wordt geen informatie van dat verzoek teruggestuurd.

POST key	Opmerkingen
reservation_list	Een komma gescheiden string met unieke nummers van de reparatieverzoeken waar de status van zal worden opgevraagd.

Data

Een lijst van volgende objecten zal teruggestuurd worden wanneer het verzoek succesvol uitgevoerd is.

Data key	Opmerkingen
id	Het unieke nummer van het reparatieverzoek waar de status toe behoort.
state	<p>Een van de volgende strings zal de status identificeren van het reparatieverzoek:</p> <ul style="list-style-type: none">• “reserved” -> Staat voor verstuurd maar nog niet bevestigd door BouwMeester.• “approved” -> Staat voor bevestigd door BouwMeester.• “denied” -> Staat voor afgewezen door BouwMeester en zal dus niet uitgevoerd worden.• “finished” -> Het reparatieverzoek is uitgevoerd.• “cancelled” -> Het reparatieverzoek is geannuleerd door de gebruiker.• “unknown” -> Wanneer door een interne fout geen status gevonden kon worden wordt deze status teruggegeven.

Specifieke foutmeldingen

oa.status.request.list.missing

Deze foutmelding betekent dat de komma gescheiden lijst van reparatieverzoek nummers ontbreekt.

POST /api/v1/getImage/

Deze API methode download een afbeelding van de server voor een bepaalde stap in het uitkiezen van een defect. Het kan zijn dat een afbeelding niet beschikbaar is voor een dergelijk object waardoor een JSON string wordt teruggestuurd. De volgende variabelen worden in de POST verwacht.

POST key	Opmerkingen
type	Het type afbeelding dat gedownload moet worden. Dit moet een van de volgende string waarden zijn: <ul style="list-style-type: none">• “Location” -> Een locatie object afbeelding• “Space” -> Een ruimte object afbeelding.• “Element” -> Een element object afbeelding.• “Defect” -> Een defect object afbeelding.
id	Het unieke nummer van bovenstaande type object, van de combinatie van type & id zal de afbeelding worden opgezocht en teruggestuurd worden.

Data

Door naar de MIME type van het verzoek te kijken dat teruggekregen is kan worden bepaald of een afbeelding is gedownload of een JSON string. Een JSON string zal altijd een van de onderstaande foutmeldingen bevatten. Daarnaast zal een afbeelding MIME type de bytes van de afbeelding in het verzoek hebben staan.

Specifieke foutmeldingen

oa.image.request.unavailable

Wordt teruggestuurd wanneer geen afbeelding gevonden kon worden bij de combinatie van POST parameters.

oa.image.request.missing.parameters

Wanneer niet alle benodigde POST parameters zijn verstuurd zal deze foutmelding teruggestuurd worden.

GET /api/v1/synchronizeDefects/

Wanneer deze API methode opgevraagd wordt zal een complete lijst van Locaties, Ruimtes, Elementen en Defecten worden teruggestuurd die door de gebruiker geselecteerd kunnen worden voor een reparatieverzoek. De gebruiker moet geauthenticeerd zijn voordat deze informatie gedownload kan worden.

Data

De volgende vier lijsten zullen in de juiste volgorde zoals die hier staat teruggestuurd worden.

“locations” lijst	Opmerkingen
id	Het unieke nummer van het locatie object.
description	De beschrijving van het locatie object, dit is een string van maximaal 64 karakters.

“spaces” lijst	Opmerkingen
id	Het unieke nummer van het ruimte object.
description	De beschrijving van het ruimte object, dit is een string van maximaal 64 karakters.
om_location_id	Het unieke nummer van het locatie object waar deze ruimte toe behoort.

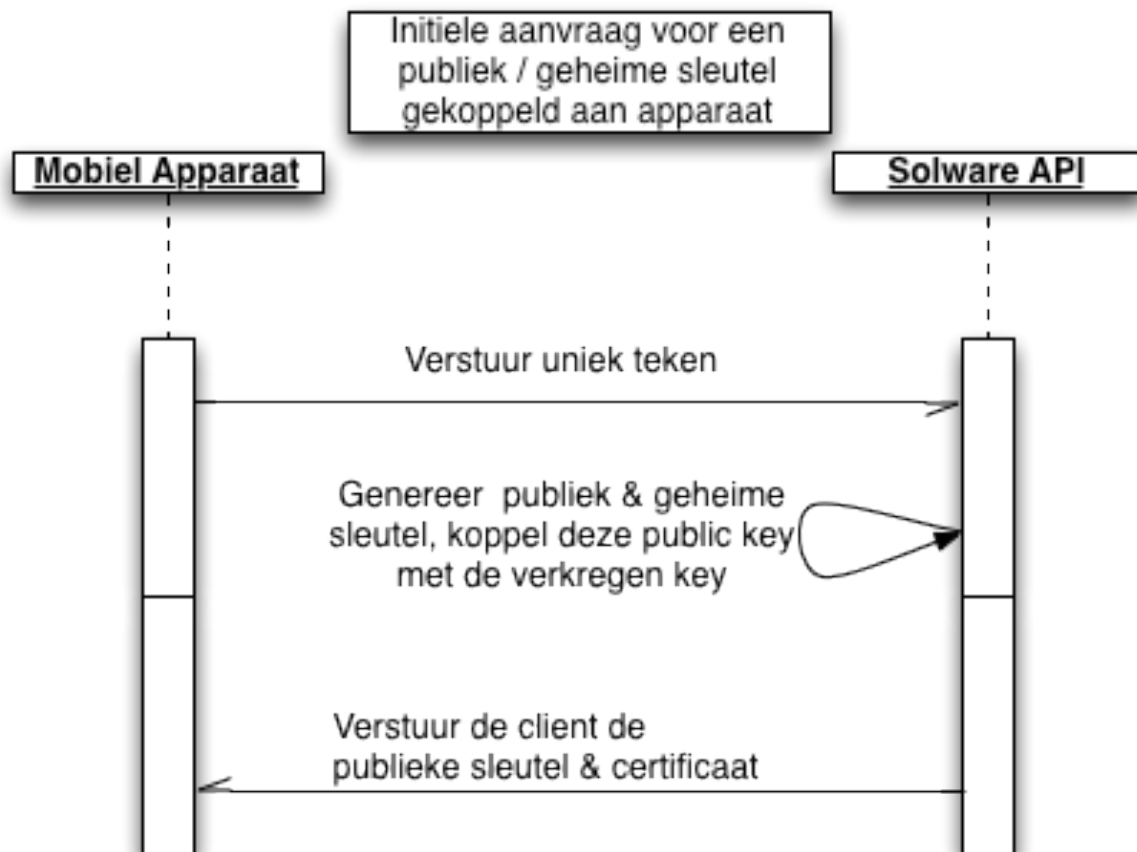
“elements” lijst	Opmerkingen
id	Het unieke nummer van het element object.
description	De beschrijving van het element object, dit is een string van maximaal 64 karakters.
om_space_id	Het unieke nummer van het ruimte object waar dit element toe behoort.

“defects” lijst	Opmerkingen
id	Het unieke nummer van het defect object.
description	De beschrijving van het defect object, dit is een string van maximaal 64 karakters.
om_element_id	Het unieke nummer van het element object waar dit defect toe behoort.

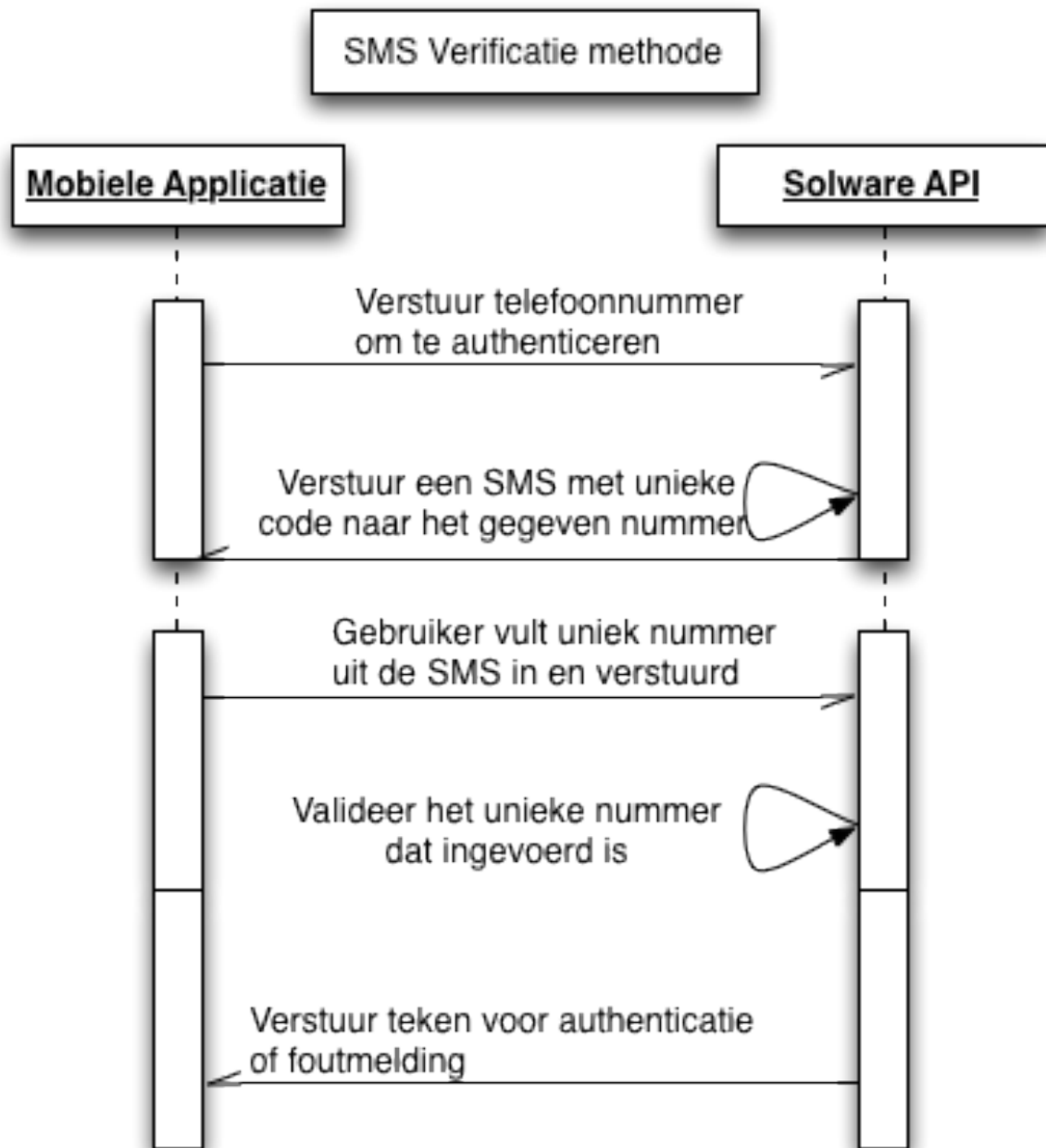
Specifieke foutmeldingen

Voor deze API methode zijn geen specifieke foutmeldingen gespecificeerd.

