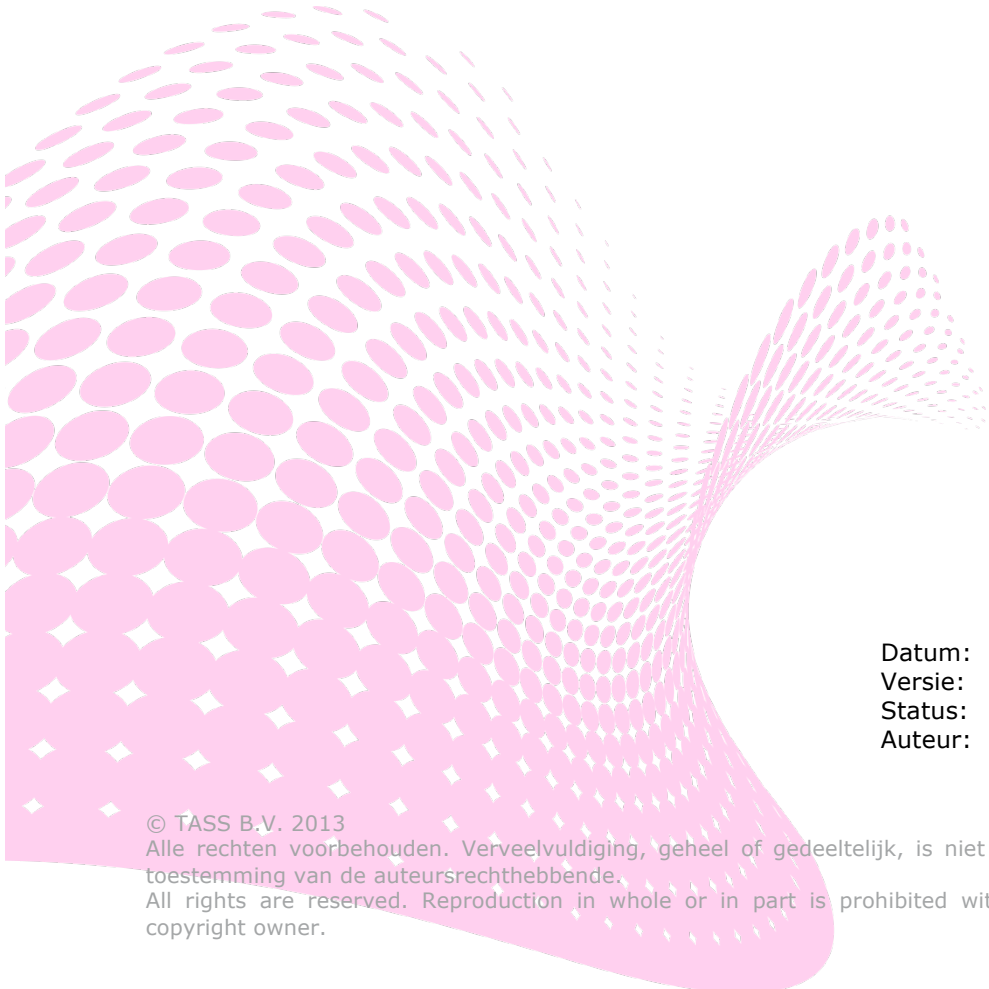


Passbook

De vervanging van de papieren koffiekaart



Datum:	11-06-2013
Versie:	1.1.0
Status:	Final
Auteur:	Matthijs van Driel

© TASS B.V. 2013

Alle rechten voorbehouden. Verveelvuldiging, geheel of gedeeltelijk, is niet toegestaan dan met schriftelijke toestemming van de auteursrechthebbende.

All rights are reserved. Reproduction in whole or in part is prohibited without the written consent of the copyright owner.