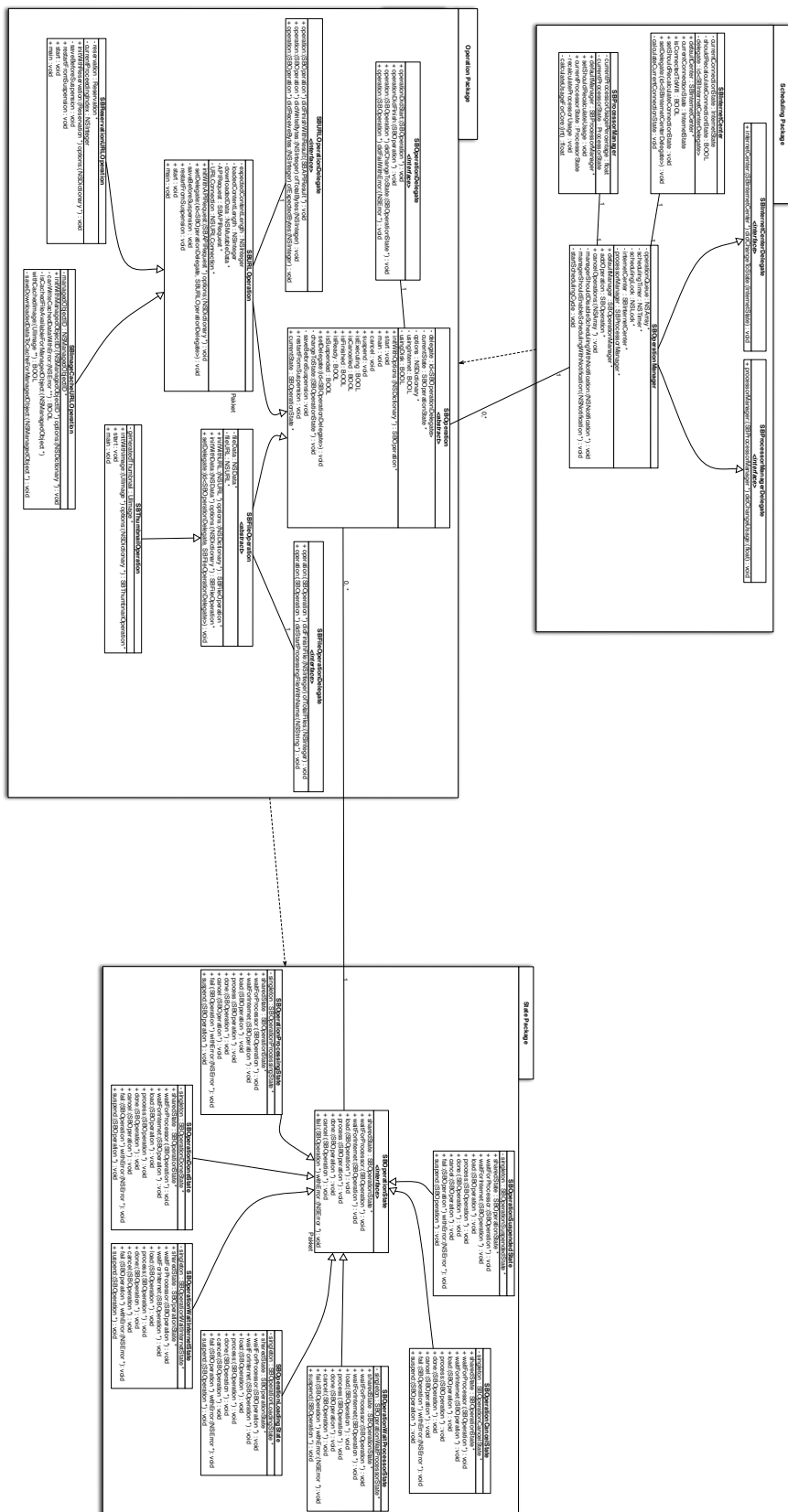
Bijlage R



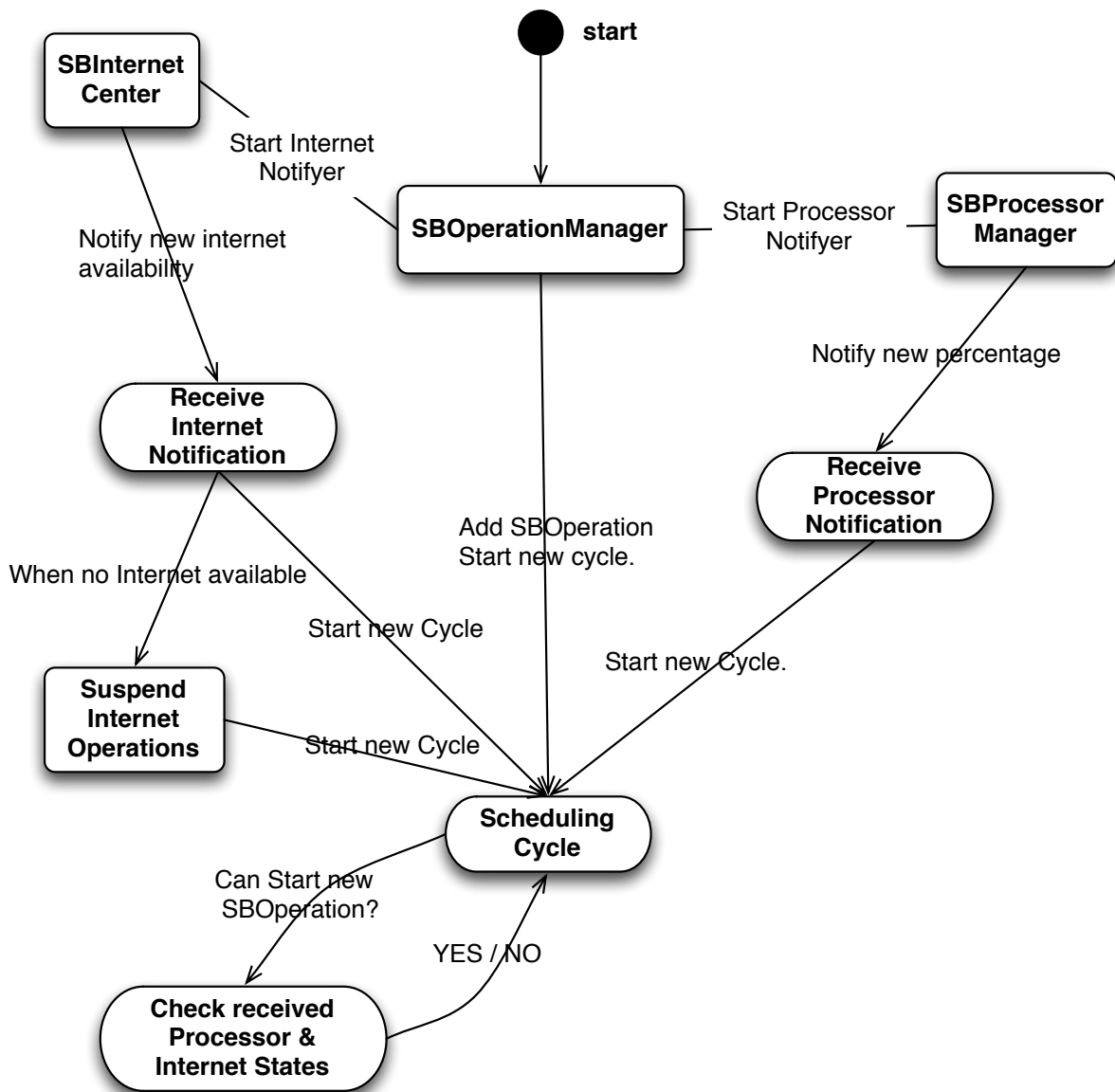
Bijlage S



Bijlage T



Bijlage U



Bijlage

In deze bijlage is alle informatie te vinden die te maken hebben met het onderzoek naar een snellere en meer efficiënte methode om internet verzoeken te versturen. De geschreven code die dit moet waarmaken is getest onder 4 omstandigheden en wordt gedraaid op een laptop of de daadwerkelijke smartphone waar de applicatie uiteindelijk op zal draaien.

- **WiFi** in combinatie met de ontwikkel machine
 - 1,7GHz Intel Core i5 (4 cores, 8 threads)
 - 4GB 1333MHz DDR3 geheugen
 - Opslag: Solid State Drive
- **WiFi** (gesimuleerd) in combinatie met iPhone 5
 - 1,3GHz Apple A6 (2 cores, 2 threads)
 - 1GB geheugen
 - Opslag: Flash
- **3G** (gesimuleerd) in combinatie met iPhone 5
- **Edge / GPRS** (gesimuleerd) in combinatie met iPhone 5

Per bovenstaand genoemd onderdeel zullen tests worden uitgevoerd die worden uitgevoerd op het nieuw geschreven scheduling systeem en zowel op het normale systeem die standaard wordt gebruikt. Het verzoek dat verstuurd wordt is altijd identiek en zal in totaal 63 bytes per verzoek binnenhalen. Voor dit aantal bytes is niet specifiek

Simulatie

gekozen. De verbinding is in het geval van de iPhone smartphone gesimuleerd aangezien er geen garantie kan worden gegeven dat de telefoon een bepaalde connectie blijft behouden. Dit is bereikt door een ingebouwde ontwikkelaar optie die het aantal binnen komende en uitgaande pakketjes limiteert zodat een bepaalde connectie gesimuleerd kan worden. Op deze manier kan betrouwbaar getest worden.

Gemiddelde berekening

Als laatste informatie zal per getest onderdeel de hoogste en laagste waarden niet meegerekend worden in het gemiddelde om op deze manier te verzekeren dat uitschieters geen invloed hebben op dit gemiddelde. Dit is namelijk een belangrijke keuze omdat het internet en de snelheid hiervan vaak even kan uitschieten waardoor de testresultaten ver uit elkaar kunnen liggen. Op deze manier wordt getracht dit te verminderen.

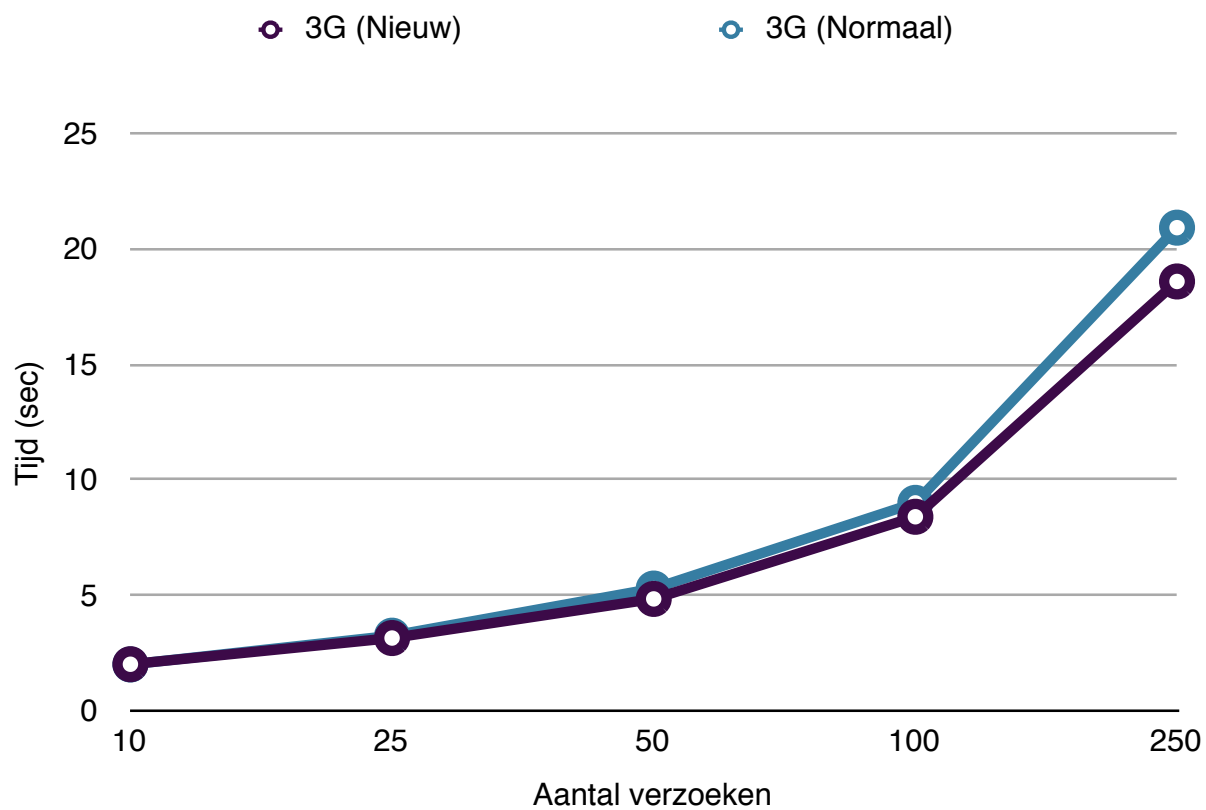
3G

#	10	25	50	100	250
1	2,263	3,228	4,873	8,117	18,663
2	1,987	3,165	5,359	8,220	18,454
3	2,012	3,095	4,962	9,293	18,799
4	1,967	3,113	4,833	8,444	19,061
5	2,007	3,327	4,838	8,243	18,502
6	2,008	3,382	4,890	8,042	18,005
7	2,009	3,090	4,827	8,197	17,999
8	2,126	3,139	4,640	8,745	18,489
9	1,928	3,006	4,798	8,152	18,755
10	2,046	3,072	4,763	9,116	20,340
Gemiddelde	2,020	3,154	4,848	8,404	18,591

Tabel 1: 3G omstandigheden met nieuwe scheduling.

#	10	25	50	100	250
1	2,497	2,954	5,263	8,841	20,998
2	2,177	3,121	5,251	8,590	19,912
3	1,959	3,516	5,406	8,483	21,433
4	1,91	3,265	4,903	8,931	20,363
5	1,918	3,294	5,090	9,255	20,963
6	1,962	3,286	5,352	8,914	19,735
7	1,962	3,214	5,216	9,006	21,444
8	1,917	3,345	5,51	9,251	22,294
9	2,091	3,298	5,509	9,295	21,193
10	2,093	3,296	5,280	10,109	21,045
Gemiddelde	2,01	3,256	5,296	9,006	20,92

Tabel 2: 3G omstandigheden net normale scheduling.