Titelblad Afstudeerscriptie

Gegevens student

Naam + voorletters	van Driel, M
Studentnummer	2160611
Afstudeerrichting	Software Engineering
Afstudeerperiode datum	04/02/2013 t/m 21/06/2013


Gegevens bedrijf

Naam bedrijf/instelling	TASS Technology Solutions
Afdeling	n.v.t
Plaats	Eindhoven
Naam + voorletters	Zielman, J
Functie van de bedrijfsbegeleidster	People Manager

Gegevens docentbegeleider

Naam + voorletters	Heijden, René M.J.W. van der
--------------------	------------------------------

Gegevens verslag

Titel afstudeerverslag	Passbook, de vervanging van de papieren koffiekaart
Datum uitgifte afstudeerverslag	11-06-2013
Indien van toepassing: vertrouwelijk	
Datum	11-06-2013
Getekend voor gezien door bedrijfsbegeleidster	

Voorwoord

Deze scriptie is geschreven naar aanleiding van het uitvoeren van de Passbook afstudeeropdracht bij TASS Technology Solutions. TASS is een dienstverlener op het gebied van technische software, zowel embedded als applicaties. (TASS, n.d.)

De opdracht moest duidelijk maken wat er allemaal mogelijk is met Passbook en hoe dit in zijn werk gaat.

Mijn eerste ontmoeting met TASS was op #devdate een evenement voor software ontwikkelaars, daarnaast ken ik het bedrijf vanuit Fontys Hogeschool. TASS is op Fontys een "Partner in Education". De afstudeeropdracht heb ik via het afstudeerboekje op de website van TASS gevonden. Na het lezen hiervan heb ik voor de Passbook opdracht gekozen. Vanwege mijn interesse in mobile, maar ook omdat ik hiervoor nieuwe technieken moet leren zoals Ruby on Rails en uiteraard Passbook.

Als eerste zou ik TASS willen bedanken dat ze mij deze kans hebben geven om deze opdracht uit te voeren. Daarnaast wil ik Josee Zielman (mijn bedrijfsbegeleidster) en Rob Knubben (mijn technisch begeleider) bedanken. Ook bedank ik alle andere afstudeerders bedanken voor de leuke tijd.

1 Inleiding	4
2 Het bedrijf	5
2.1 TASS	5
2.2 Geschiedenis	5
2.3 Organogram	5
3 De opdracht	6
3.1 Aanleiding	6
3.2 Toevoeging opdracht	6
3.3 Product Visie	6
3.4 Scope	7
3.5 Randvoorwaarden	7
4 Fasering en het Tien stappenplan	8
4.1 Oriëntatiefase	8
4.2 Ontwikkelingsfase	8
4.3 Invoeringsfase	8
5 Onderzoek	9
5.1 Voorbereiding	9
5.1.1 Kader & Doelstelling	9
5.1.2 Vraagstelling	9
5.1.3 Materiaal en Strategie	9
5.2 Resultaten	10
5.2.1 Welke beveiligingsmogelijkheden zijn er beschikbaar binnen Ruby on Rails?	10
5.2.2 Aan welke mogelijkheden zijn kosten verbonden en hoe hoog zijn deze?	10
5.2.3 Welke eisen stelt Apple aan de beveiliging van de Passbook kaarten?	10
5.2.4 Met welke mogelijkheid/mogelijkheden is er al ervaring binnen TASS?	10
5.3 Conclusie	11
6 Methoden	12
6.1 Scrum	12
6.1.1 Rollen	14
6.1.2 Sprint Planning	15
6.1.3 Einde sprint	15
6.2 Versie beheer	15
7 Tools	16
7.1 Xcode	16
7.2 Rubymine	16
7.3 Scrumwise	16
7.4 Visual Paradigm for UML	16
8 Ontwikkelfase	17

8.1 Sprint 1	17
8.1.1 Koffiekaart	17
8.1.2 Webservice	17
8.2 Sprint 2	18
8.2.1 Communicatie	18
8.2.2 Medewerkers Applicatie	18
8.2.3 Webservice	19
8.3 Sprint 3	20
8.3.1 Documentatie	20
Medewerkers-applicatie	20
8.3.2 Webservice	21
8.4 Sprint 4	22
8.4.1 Beveiliging	22
8.4.2 Calabash – Tests	23
8.4.3 Companion app	23
8.5 Sprint 5	24
8.5.1 Documentatie	24
8.5.2 Calabash Tests	24
8.5.3 Companion App	24
8.6 Sprint 6	25
8.6.1 Compantion App	25
8.6.2 HTTPS	25
8.7 Sprint 7	26
8.7.1 GUI Webservice	26
8.7.2 Compantion App	26
9 Invoeringsfase	27
10 Conclusie(s) en aanbeveling(en)	28
Evaluatie	29
Literatuurlijst	30
Bijlage I: Project Management Plan (PMP)	
Bijlage II: Requirements document	
Bijlage III: Onderzoek	

Samenvatting

TASS Technology Solutions is een dienstverlener op het gebied van technische software, zowel embedded als applicaties. TASS is vooral bekend als detacheerder, maar ontwikkelt zich waardoor er steeds vaker zelf projecten worden uitgevoerd. Er zijn zo'n 250 medewerkers bij TASS, die vooral vooral gedetacheerd zijn bij bijvoorbeeld bedrijven als ASML en Philips. TASS is onderdeel van Total Specific Solutions (TSS), één van de grootste niet-beursgenoteerde IT-ondernemingen van Nederland. (TASS, n.d.)

De afstudeeropdracht beschreven in deze scriptie gaat over het opzetten en realiseren van een Passbook oplossing. Dit is een nieuwe technologie van Apple die in 2012 is geïntroduceerd. Het bestaat uit een Passbook applicatie waarin digitale kaarten geplaatst kunnen worden. Deze kaarten kunnen bijvoorbeeld een vervanging zijn voor vliegtickets of een klantenkaarten. TASS wilde weten hoe Passbook werkt en wat de mogelijkheden ervan zijn. Omdat er nog geen bestaande opzet is moest alles van de grond af aan worden opgebouwd. De complete opzet van het project bestaat uit drie applicaties en de Passbook kaart, een webservice gemaakt in Ruby on Rails en een iOS-applicatie voor de iPad.

Met dit project is bewezen dat Passbook een goed alternatief is voor de papieren koffiekaart. Hij is makkelijk in gebruik en biedt een aantal voordelen over de papieren versie. Zo kunnen klanten hun kaart niet kwijtraken. De kaart kan worden opgeladen en kan op meerdere apparaten geïnstalleerd worden. Hierdoor heeft de klant zijn koffiekaart altijd bij zich.

Om alles goed te beveiligen is hier een onderzoek naar gedaan. hierin de volgende hoofdvraag beantwoord: *Hoe kan de toegang en verbinding met de Shakies webservice het beste beveiligd worden zodat alleen gemachtigde personen en kaarten toegang krijgen?* Door het doen van bureauonderzoek en het beantwoorden van alle deelvragen werd duidelijk wat de beste manier is voor de beveiliging. Er is gekozen voor OAuth2.0 in combinatie met SSL. Hierdoor blijft het werkbaar voor de gebruikers van de applicatie terwijl de veiligheid gewaarborgd wordt. De opdrachtgever is hiermee akkoord gegaan waarna dit gerealiseerd is.

Summary

TASS Technology Solutions is a service provider in technical software, both in embedded and applications. TASS mainly provides flexible staffing for other companies, but the company is starting to do more in house projects. There are about 250 employs working for TASS, most of them are working at companies as ASML and Philips. TASS is part of Total Specific Solutions (TSS), one of the biggest private IT-company's of the Netherlands.

The graduation project is described in this thesis, its about the setting up and creating a Passbook solution. This is a new technology introduced by Apple in 2012. It consists of the Passbook application where digital cards can be placed. The cards can replace this like airplane tickets or customer loyalty cards. TASS wants to know how Passbook works and what the possibility's are. Because there was no existing codebase everything is build from the ground up. The complete solution consists of three applications, the webservice made in ruby on rails, two iOS applications and the Passbook card.

The project proves that Passbook is a good alternative for the paper version of the Shakies coffee card. The card is easy in use and offers multiple benefits compared to the paper version. Customers can't lose there card, the card can be charged to increase the balance and it can be installed on multiple devices. This way a customer always has the card with him or her.