Grafiek 1: 3G performance resultaten tegenover elkaar. (Gemiddelde)

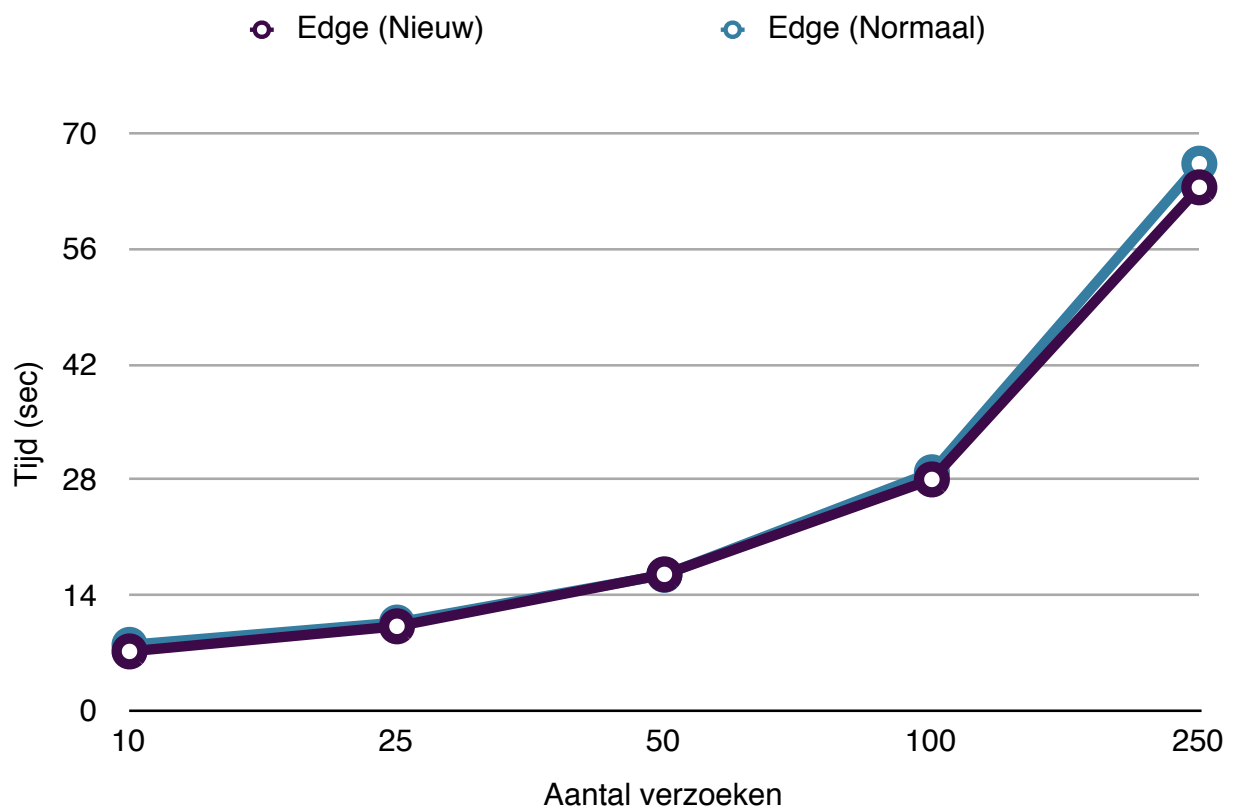
Edge / GPRS

#	10	25	50	100	250
1	7,254	10,231	16,593	28,841	63,157
2	7,182	10,250	16,704	27,08	63,344
3	7,155	10,232	16,485	27,821	64,466
4	7,170	10,229	16,554	28,019	63,378
5	7,213	10,259	16,370	28,122	63,070
6	7,206	11,129	16,603	27,899	63,320
7	7,203	10,173	17,691	28,740	64,216
8	8,109	10,301	16,504	27,728	63,926
9	7,205	10,352	16,503	28,064	63,346
10	7,266	10,238	16,468	28,056	63,063
Gemiddelde	7,212	10,255	16,552	28,061	63,470

Tabel 3: Edge omstandigheden met vernieuwde scheduling.

#	10	25	50	100	250
1	8,163	10,580	16,452	30,718	66,268
2	7,986	11,397	16,444	28,558	66,522
3	8,006	10,570	17,378	29,238	66,234
4	8,044	10,509	16,414	28,665	67,272
5	8,040	10,557	16,513	28,423	66,100
6	7,935	13,321	16,465	29,024	66,009
7	7,933	10,720	16,533	28,386	66,202
8	7,990	10,661	16,469	30,313	67,265
9	8,020	10,547	16,673	28,671	65,988
10	8,023	10,639	16,420	28,551	66,090
Gemiddelde	8,006	10,709	16,496	28,930	66,336

Tabel 4: Edge omstandigheden met normale scheduling.



Grafiek 2: Edge performance resultaten tegenover elkaar (gemiddelde).

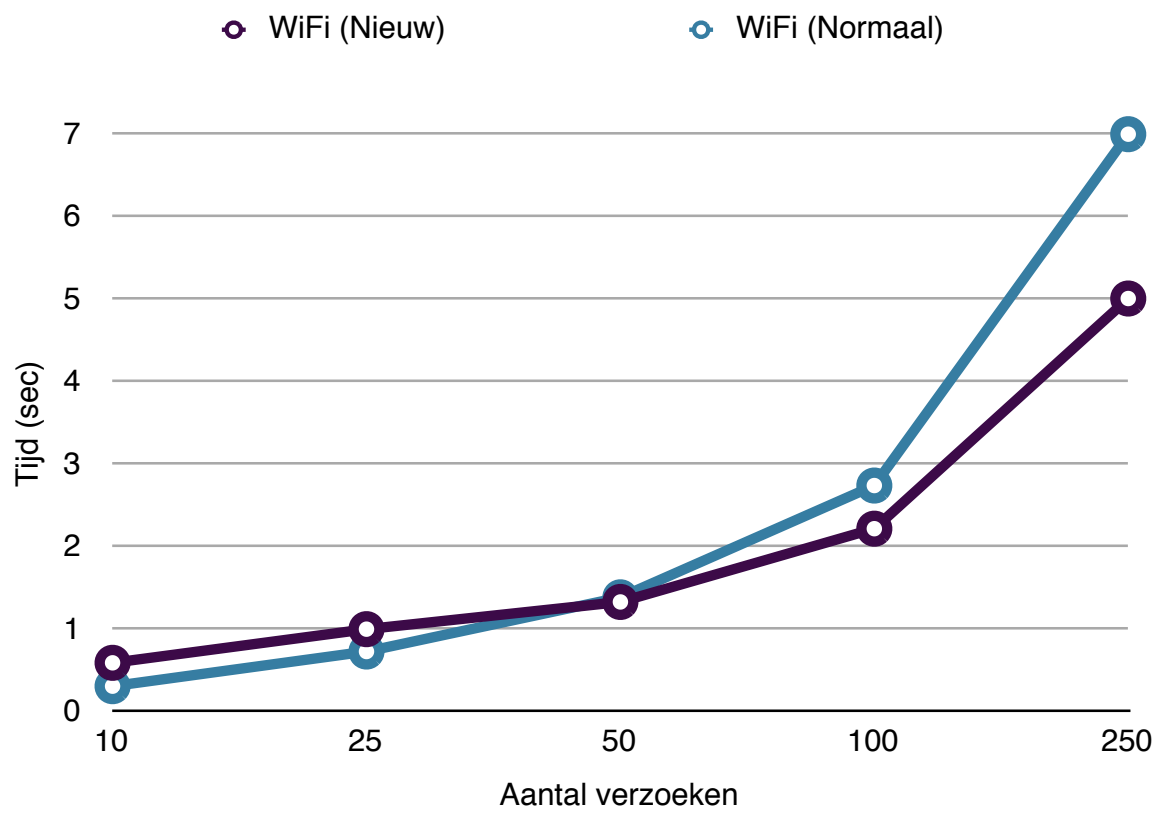
WiFi

#	10	25	50	100	250
1	0,581	1,608	1,311	2,337	5,131
2	0,580	1,009	1,286	2,212	4,952
3	0,577	0,967	1,233	2,195	4,988
4	0,595	1,067	1,348	2,221	4,924
5	0,623	1,031	1,379	2,171	5,070
6	0,595	1,098	1,256	2,182	4,883
7	0,600	0,973	1,350	2,144	4,913
8	0,52	0,939	1,297	2,230	6,037
9	0,510	0,804	1,339	2,252	4,946
10	0,661	0,840	1,419	2,208	5,064
Gemiddelde	0,583	0,991	1,321	2,209	4,999

Tabel 5: Wifi omstandigheden met vernieuwde scheduling.

#	10	25	50	100	250
1	0,305	0,694	1,352	2,745	6,887
2	0,305	0,743	1,361	2,94	6,761
3	0,300	0,760	1,390	2,733	7,056
4	0,278	0,706	1,361	2,719	7,615
5	0,302	0,721	1,371	2,718	6,841
6	0,293	0,723	1,386	2,701	6,831
7	0,297	0,677	1,356	2,662	7,668
8	0,288	0,728	1,419	2,704	6,909
9	0,310	0,687	1,375	2,803	6,839
10	0,316	0,767	1,413	2,662	6,947
Gemiddelde	0,3	0,720	1,377	2,732	6,991

Tabel 6: Wifi omstandigheden met normale scheduling.



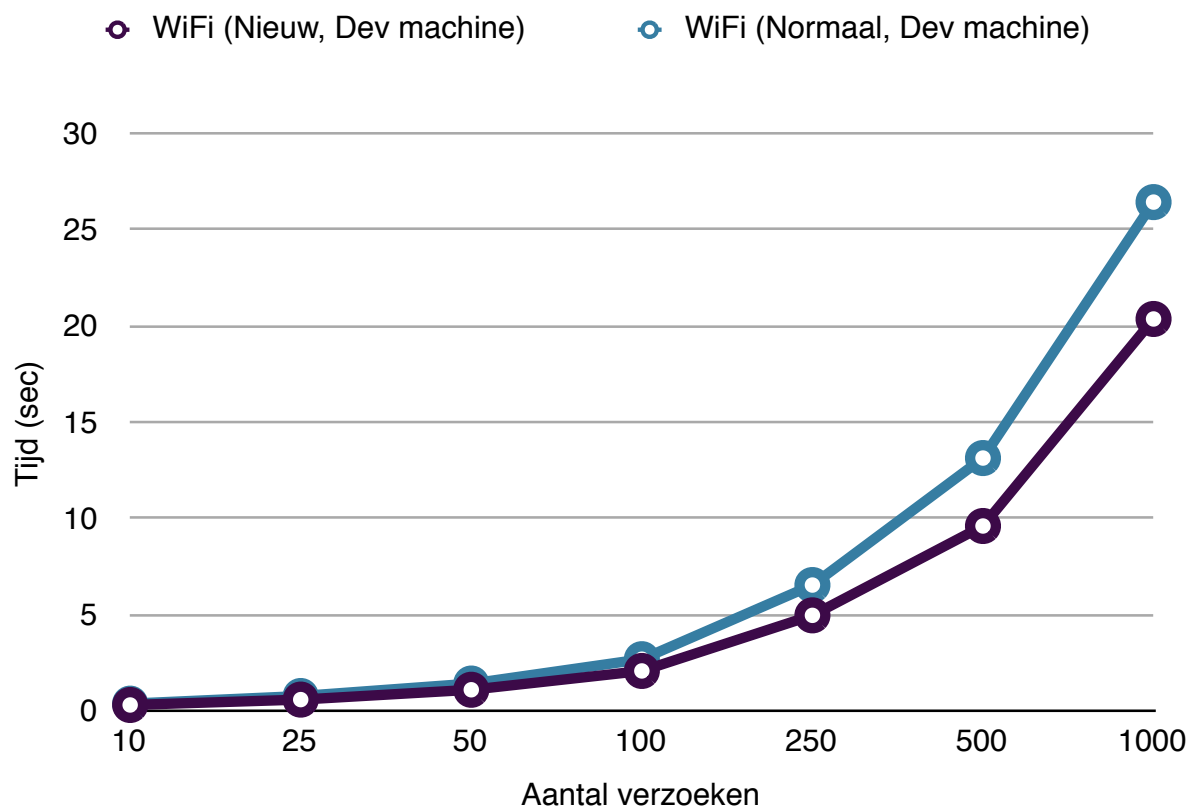
WiFi (Laptop)

#	10	25	50	100	250	500	1000
1	0,306	0,650	1,035	1,973	4,723	12,985	19,413
2	0,308	0,593	1,029	2,194	4,912	9,390	19,726
3	0,291	0,566	1,082	1,938	6,144	9,380	19,381
4	0,310	1,225	1,056	2,047	4,857	9,560	18,968
5	0,289	0,602	1,038	2,100	4,784	10,177	19,245
6	0,365	0,562	1,054	2,207	4,903	9,377	26,408
7	0,381	0,561	1,394	2,716	5,566	9,531	19,774
8	0,283	0,569	1,372	2,002	4,895	9,848	26,91
9	0,291	0,570	1,084	2,029	4,850	9,509	19,739
10	0,283	0,589	1,072	2,064	4,952	9,464	19,208
Gemiddelde	0,309	0,588	1,099	2,077	4,965	9,607	20,362

Tabel 7: WiFi resultaten op een developer machine (Nieuw).

#	10	25	50	100	250	500	1000
1	0,402	0,745	1,436	2,698	6,554	13,204	27,002
2	0,366	0,857	1,421	2,755	6,472	13,307	27,513
3	0,372	0,761	1,389	2,595	6,519	13,349	25,919
4	0,367	0,770	1,386	2,637	6,459	12,924	26,361
5	0,394	0,769	1,401	2,646	6,687	12,739	26,248
6	0,381	0,763	1,420	2,629	6,385	13,585	26,692
7	0,364	0,773	1,623	2,659	6,642	13,451	26,477
8	0,376	0,715	1,463	2,789	6,509	13,106	26,662
9	0,469	0,781	1,438	2,766	6,469	9,49	26,109
10	0,367	0,835	1,408	2,709	6,718	13,013	25,687
Gemiddelde	0,378	0,775	1,422	2,687	6,539	13,137	26,434

Tabel 8: WiFi resultaten op een developer machine (Normaal).