To find out what the best way of securing the applications, some research was done. The main question answered is: What is the best way to secure the authentication and the connection of the Shakies webservice so that only authorised people and cards can get access? By doing desk research and answering all the sub questions it became clear that OAuth2.0 in combination with SSL is the best solution. By using these two together the applications remain workable while the security is guaranteed.

Verklarende woordenlijst

Term	Definitie
iOS	iOS is het besturingssysteem dat door Apple gebruikt wordt op de iPhone, iPod touch en de iPad
Webservice	Een webservice is een applicatie die toegankelijk is vanaf het internet en biedt vaak informatie aan die dan ergens anders getoond wordt.
Ruby on Rails	Ruby is een programmeertaal, on Rails is boven op ruby gebouwd en is bedoeld voor het maken van een webservice's en websites.
Gem	Een gem is een library of framework dat gebruikt word binnen Ruby on Rails.
Shakies	Shakies is een locatie op NS stations waar dranken gekocht kunnen worden.
Algoritme	Een Algoritme is een methode binnen software met een bepaald doel. Bijvoorbeeld het sorteren van een lijst.
Framework	Een framework is een stuk computer software wat gebruikt kan worden door externa applicaties. Bijvoorbeeld het PassKit framework van Apple stelt een ontwikkelaar in staat om gegevens uit te lezen van de Passbook-applicatie.
OAuth	OAuth staat voor Open Authorization. Dit is een open standaard gemaakt voor autorisatie tussen een server en een client.
Embedded software	Embedded software is gemaakt voor het bedienen van hardware en draait vaak op gespecialiseerde hardware.
App	App is een afkorting van Applicatie en wordt veel gebruik als aanduiding van een Applicatie op een smartphone
Server	Een server is een computer die vooral bedoel is voor het draaien van applicaties die veel online moeten zijn. Bijvoorbeeld een e-mailserver.
API	Een application programming interface (API) is een verzameling definities op basis waarvan een computerprogramma kan communiceren met een ander programma of onderdeel. (Wikipedia, 2013)

1 Inleiding

Je kent het wel een portemonnee vol pasjes, haast elke winkel heeft er een. Maar wat als al deze pasjes vervangen kunnen worden voor een digitale variant op je telefoon? Dit is mogelijk met Passbook. Passbook is een standaard applicatie op de iPhone en iPod touch. Het aantal mensen dat gebruik kan maken van Passbook is enorm. Wereldwijd zijn er namelijk er 350 miljoen potentiële apparaten die Passbook ondersteunen. Daarom wil TASS weten hoe deze service precies werkt en wat alle mogelijkheden hiervan zijn.

TASS Technology Solutions is een dienstverlener op het gebied van technische software, zowel embedded als applicaties. Het bedrijf is vooral bekend als detacheerder, en voert daarnaast ook in-house projecten uit. (TASS, n.d.)

Voor dit project worden drie applicaties en een Passbookkaart gerealiseerd. Deze applicaties zullen allemaal communiceren met de webservice. Het project is uitgevoerd met gebruik van het tien stappenplan. Daardoor is het project opgedeeld in drie fases: de oriëntatiefase, realisatiefase en de invoeringsfase. Bij het lezen van het verslag zal duidelijk worden wat er in elke fase is gedaan.

In hoofdstuk 2 zal meer worden verteld over het bedrijf: TASS, hoofdstuk 3 gaat over de opdracht, wat is de aanleiding?, hoofdstuk 4 over de opzet van het project, het tien stappen plan en de fasering, in hoofdstuk 5 komt het onderzoek naar de beveiliging van de webservice aan bod, hoofdstuk 6 en 7 gaan over de gebruikte methoden en tools, hoofdstuk 8 gaat over de ontwikkelfase, hierin word het product gerealiseerd, in hoofdstuk 9 wordt de invoeringsfase besproken en hoofdstuk 10 zal gaan over de conclusies en aanbevelingen.

2 Het bedrijf

2.1 TASS

TASS is een dienstverlener op het gebied van technische en embedded software. De dienstverlening wordt daarbij afgestemd op de wensen van de opdrachtgever en de aard van de oplossing. Daarnaast neemt TASS deel als volwaardig partner deel aan productontwikkelingstrajecten of zijn ze zelf opdrachtgever hiervan (TASS, n.d.).

2.2 Geschiedenis

TASS is opgericht in 1978 als onderdeel van Philips maar is vanaf 2007 verder gegaan als een zelfstandige organisatie. Sindsdien is TASS uitgegroeid tot een bedrijf met circa 250 medewerkers, die vanuit twee vestigingen in Nederland (Eindhoven en Apeldoorn) werken en vanuit twee in België (Gent en Leuven).

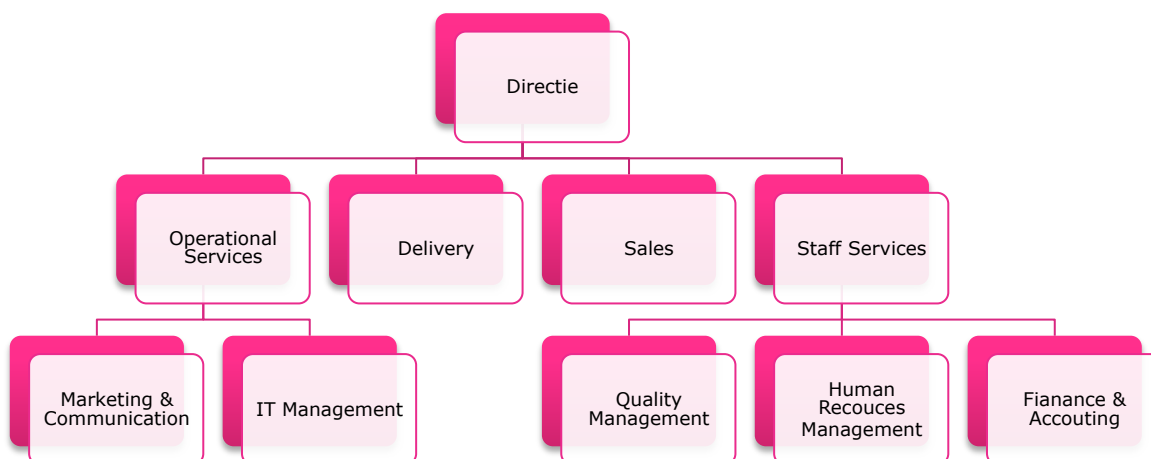
TASS is sinds 2007 onderdeel van Total Specific Solutions (TSS), één van de grootste niet-beursgenoteerde IT-ondernemingen van Nederland. (TASS, n.d.)



Figuur 2-1 Logo TASS binnen kantoor

2.3 Organogram

Binnen TASS vallen de afstuderende studenten onder 'delivery' zoals hieronder getoond. De studenten hebben geen externe klant, maar werken met een interne medewerker (meestal een Peoplemanager). De meeste studenten werken –dertien in totaal– werken vanuit het kantoor in Eindhoven, van februari tot en met juni aan hun afstudeeropdracht, maar een aantal is actief in Apeldoorn. (TASS, 2013)



Figuur 2-2 Organogram TASS

3 De opdracht

3.1 Aanleiding

Passbook is een nieuwe manier om bijvoorbeeld instapkaarten, klantenkaarten, kortingsbonnen en bioscoopkaartjes gemakkelijk bij je te dragen. Het is geïntroduceerd met de nieuwe iOS versie 6.0 en is beschikbaar op de iPhone en iPod touch. Er zijn al veel bedrijven die gebruik maken van deze functionaliteit, dit gaat van vliegmaatschappijen met vliegtickets tot kortingsbonnen van de Dixons.

TASS is geïnteresseerd in de functionaliteiten en werking van Passbook. Hoe werkt het, en wat is er mogelijk?

Ook wil TASS meer ervaring opdoen het met maken van applicaties voor iOS, aangezien er op dit gebied nog weinig ervaring is. Bovendien moet er een werkende applicatie worden opgeleverd.

TASS is zelf klant van deze opdracht die intern uitgevoerd zal worden. Het resultaat wordt gebruikt als indicatie of de Passbook functionaliteit gebruikt kan worden in toekomstige projecten. (TASS B.V.)