Unit Tests

In dit document worden alle uitgevoerde unit tests beschreven die ervoor zorgen dat de mobiele applicatie of API voldoet aan functionele eisen en geen onnodige en gewilde fouten genereert.

Mobile Application

Om functionaliteit van de mobiele applicatie te garanderen is ervoor gekozen om door middel van Unit Tests te valideren of elk onderdeel van deze applicatie naar behoren werkt. Zo is bijvoorbeeld belangrijk om te controleren of de applicatie toegang heeft tot alle verschillende API methodes. Daarnaast moet ook gegarandeerd worden dat mogelijke gebruiker gegevens goed gekeurd worden. Terwijl foutieve email adressen etcetera gefilterd moeten worden en foutmeldingen genereren. Alle verschillende test cases die uitgevoerd worden op de mobiele applicatie zijn hier opgesomd.

- Validatie van gebruikers gegevens.
- Testen van elke API call of het verwachte resultaat wordt teruggestuurd.
- Token generatie en middel van API authenticatie moet altijd goed gaan.
- Testen van afbeelding verwerken en het juiste thumbnail formaat.

In de volgende afbeeldingen zijn de testresultaten te vinden per onderdeel waar een test case voor geschreven is.

Case	Fouten	Opmerkingen
Volledige naam	0	-
E-mail	2	- Het .web domein wordt niet geaccepteerd. - Een ongeldig IP adres wordt soms goedgekeurd.
Telefoonnummer	0	-
Adres	0	-
Base64 conversie	0	- Ongeldige authenticatie ligt niet aan de base64 conversie.
Afbeeldingen	2	- Memory management fout die ervoor zorgde dat uitgevoerde operaties niet werden opgeruimd. - Teveel tegelijk operaties zorgt ervoor dat veel bestanden niet worden verwerkt.(IO overload?).
API beschikbaarheid	0	- Wordt vaak uitgevoerd en controleert beschikbaarheid van API. Status code 200 wordt verwacht als response.
URL operaties	1	- 1 op de 500 operaties heeft de kans om een foutmelding te genereren.

Naam validatie

Een naam wordt als geldig gezien als wordt voldaan aan de volgende eisen; de naam bevat geen numerieke karakters, de volledige naam is meer dan of gelijk aan twee karakters en deze naam mag niet meer zijn dan 150 karakters. Namen met een numeriek karakter worden niet verwacht terwijl alle andere tekens die gebruikt kunnen worden wel worden goedgekeurd. Hier is voor gekozen omdat vooral buitenlandse namen vaak gebruik maken van tekens anders dan het alfabet. Door hier rekening mee te houden zorgen we ervoor dat niet onnodig foutmeldingen worden gegenereerd terwijl de naam eigenlijk wel correct is. De methode die de volledige naam test is [testFullNameValidation](#).

Tijdens het testen van de volledige naam zijn geen eigenaardigheden opgetreden wat zou betekenen dat de validatie van de naam werkt naar behoren.

E-mail validatie

Een e-mail adres moet voldoen aan de specificatie zoals dat is beschreven in de RFC-2822. In dit bestand staan alle voorwaarden waar een email adres aan moet voldoen. In de code voorbeelden ziet u welke tests zijn uitgevoerd om ervoor te zorgen dat de email validatie voldoet aan dit formaat. Zo mag een gebruikersnaam gedeelte van het adres een '+' symbool bevatten en moet een domeinnaam dat als IP adres ingevuld is de juiste vorm van een IP adres bevatten. De methode die dit afhandelt is de methode [testEmailValidation](#).

Tijdens het testen werd de conclusie getrokken dat twee tests niet zoals verwachten uitkwamen. Dat wil zeggen dat een aanneming fout is gegaan, bijvoorbeeld de test case dat een .web niet goed gekeurd mag worden is als resultaat dat het wel goedgekeurd wordt. In totaal zijn twee testcases uitgevoerd met het verkeerde resultaat van de totaal 31 cases die voor email validatie zijn uitgevoerd.

Uiteindelijk is gekozen om deze twee test cases niet te verbeteren, om dit voor elkaar te krijgen zou dat betekenen dat een ingewikkelde reguliere expressie aangepast moet worden. Deze expressie is gemaakt voor de RFC-2822 versie en heeft een aantal type adressen die niet afgevangen worden. Deze fouten zullen in de code worden gelaten om er zo voor te zorgen dat niet andere fouten worden geactiveerd wanneer de reguliere expressie wordt aangepast. Ook de doelgroep die de applicatie zal gebruiken heeft geen kennis van IP adressen als domeinnamen of het .web domein extensie.

Telefoonnummer validatie

Om een telefoonnummer te valideren wordt ook gebruik gemaakt van een reguliere expressie. Deze expressie controleert of het om een Nederlands mobiel nummer of prive nummer gaat, daarnaast worden een aantal nummers die door bedrijven gebruikt worden ook geaccepteerd. Om deze validatie te testen zijn 14 verschillende foute en goede telefoonnummers getest om te kijken of het juiste resultaat wordt teruggestuurd.

Deze test cases zijn in de code terug te vinden in de methode [testPhoneNumberValidation](#). In deze methode bevinden zich alle testcases die ervoor zorgen dat een juiste telefoon validatie wordt uitgevoerd. Uiteindelijk bleek een fout in de reguliere expressie ervoor te zorgen dat sommige telefoonnummers die achter elkaar numerieke karakters bevat en ver de juiste vorm heeft worden afgekeurd. De oplossing die

ervoor zorgde dat deze bug werd opgelost is het verwijderen en verplaatsen van een combinatie '(' & ') ' karakters in de reguliere expressie die de telefoonnummers controleert.

Adresgegevens validatie

Het laatste onderdeel van gebruiker gegevens validatie is de postcode en huisnummer combinatie. Een postcode moet uit 4 nummers bestaand gevolgd door twee alfabetische karakters. Eventueel kan tussen deze twee onderdelen een spatie worden opgegeven. Een huisnummer moet in eerste instantie uit nummers bestaan, met een maximaal nummer van 99999. Na enig onderzoek blijkt dat er namelijk geen Nederlandse huisnummers bestaan hoger dan dit getal. In het huisnummer wordt automatisch het toevoeging gedeelte zoals 'a' of 'boven' gefilterd zodat alleen het numerieke gedeelte wordt gevalideerd.

In de methode [*testZipcodeHouseNumberValidation*](#): wordt de combinatie van beide onderdelen gecontroleerd op juistheid. Wanneer beide onderdelen correct gevalideerd worden is de combinatie goedgekeurd. In totaal zijn 15 test cases uitgevoerd waar geen onverwachte resultaten uit zijn opgetreden. Hierover kunnen we dus concluderen dat het valideren van deze postcode naar behoren werkt in de tests is te zien dat geen rare combinaties een foute waarde opleveren.

Base64 validatie

Tijdens het uitvoeren van de applicatie in de simulator en test telefoon is het wel eens voorgekomen dat van een string geen base 64 representatie object van kon worden aangemaakt. Deze functie moet veel getest worden aangezien dit een belangrijk onderdeel is voor de communicatie met de API, het wordt namelijk bij de authenticatie gebruikt om een gebruiker te herkennen. Om dit te testen is een case opgezet die 100000 keer het aanmaken van een base64 representatie aanmaakt, wanneer de waarde 'null' wordt teruggestuurd mislukt de test. Wanneer de test gelukt is kunnen we aannemen dat het aanmaken van deze representaties niet het probleem is waardoor soms de authenticatie mislukt met de API.

Na de tests kunnen we concluderen dat het probleem dat validatie met de API mislukt niet ligt aan de base64 conversie methode. Na herhaaldelijk testen is namelijk geen enkele keer voorgekomen dat deze conversie is mislukt. Er zal op een andere plek moeten worden onderzocht waarom de API authenticatie mislukt. Voor referentie naar de methode die gebruikt wordt om de test uit te voeren is de methode naam [*testBase64Conversion*](#).

API beschikbaarheid

Een belangrijke test die vaak wordt uitgevoerd om te controleren of alle API methodes bereikbaar zijn voor de mobiele applicatie wordt de methode [*testAPIAvailability*](#). Deze test roept alle mogelijke methodes op de API aan en controleert of de status code 200 wordt teruggestuurd, wanneer dit niet gebeurt is het mogelijk dat de methode naam fout is opgegeven wat ervoor kan zorgen dat de mobiele applicatie niet naar behoren werkt. Er wordt geen POST informatie verstuurd wat kan betekenen dat überhaupt niet het juiste resultaat wordt teruggestuurd, dit is niet vereist aangezien we in deze test alleen willen weten of de API bereikbaar is.

Een belangrijk punt waar bij deze test rekening mee moet worden gehouden is het feit dat er een authenticatie base64 string aanwezig moet zijn voordat de test gestart kan worden.

Anders zal elke methode behalve de authenticatie aanvraag methode mislukken wat geen juist resultaat is. Deze test geeft op moment van schrijven geen foutmeldingen, deze methode zal vaker gebruikt worden om te kunnen controleren op een snelle niet handmatige wijze of de API beschikbaar is.

Afbeelding verkleining

Afbeelding verkleining wordt gebruikt om thumbnails te maken van geselecteerde of nieuwe foto's die aan een reparatieverzoek toegevoegd worden. Verwacht wordt dat een afbeelding die verkleind of 800 pixels hoog of 800 pixels breed is. Er is namelijk opgegeven dat een dergelijke afbeelding tot maximaal dat formaat wordt verkleind, dit is ook een mooi formaat om op het scherm te tonen. De test hiervoor wordt uitgevoerd met een tiental aantal afbeelding van rond de 5 - 8 MegaByte, deze afbeelding worden verkleind en de resulterende afbeelding wordt gevalideerd op grootte. Tegelijkertijd wordt ook gecontroleerd of het verwerken van meerdere bestandsoperaties juist werkt. De methode die deze test uitvoert is de [testImageConversion](#) methode.

Tijdens het uitvoeren van deze test kwam een bug aan het licht die een groot probleem had kunnen veroorzaken wanneer de applicatie uitgegeven zou zijn. Bestandsoperaties worden namelijk niet goed in het geheugen beheert waardoor de thread niet wordt opgeruimd, dat betekent op zijn beurt dat de queue volloopt en geen andere threads wil inplannen. Op deze manier kunnen geen nieuwe achtergrondtaken worden uitgevoerd waardoor de mobiele applicatie nutteloos wordt. Door deze geheugen problemen op te lossen is de mobiele applicatie stabiel gemaakt.

Een volgend probleem dat optrad was het feit dat de bestandsoperaties fouten genereerden wanneer er meer dan 5 tegelijk worden uitgevoerd. Dit komt waarschijnlijk doordat teveel schijf activiteit tegelijk wordt uitgevoerd op meerdere bestanden waardoor de IO controller vast kan lopen. Dit is echter een aanname en kan ook een andere reden hebben. Om dit op te lossen moet echter een limiet aan het aantal bestandsoperaties worden gezet dat tegelijk mag worden uitgevoerd. Voor testen is dit maximale aantal op 2 gezet dat ervoor zorgde dat de operaties veel sneller en zonder fouten uit werden gevoerd. In de uiteindelijke versie moet een uitbreiding worden toegevoegd die dit probleem kan omzeilen.

URL operaties

De laatste test is het tegelijk uitvoeren van verschillende URL operaties, dat wil zeggen communiceren met de API. In totaal wordt tegelijk 100 connecties in de queue gezet om uitgevoerd te worden. Het ideale resultaat hiervan is dat geen enkele connectie een fout oplevert, dat wil zeggen een ander resultaat dan code 200. Deze test is al vaak uitgevoerd tijdens het onderzoek naar scheduling en is daardoor al betrouwbaar. De methode die de test uitvoert is [testConcurrentURLOperations](#).

Het resultaat van deze test is dat 80% van de keren dat de test wordt uitgevoerd geen enkele connectie een fout oplevert. In 20% van de keren dat het wordt uitgevoerd wil het voorkomen dat een enkele connectie een fout oplevert, in totaal kan je zeggen dat 1 op de 500 connecties fout wordt uitgevoerd. De foutmelding wordt wel netjes aangemaakt maar onvermijdelijk is het niet omdat internet altijd in snelheid anders kan zijn en om andere redenen en in andere gevallen onbetrouwbaar zal zijn.

API

De API wordt getest door een aantal mogelijke API request na te bootsen. Een test wordt dan goedgekeurd door een aanneming te doen op de teruggestuurde data, zo moet een bepaald verzoek de status “error” terugsturen en een ander verzoek de status “success”. Wanneer deze waarden anders zijn dan verwacht wordt een test niet succesvol uitgevoerd. Het is belangrijk dat al deze test goed worden uitgevoerd aangezien dit het aanspreekpunt is voor de mobiele applicatie.

Het testen zal gebeuren door middel van php en de cURL extensie. Met een cURL object kunnen we een connectie aanmaken die in een enkele regel het resultaat hiervan zal terugsturen. Deze informatie wordt gevalideerd op de manier zoals hierboven uitgelegd is. Foute cURL objecten worden in het test resultaat gelogd.

Bugs

Tijdens het testen zijn een aantal bugs ontdekt die gelijk opgelost zijn. Een voorbeeld hiervan is een foute authenticatie controle die ervoor zorgt dat het resultaat altijd tegenovergesteld was, door dit om te draaien kunnen we garanderen dat de authenticatie van gebruikers vanaf nu wel goed zal worden afgehandeld.

Een ander voorbeeld is de bug die ervoor zorgt dat juiste adresgegevens verkeerd werden verwerkt, dit gebeurde wanneer een postcode een spatie als tussen karakter bevat. De API waar de geschreven API mee praat verwerkt dit namelijk als een verkeerd adres. Hierdoor kan het gebeuren dat een fout resultaat wordt teruggestuurd. Een laatste bug was dat wanneer een woningcorporatie een ander id heeft dan in de applicatie gekozen kan worden (id aanpassen in request) een verkeerde woningcorporatie kan worden uitgekozen die niet beschikbaar is voor de mobiele applicatie. Bij het ophalen van deze gegevens is de controle hiervoor al geschreven, echter is deze controle nu ook bij het verwerken van een nieuw verzoek geïmplementeerd om er zo voor te zorgen dat het id niet voor verkeerde data zorgt.

Test code

De code die het testen van bovenstaande API methodes afhandelt kan worden teruggevonden in de bijlagen.

Afstudeerplan

Informatie afstudeerder en gastbedrijf

Afstudeerblok: 2013-1.2 (start uiterlijk 6 mei 2013)
Startdatum uitvoering afstudeeropdracht: 6 mei 2013
Inleverdatum afstudeerdossier volgens jaarrooster: 23 september 2013

Studentnummer:
Achternaam: dhr
Voorletters: S.H.
Roepnaam: Stephan
Adres: Jan Wapstraat 45
Postcode: 2523GB
Woonplaats: Den Haag
Telefoonnummer: 0641143752
Mobiel nummer: 0641143752
Privé emailadres: sh.debakker@gmail.com

Opleiding: Technische Informatica
Locatie: Delft
Variant: voltijd

Naam studieloopbaanbegeleider: Dave Stikkorum
Naam begeleidend examiner: Cobie van der Hoek
Naam tweede examiner: Henk van den Bosch

Naam bedrijf: Solware B.V.
Afdeling bedrijf: Ontwikkeling
Bezoekadres bedrijf: Jupiter 220
Postcode bezoekadres: 2675LV
Postbusnummer: 128
Postcode postbusnummer: 2670AC
Plaats: Honselersdijk
Telefoon bedrijf: 088-1600700
Telefax bedrijf:
Internetsite bedrijf: www.solware.nl

Achternaam opdrachtgever: dhr Micheal van Lier
Voorletters opdrachtgever: M.
Titulatuur opdrachtgever: CEO
Functie opdrachtgever: CEO
Doorkiesnummer opdrachtgever: 0622055680
Email opdrachtgever: m.vanlier@solware.nl

Achternaam bedrijfsmentor: dhr Michael van Lier
Voorletters bedrijfsmentor: M.
Titulatuur bedrijfsmentor: CEO
Functie bedrijfsmentor: CEO
Doorkiesnummer bedrijfsmentor: 0622055680
Email bedrijfsmentor: m.vanlier@solware.nl

Doorkiesnummer afstudeerder: 0641143752
Functie afstudeerder (deeltijd/duaal):

Titel afstudeeropdracht:

Het ontwikkelen van een applicatie waar gebruikers onderhoudswerkzaamheden mee kunnen inplannen bij vastgoedbeheerders, de opdracht wordt uitgevoerd door Solware B.V.

Opdrachtomschrijving

1. Bedrijf

Solware ontwikkelt en exploiteert high impact apps. De web- en mobiele apps van Solware hebben een grote impact op het primaire proces van onze toonaangevende klanten. Solware is gevestigd in Honselersdijk. Solware heeft een opdracht ontvangen van vastgoedbeheerders die door de afstudeerder uitgevoerd mag worden.

Met het uitvoeren van de afstudeeropdracht wordt een mogelijkheid bereikt waarmee het voor vastgoedbeheerders eenvoudig wordt gemaakt om werkzaamheden bij te houden voor zijn klanten (bewoners). Bewoners kunnen op dit platform hun werkzaamheden toevoegen zodat de vastgoedbeheerder deze werkzaamheden niet verder meer hoeft te verwerken, deze werkzaamheden kunnen zodanig direct worden ingepland.

Van de afstudeerder wordt verwacht dat een mobiele applicatie ontwikkeld wordt, belangrijke onderdelen om onderzoek naar te doen is de beveiliging en performance van deze applicatie. Hierbij kan gedacht worden aan het beveiligd versturen van gevoelige gegevens (persoonsgegevens), encryptie zal daarom gebruikt moeten worden. Ook moeten foto's worden verstuurd naar de server die van goede kwaliteit zijn maar snel verzonden moeten worden. Een efficiënte foto bewerking bibliotheek zal daarom moeten worden geschreven die ervoor zorgt dat afbeelding optimaal verzonden kan worden. Uiteindelijk zal een voorstel worden opgesteld voor de beveiliging van data op het platform iOS. Dit beschrijft hoe data op de telefoon het beste wordt beveiligd en hoe dit het beste kan worden verstuurd.

De performance van de applicatie is ook van belang, de GUI mag niet vastlopen wat anders voor een slechte ervaring zorgt. Daarom zal goed nagedacht moet worden over gebruik van background Threads voor een snellere uitvoering van langdurige acties.

Voor dit gedistribueerde systeem is ook de server van belang. De web API die door de client gebruikt gaat worden zal gemaakt moeten worden op basis van het REST protocol. Deze API zal geschreven moeten worden op een reeds bestaand platform.

De opdracht is nog niet gestart met de ontwikkeling, dat wil zeggen dat het vanaf de eerste stap moet worden uitgevoerd. Door gebruik van Agile Development wordt verwacht dat in samenwerking met de opdrachtgever werkende software wordt geleverd. Er is gekozen voor een Agile Development methodiek omdat Solware altijd al met deze ontwikkelmethodiek heeft gewerkt. Dit heeft vooral als reden omdat Agile als hoofddoel heeft een nauwe samenwerking tussen klant en ontwikkelaar door middel van korte iteraties. Dit in tegenstelling tot andere ontwikkelmethododes zoals RUP of IAD.

2. Probleemstelling

Vastgoedbeheerders zijn veel tijd kwijt aan aan bewoners in de vorm van gesprekken over problemen die in een huis aanwezig zijn, werkzaamheden worden op dit ingepland door middel van telefonisch contact met de vastgoedbeheerder. Op gebied van communicatie tussen de vastgoedbeheerder en bewoner is veel tijd te winnen door middel van automatisering.

3. Doelstelling van de afstudeeropdracht

De doelstelling van de afstudeeropdracht is het voor bewoners eenvoudiger maken om werkzaamheden voor vastgoedbeheerders in te plannen. Dit heeft als voordeel dat er minder direct contact nodig is met de vastgoedbeheerder. Afspraken kunnen worden ingepland waarna de gebruiker een notificatie krijgt, hierdoor

zal de communicatie sneller verlopen. Daarnaast wordt verwacht dat de gehele applicatie goed beveiligd is en alle communicatie ook beveiligd zal verlopen en zo efficiënt mogelijk zal verlopen.

Een tweede doelstelling is het creëren van een voorstel over de beveiliging opties die te gebruiken zijn voor het iOS platform. In het voorstel zal beschreven staan hoe informatie het beste beveiligd op de telefoon kan worden en hoe het beste versleutelde data kan worden verstuurd. Bij de op te leveren applicatie zal ook de benodigde documentatie worden toegevoegd naast het systeem ontwerp.

4. Resultaat

Het resultaat van de afstudeeropdracht is het opleveren van een mobiele applicatie voor het platform iOS inclusief documentatie en systeemontwerp, deze applicatie heeft als doel het vereenvoudigen van het inplannen van werkzaamheden in huizen van vastgoedbeheerders. Daarnaast zal ook een voorstel worden opgeleverd over de beveiligingsmethodes die gebruikt kunnen worden op het iOS platform.

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

De volgende werkzaamheden worden uitgevoerd om de opdracht met succes uit te kunnen voeren:

- Het schrijven van een Plan van Aanpak (2 dagen)
- Het vaststellen van requirements (5 dagen)
- Het onderzoeken naar de verschillende onderdelen van de applicatie, ontwerpen van de applicatie (Iteraties) (10 dagen)
- Ontwikkelen van de applicatie (Iteraties) (40 dagen)
- Testen van de applicatie (10 dagen)
- Communicatie met de opdrachtgever (5 dagen)

Tijdens de ontwikkeling van de applicatie krijgen we te maken met allerlei problemen zoals de security, performance en architectuur. Het is aan de student om te zorgen dat al deze problemen op de meest toepasselijke wijze worden opgelost. Om de twee weken wordt een iteratie opgeleverd, dat betekent dat de te maken applicatie dan aan een belangrijke functionaliteit uit de eisen voldoet.

Ook wordt verwacht dat er nauw wordt samengewerkt met de opdrachtgever om zo tot een applicatie te komen die zoveel mogelijk aan de eisen voldoet die de opdrachtgever heeft opgegeven.

Het testen van de applicatie moet volledig worden uitgevoerd om zo voor een robuuste applicatie te komen. Dit zal bereikt worden door verschillende test documenten te schrijven waarna ze uit te voeren.

De methode die voor de ontwikkeling van deze applicatie gebruikt gaat worden is Agile Development. Deze ontwikkelmethode is tijdens de opleiding kort behandeld, maar ik heb hier nog niet eerder een opdracht mee gemaakt.

De belangrijkste en meest interessante onderdelen die tijdens de ontwikkeling goed moeten worden onderzocht zijn de volgende:

- Een software bibliotheek die afbeelding bewerken mogelijk maakt en zonodig bewerkt op pixel niveau om zo tot de juiste afmetingen te komen. Dit is belangrijk om zo goede kwaliteit afbeeldingen te versturen die ook snel over een netwerk verstuurd kunnen worden.
- Een multithreaded systeem die alle langdurige taken op een efficiënte methode inplant en uitvoert op verschillende telefoon types. Elke telefoon die ondersteund wordt zal de applicatie soepel moeten kunnen laten verlopen, dit vereist veel testen onder verschillende systeem omstandigheden.
- Het te ontwerpen systeem moet uitbreidbaar zijn voor eventuele nieuwe versies van de software.
- De server moet geïmplementeerd worden, dit door middel van software die gebouwd wordt op een bestaand platform. De communicatie tussen server en client verloopt via de REST API.

6. Op te leveren (tussen)producten

Tijdens het uitvoeren van de afstudeeropdracht word het Plan van Aanpak en de verschillende ontwerpdocumenten tussentijds opgeleverd. Denk hierbij aan een klassediagram en ontwerp van hoe het communicatie tussen server - client verloopt. Als laatste wordt uiteraard de gehele applicatie inclusief source code opgeleverd met de daarbij behorende documentatie, testplan en ontwerpdocumenten.

7. Te demonstreren competenties en wijze waarop

De volgende competenties verwacht ik met mijn afstudeeropdracht te voltooien:

A1: Analyseren van het probleemdomein

Er zijn vanuit de opdrachtgever in grote lijnen functionele eisen opgegeven, met deze gegevens moet gekeken worden naar de precieze eisen. Daarnaast moet de server op een bestaand platform worden geïmplementeerd, dit vereist een onderzoek naar het bestaande platform om te ontdekken hoe dit in elkaar zit. Van dit alles wordt verwacht dat de afstudeerder dit zelfstandig kan uitvoeren, deze verantwoordelijkheid leidt tot het bereiken van niveau 3. Tijdens de opleiding is het uitvoeren van deze opdrachten nog nooit zelfstandig gebeurt.

A3: Achterhalen van behoeften van belanghebbenden

De precieze functionele eisen moeten nog worden vastgesteld, deze worden samengesteld in (2) wekelijks overleg met de opdrachtgever. In brede zin zijn de eisen al vastgesteld maar dit kan naar de behoefte van de opdrachtgever nog worden uitgebreid of aangepast. Doordat de afstudeerder zelf verantwoordelijk wordt gesteld voor dit contact, voldoet dit aan de eisen voor niveau 3. Goede communicatieve eigenschappen zijn nodig om dit contact goed te onderhouden. Deze eigenschappen zijn alleen naar voren gekomen in het bedrijfsproject in het derde jaar.

A5: Opstellen van systeemeisen

Voor het gehele systeem zullen eisen moeten worden opgesteld in de vorm van wat de App precies moet kunnen, daarnaast is het belangrijk dat alle niet-functionele eisen worden opgesteld en randvoorwaarden om er zo voor te zorgen dat de klant krijgt in de App wat hij daadwerkelijk wilt.

C8: Ontwerpen van een technisch informatie systeem

Zoals eerder geschreven moeten de systeemeisen van dit systeem worden uitgewerkt, er zal rekening moeten worden gehouden met eventuele randvoorwaarden. Daarnaast moet een geheel systeemontwerp worden gemaakt in de vorm van UML om het gehele systeem te omschrijven kwa werking. Ook dit onderdeel zal geheel door de afstudeerder moeten worden uitgewerkt.

C10: Het ontwerpen van een systeem architectuur

De gehele applicatie moet ontworpen worden met een achterliggend systeem, dit systeem moet nog in kaart worden gebracht en zal zo worden ontwikkeld dat het voor andere projecten eventueel herbruikbaar is. Er zal veel aandacht worden besteed aan het juist ontwerpen van bijvoorbeeld het klassediagram. Dit is een van de hoofd competenties voor deze afstudeeropdracht. Deze competentie is ook op niveau 3 omdat de afstudeerder op zichzelf een volledige architectuur moet ontwikkelen aan de client kant, dit zelfstandig uitvoeren is nog niet aan bod gekomen tijdens de studie.

C12: Ontwerpen van een gedistribueerd systeem

De gehele App moet zo efficiënt mogelijk kunnen samenwerken en communiceren met het systeem op de server. Daarom moeten beide delen van het gedistribueerde systeem zo worden ontworpen zodat hier allemaal rekening mee kan worden gehouden. Het gehele traject zal worden beschreven door flow charts om zo te beschrijven hoe, wanneer, en waarom bepaalde communicatie zal verlopen.

D16: Het realiseren van software

Om het volledige systeem te realiseren aan zowel de client en de server kant zal gebruik gemaakt worden van het ontwerp. Daarom zijn veel andere competenties afhankelijk van deze competentie. Deze competentie behaald niveau 3 omdat zowel een server als client gerealiseerd moeten worden, daarbij nadenkend zodat beide onderdelen efficiënt kunnen samenwerken.

D17: Testen van een software systeem

Zowel de infrastructuur als het gehele systeem zal moeten worden getest op de server en de client. Omdat de applicatie later echt gebruikt zal worden is het belangrijk dat de testfasen veel bugs aan het licht kunnen

brengen om op te worden gelost. Dit maakt deze competentie ook niveau 3 waardig omdat het door veel gebruikers zal worden gebruikt.

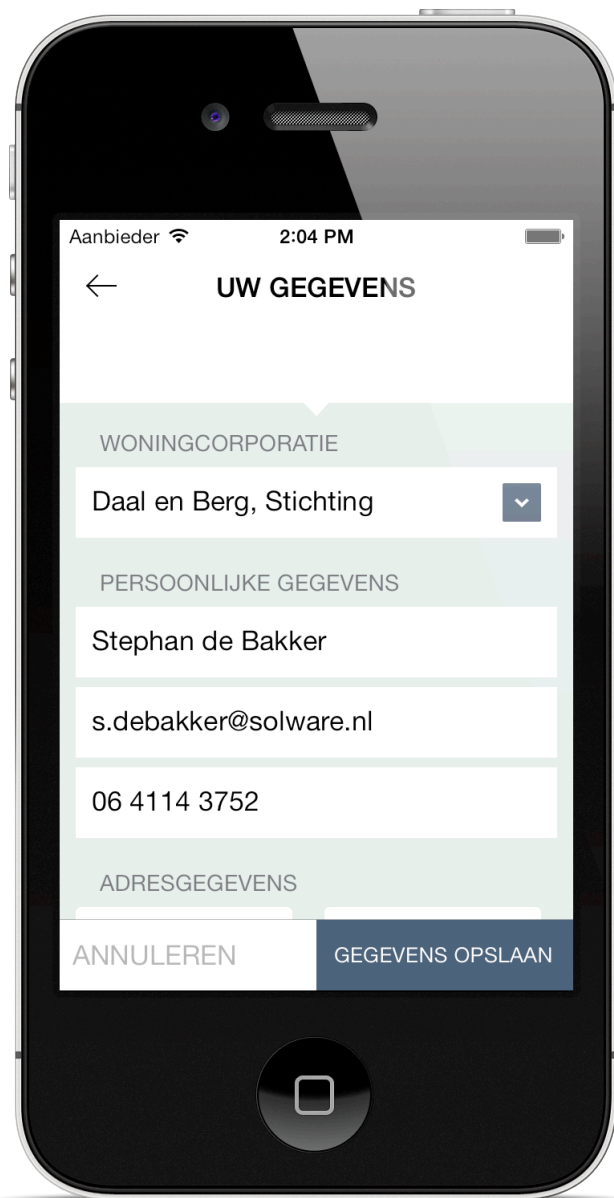
D18: Testen van een infrastructuur

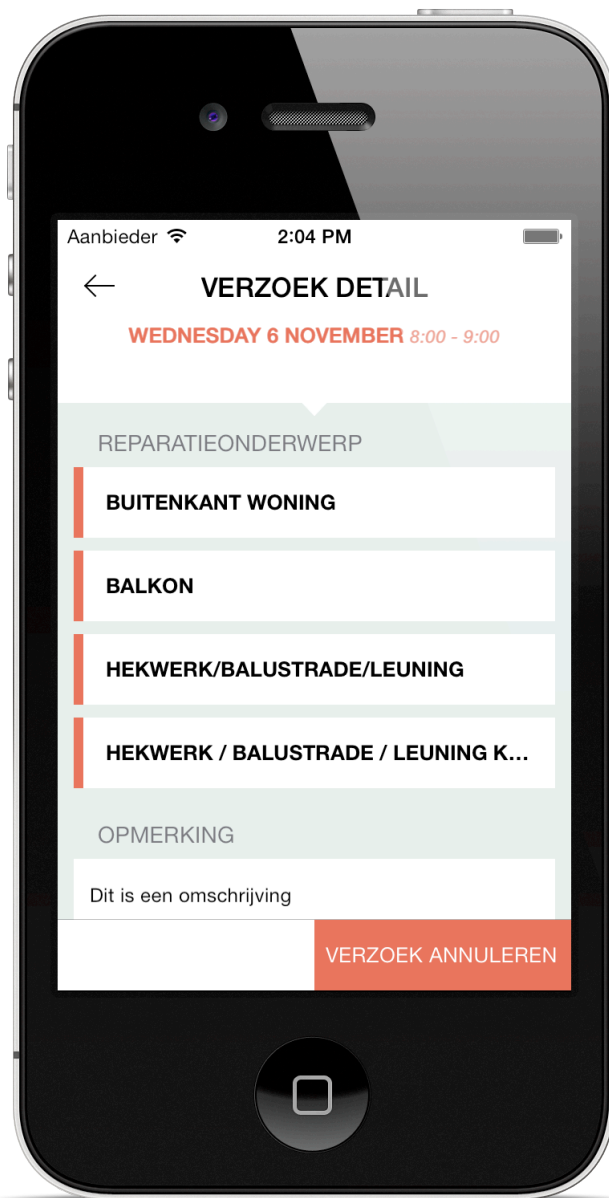
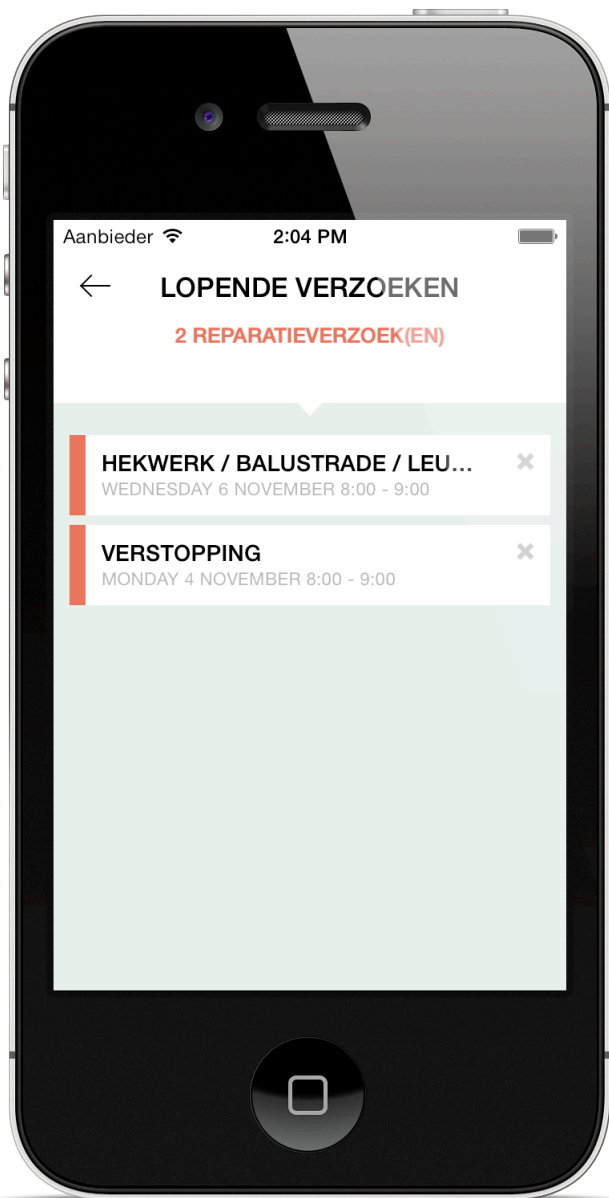
Het testen van de applicatie is ook een belangrijk gedeelte omdat de applicatie door veel mensen gebruikt zal worden, daardoor moeten de mogelijkheden dat de applicatie crasht volledig worden uitgesloten. Hiervoor is een gedocumenteerd testplan nodig met informatie over elke test case. Dit is te behalen in niveau 3 omdat door gebruik van Agile development er per increment moet worden getest. Dit wordt behaald door middel van testplannen en de hiervan behaalde resultaten verder te verwerken.

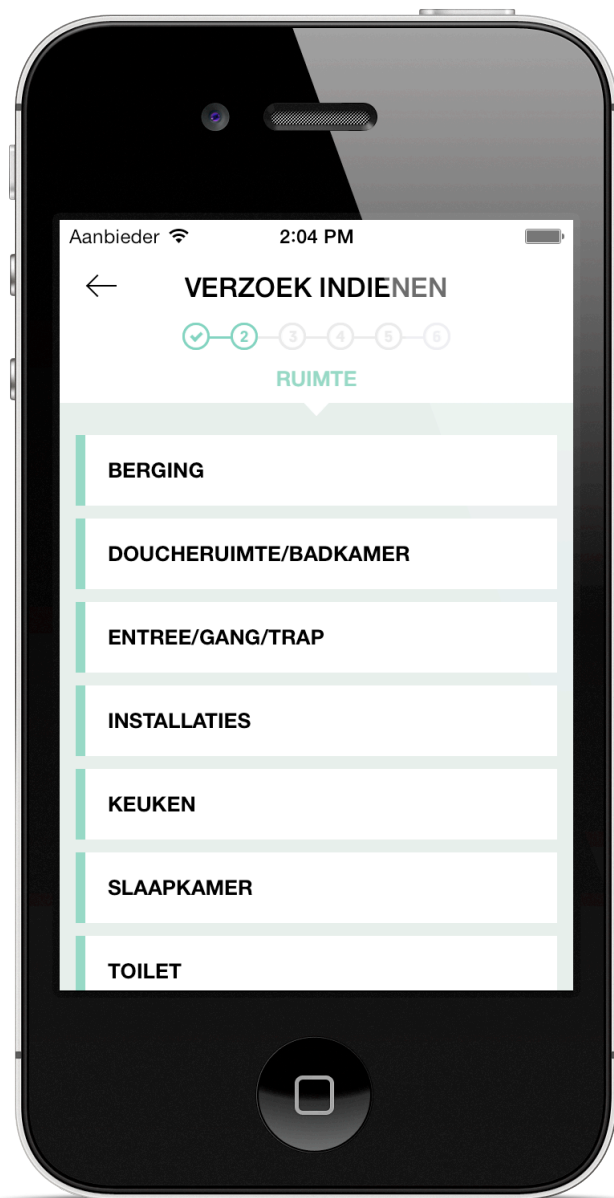
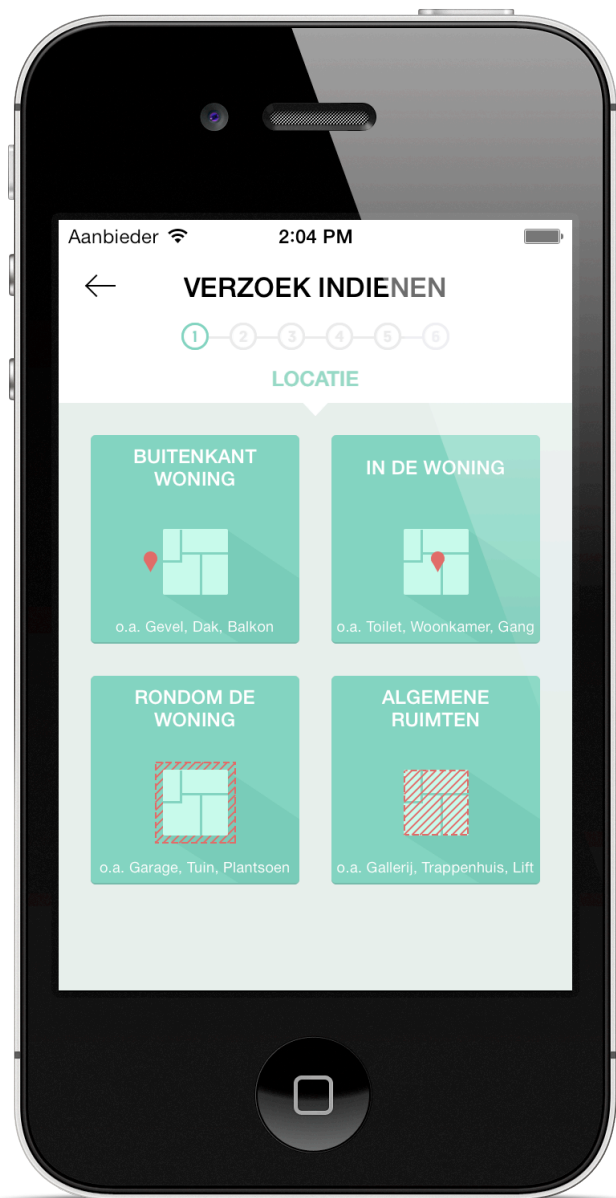
D19: Realiseren van een gedistribueerd systeem

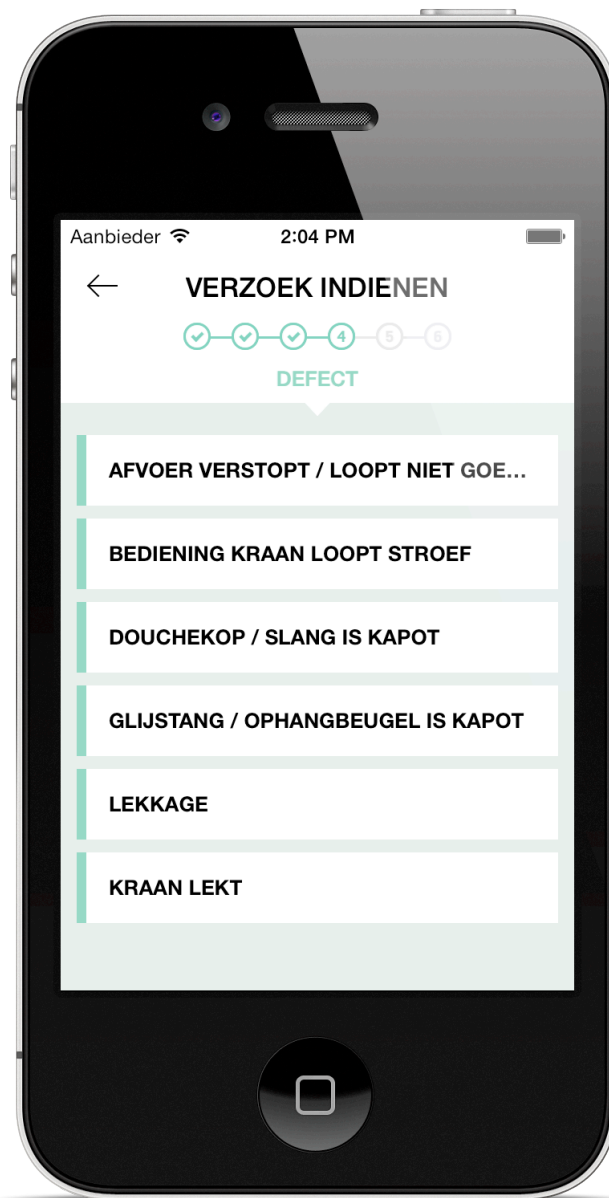
Deze competentie lijkt op het realiseren van een software systeem, maar focust zich meer op de infrastructuur van het gedistribueerde systeem dat ontworpen is. Volgens het ontwerp zal de server en client gerealiseerd worden.

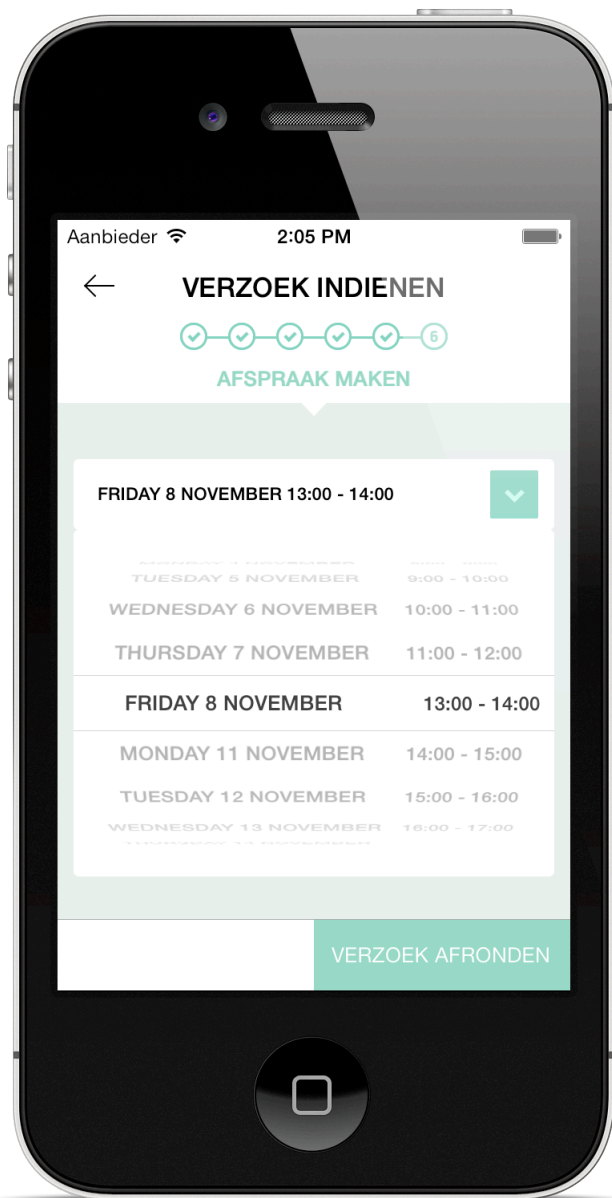
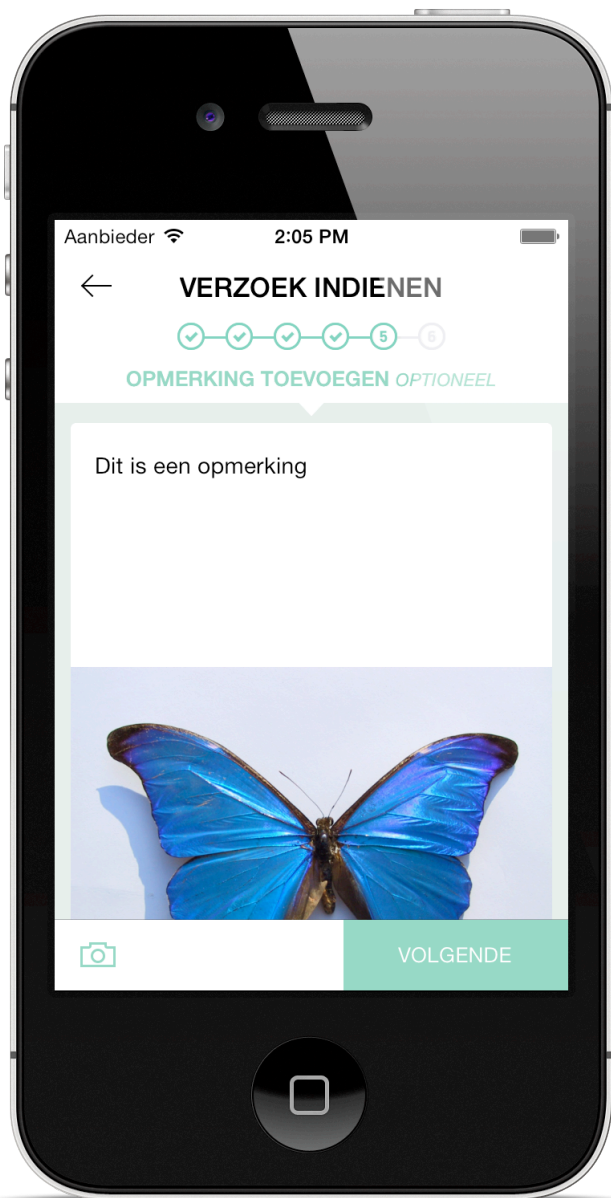
Voor meer informatie of wanneer de opdracht niet duidelijk mocht zijn qua technische waarde kunt u contact opnemen met dhr Michael van Lier (0622055680).

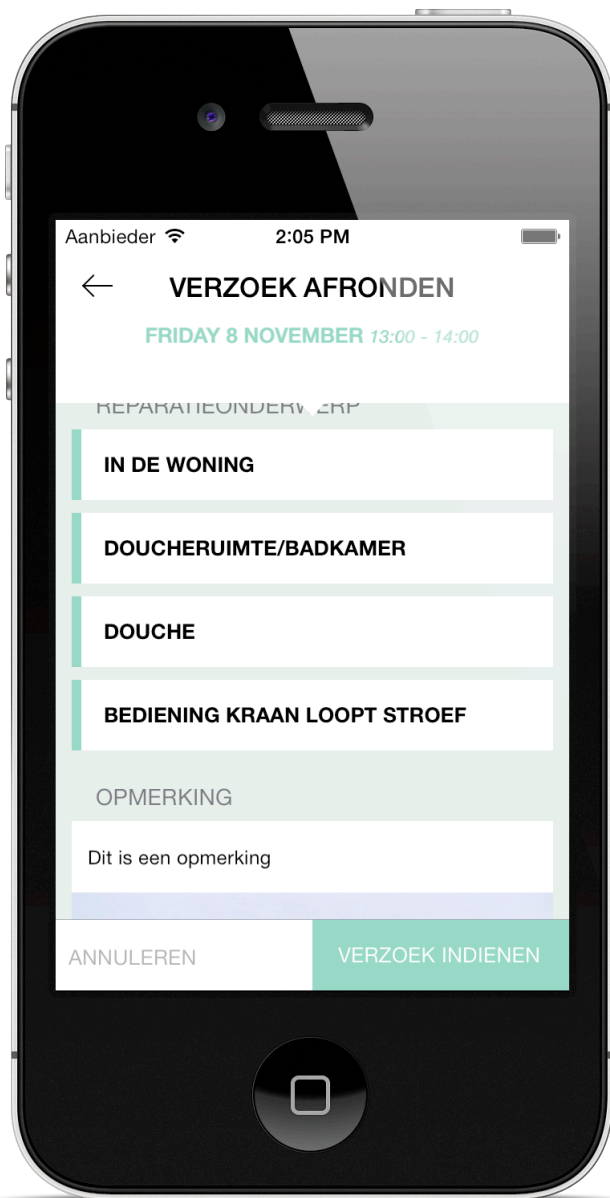












Bijlage Z - Competenties

In deze bijlage wordt beschreven op welke wijze de student de benodigde competenties zoals beschreven in het afstudeerplan (zie bijlage X) heeft uitgevoerd op het vereiste niveau.

A1: Analyseren van het probleemdomein & A3: Achterhalen van behoeften van belanghebbenden.

Tijdens het doorlopen van het project is aan deze competentie voldaan doordat er vanuit de opdrachtgever meer gedetailleerde informatie verkregen moest worden over de werking van de applicaties. Om dit voor elkaar te krijgen zijn afspraken gemaakt om over de voortgang van de applicaties te praten. Uit deze gesprekken met de opdrachtgever kwamen nieuwe (niet-) functionele eisen naar voren die het probleemdomein verder wisten te beschrijven. Hierdoor kon een beheersysteem en mobiele applicatie worden opgeleverd die aan de eisen van de opdrachtgever voldoen. Deze competentie is uitgevoerd op niveau 3 omdat er persoonlijk contact is gevoerd met de opdrachtgever in plaats van een fictieve opdrachtgever. Dit in tegenstelling tot de meeste opdrachten die studenten krijgen tijdens hun opleiding.

A5: Opstellen van systeemeisen.

Deze competentie sluit aan op A3 aangezien dit te maken heeft met het opstellen van functionele eisen, dit zijn de dingen die de applicatie moet kunnen. Het achterhalen van de systeem eisen is in deze competentie ook gedaan. Hierdoor is het opstellen van de functionele eisen op het vereiste niveau uitgevoerd. Het bewijs hiervoor is het gebruik van zogenaamde Product backlogs in combinatie met Sprint backlogs. De functionele eisen worden hier overzichtelijk bijgehouden, daarnaast is eenvoudig te zien wanneer een bepaalde functionele eise geïmplementeerd moet zijn aan de hand van de Sprint en prioriteit. Het op deze manier bijhouden en uitvoeren van functionele eisen is niet aangeleerd tijdens de studie maar is overgenomen van de werkwijze van Solware. Door het overnemen van deze efficiënte werkwijze kan worden bewezen dat de competentie op het vereiste niveau is uitgevoerd.

C8: Ontwerpen van een technisch informatie systeem.

Het uitwerken van de functionele eisen leidt tot het uiteindelijke ontwerp. Bij het ontwikkelen van het technische informatie systeem is een belangrijk ontwerp het use case diagram. Dit diagram beschrijft de werking van het systeem dat geschreven dient te worden. Door al deze diagrammen is het voor personen die niet bekend zijn met het systeem redelijk eenvoudig om de werking te begrijpen.

C10: Het ontwerpen van een systeem architectuur.

Bij het ontwerpen van de systeem architectuur behoren gedetailleerde diagrammen zoals een database ontwerp en klasse diagram van het systeem. In het afstudeerverslag worden een aantal diagrammen getoond die bewijzen dat aan deze competentie voldaan is. Voor zowel het server gedeelte als de mobiele applicatie is nagedacht over de implementatie van de code door de benodigde klasse diagrammen te maken.

C12: Ontwerpen van een gedistribueerd systeem.

Deze competentie lijkt op de zojuist genoemde competenties C8 en C10. Alledrie hebben ze te maken met het ontwerpen van een software systeem. Deze laatste competentie is ook aan voldaan. Dit werd aangetoond door de beschrijving van hoe een mobiele applicatie met de server

database praat door middel van een API. Door een documentatie van de API te maken is er ook een ontwerp beschikbaar dat het voor personen die onbekend zijn met het systeem eenvoudig maakt om het te begrijpen en eventueel uit te breiden.

D16: Het realiseren van software.

Deze competentie is een belangrijk onderdeel van het gehele afstudeertraject. De meeste tijd is gestoken in het implementeren van de ontwerpen om zo een werkend gedistribueerd systeem te kunnen realiseren. In de bijlagen zijn alle documenten met code terug te vinden. Doordat bij Solware alle code die geschreven is door de ontwikkelaars gereviewed wordt, kan worden aangetoond dat de kwaliteit van de software gewaarborgd is. Daarnaast kan gezegd worden dat deze stap op het vereiste niveau is uitgevoerd door de kwaliteit van de code en de uitgebreide documentatie die zich in de code bevindt en beschrijft hoe functies en methodes werken.

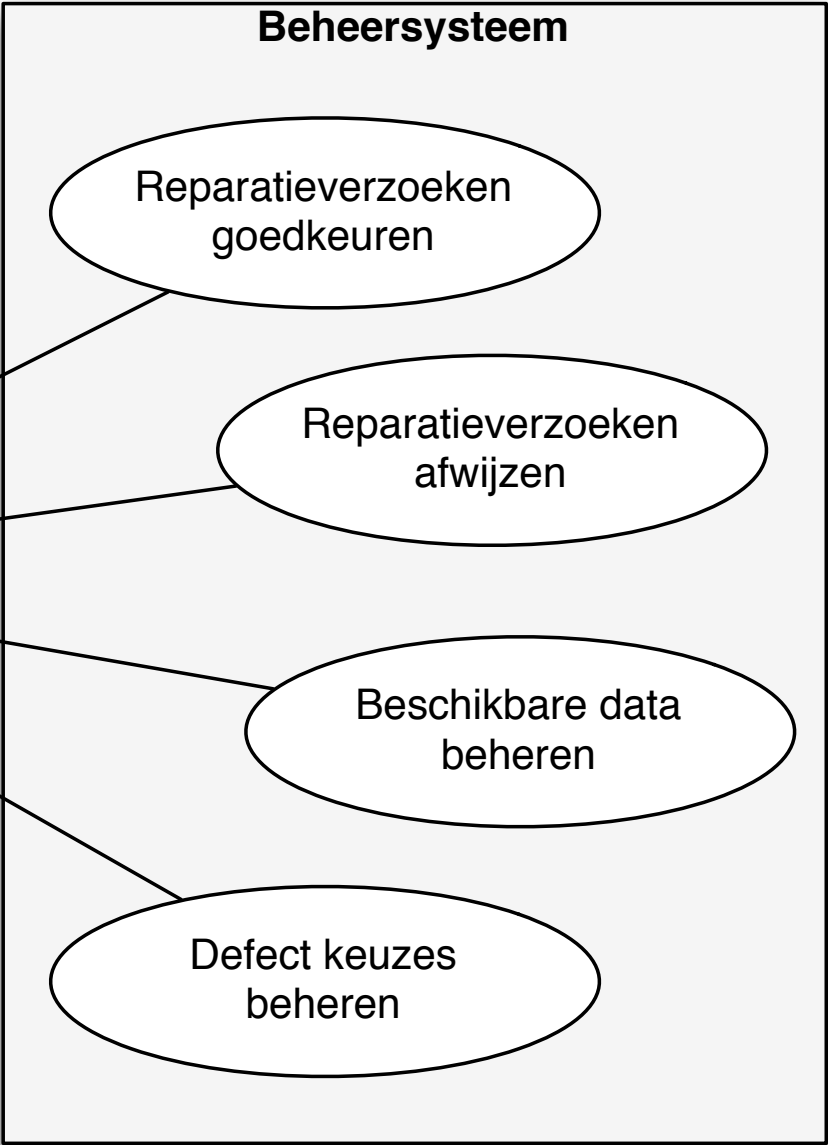
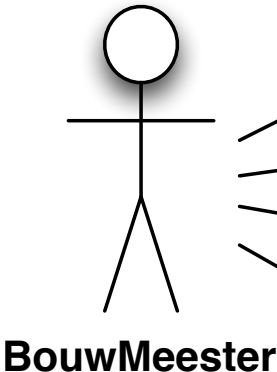
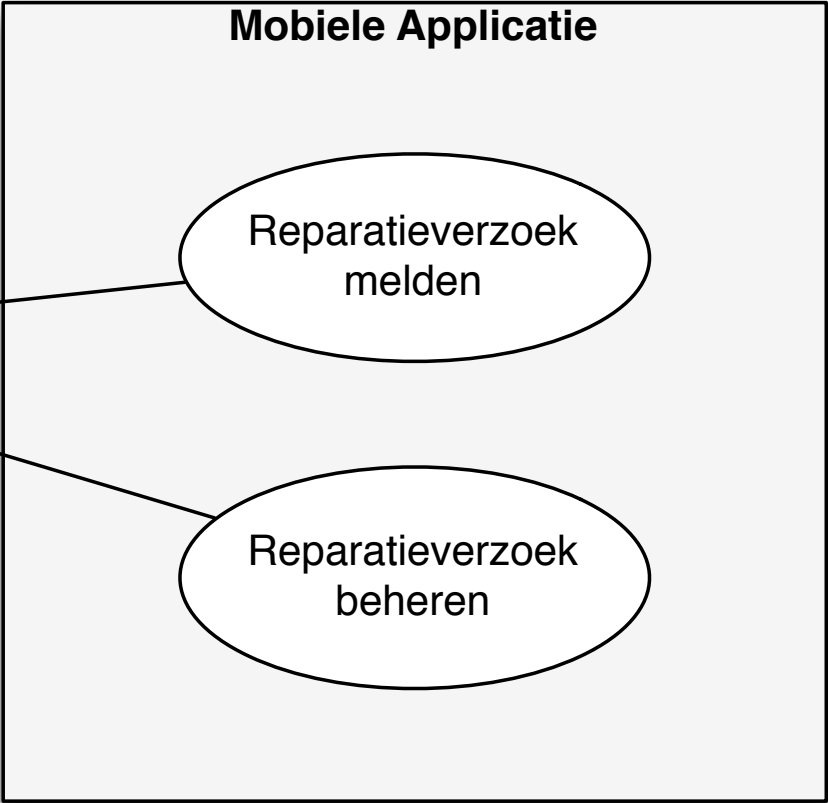
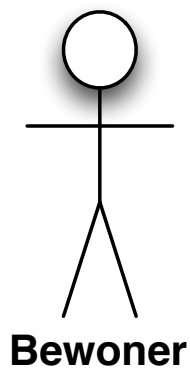
D17: Testen van een software systeem &

D18: Testen van een infrastructuur.

Het testen van het gehele systeem is niet helemaal zoals gepland verlopen. Doordat er meer tijd is gestoken in onderzoek naar snelheid en efficiëntie in de applicatie zijn niet alle belangrijke onderdelen naar behoren getest. Het was de bedoeling om fundamentele methodes en functies door middel van Unit Tests grondig te testen. Voor een gedeelte is dit gedaan met een aantal belangrijke methodes zoals validatie van de gegevens en de communicatie met de server. Dit laatste gedeelte hoort ook bij het testen van een infrastructuur. Voor de tests die zijn uitgevoerd is een testplan geschreven die te vinden is in bijlage W. De tests die zijn uitgevoerd bewijzen dat de code werkt, wat aangeeft dat het vereiste niveau voor die bepaalde tests is behaald.

D19: Realiseren van een gedistribueerd systeem.

Deze competentie is gericht op het realiseren van het gehele gedistribueerde systeem. Uiteindelijk is een volledig gedistribueerd systeem gerealiseerd bestaande uit een mobiele applicatie en een API inclusief database. Deze database kan ook benaderd worden door de opdrachtgever via een privé web interface. Het gehele systeem is opgeleverd in een status zodat het met elkaar kan communiceren. Daarnaast is de kwaliteit gewaarborgd door de eerder genoemde reviews van Solware.



Evaluatieformulier afstuderen

In te vullen door opdrachtgever c.q. bedrijfsmentor(en)

Student: [Stephan de Bakker](#)

Periode: [24-06-2013](#) tot en met [8-11-2013](#)

Bedrijf c.q. instelling: [Solware](#)

Bedrijfsmentor: [Chi-Fai Lam](#)

Plaats: [Honselersdijk](#)

Datum: [2-01-2014](#)

1. Heeft de student zich zelf snel en goed ingewerkt in het bedrijf en de uit te voeren afstudeeropdracht?

[Ja, heeft zich snel ingewerkt en zich een plek in het team verworven. Al vanaf dag 1 draaide hij goed mee met alle Scrum meetings.](#)

2. Hoe beoordeelt u de communicatieve vaardigheden van de student (in de samenwerking met collega's, in contacten met de opdrachtgever, bij mondelinge presentaties, schriftelijke rapportages)?

[Zijn communicatieve vaardigheden zijn goed. Stephan heeft iedere dag verteld waar hij mee bezig was tijdens het ochtendoverleg. Daarnaast heeft hij zelfstandig elke twee weken een interne productiepresentatie afgeleverd aan het team ter evaluatie van de voortgang. Richting de klant/opdrachtgever heeft hij met goed resultaat een demo](#)

gegeven voor de tussentijdse evaluatie en bij de uiteindelijke oplevering. En ook zijn verslag ziet er nu goed uit.

3. Hoe heeft de student tijdens het uitvoeren van de opdracht gefunctioneerd?

• Qua verantwoordelijkheid	goed
• Qua zelfstandigheid	goed
• Qua planmatig werken	goed
• Qua creativiteit	voldoende
• Qua productiviteit	voldoende
• Qua samenwerken met collega's	voldoende
• Qua draagvlakontwikkeling	voldoende
• Qua inspelen op bedrijfscultuur	goed
• Qua rekening houden met de specifieke context van het bedrijf	goed
• Qua het op gang brengen van de nodige veranderingen	goed

4. Hoe beoordeelt u de kennis en kunde van de student in verhouding tot wat u verwacht van een bijna afgestudeerde?

Stephan heeft de vaardigheden om zelfstandig een project te kunnen draaien en het gevraagde resultaat op te leveren. Gaandeweg heeft ook getoond dat hij goed met feedback kan omgaan die we op basis van tussentijdse evaluaties hebben verkregen. En dat zijn precies de kennis en vaardigheden die wij van een afstudeerder verlangen.

5. Hoe beoordeelt u de kwaliteit van de opgeleverde (tussen)producten?

De kwaliteit van de opgeleverde producten is voldoende. Mede dankzij code reviews van het team en de inzet van Stephan om het commentaar op te pakken, is de kwaliteit van de code gewaarborgd.

6. Bent u tevreden over het opgeleverde (eind)product?

- **In hoeverre heeft u gekregen wat is afgesproken?**

We hebben alles opgeleverd gekregen wat is afgesproken.

- **In hoeverre voldoet het (eind)product aan uw verwachtingen?**

Het eindproduct voldoet goed aan onze verwachtingen.

- **Wat is de bruikbaarheid en onderhoudbaarheid hiervan?**

Het eindproduct is zeer goed bruikbaar. Het doel is allereerst om meer feedback te krijgen. En De onderhoudbaarheid is voldoende.

- **Wat gebeurt er met het opgeleverde (eind)product?**

Allereerst feedback verzamelen op het opgeleverde eindproduct, daarna doorontwikkeling met uiteindelijk ingebruikname als einddoel.

- **Kunt u direct met het opgeleverde product aan de slag?**

Ja.

7. Zijn er nog aspecten voor u van belang die nog niet aan de orde zijn geweest?

Nee.

8. Bent u bereid een volgende keer weer uw medewerking te verlenen aan het beschikbaar stellen van een afstudeerplaats (graag met toelichting)?

Uiteraard, de samenwerking is tot nu toe goed bevallen. Het resultaat is goed en Stephan heeft naar eigen zeggen veel ervaring opgedaan tijdens zijn periode hier bij Solware. Dus er zijn enkel winnaars.