3.2 Toevoeging opdracht

Tijdens de realisatie van de opdracht is alles zeer voorspoedig verlopen, waardoor deze eerder voltooid kon worden dan aanvankelijk gedacht. Hierdoor is in overleg met de opdrachtgever en technisch begeleider besloten om een extra element toe te voegen. Dit is een iOS applicatie waarin klanten zelf een nieuwe koffiekaart kunnen kopen of van een bestande koffiekaart het tegoed kunnen verhogen. Binnen de companion app wordt gebruikt gemaakt van in App aankopen functionaliteit die door Apple worden aangeboden. Hierdoor hoeven klanten geen extra account of betaal methodes te hebben, in plaats daarvan wordt gebruikt gemaakt van hun Apple ID.

3.3 Product Visie

De Passbookkaart moet de papieren versie van de koffiekaart gaan vervangen, waardoor klanten hun koffiekaart haast onmogelijk kwijt kunnen raken, wanneer dat toch gebeurt, kan deze gewoon opnieuw geïnstalleerd worden. Bovendien kunnen klanten sneller afrekenen door simpel hun kaart te laten scannen.

Voor het bedrijf zal het makkelijker worden om gegevens over de uitgegeven kaarten bij te houden en te analyseren. Ook kunnen er makkelijk aanbiedingen worden verstrekt aan klanten met een digitale koffiekaart.

3.4 Scope

Binnen de scope van het project valt het onderzoek naar de juiste beveiligingsoplossingen voor het beveiligen van de webservice. De uitkomst van het onderzoek zal worden meegenomen in het realiseren van de drie applicaties en de Passbook kaart:

- Koffiekaart (Passbook kaart)
- iOS companion app (iPhone)
- Medewerkers applicatie (iPad)
- Webservice.

Dit onderzoek is nodig om te kunnen bepalen wat de juiste manier is om gegevens te beschermen tegen oneigenlijke toegang tot de gegevens. Maar ook hoe de communicatie tussen de koffiekaart, Medewerkers-applicatie, companion app en de webservice beveiligd zal moet worden om af luisteren tegen te gaan.

3.5 Randvoorwaarden

De belangrijkste randvoorwaarden die de klant heeft opgesteld is dat het ontwikkelen van de companion app en medewerkers-applicatie binnen iOS gemaakt moeten worden. Daarnaast moet de webservice in Ruby on Rails ontwikkelt worden, dit omdat binnen TASS gewerkt word met het Chimera framework.

4 Fasering en het Tien stappenplan

Voor de opzet van het project is gebruikt gemaakt van het tien stappenplan. Er hiervoor gekozen zodat het project overzichtelijk wordt opgezet en het in een ordelijke manier kan worden uitgevoerd. Het project is daarom opgedeeld in drie fases: De oriëntatiefase (4.1), ontwikkelingsfase (4.2) en de invoeringsfase (4.3). Elke fase heeft zijn eigen doel en bouwt voort op de vorige fase.

4.1 Oriëntatiefase

De oriëntatiefase is bedoeld om kennis te maken met het bedrijf en het bestuderen van de opdracht. Ook wordt er gekeken of er is er al iets bestaands is of dat moet alles nog gemaakt moet worden.

Gedurende deze fase zijn er intake gespreken met de opdrachtgever, het inlezen en bestuderen van de huidige situatie. Het opstellen van het Project Management Plan (PMP), requirement specificatie en het uitvoeren van het onderzoek. Het onderzoek wordt beschreven in hoofdstuk 5.

Uit de gesprekken met de klant werd duidelijk wat er precies gemaakt moet worden en wat hiervoor de eisen zijn. Het opstellen van het PMP en de requirement specificatie volgde.

Het PMP beschrijft:

- Hoe het project is georganiseerd
- Wat het globale schema is en welke werkproducten wanneer worden geleverd
- Op welke aannames en randvoorwaarden het plan is gebaseerd
- Hoe de kwaliteit wordt gewaarborgd
- Afspraken over acceptatie en onderhoud

De requirement specificatie beschrijft:

- De functionele en niet-functionele requirements voor de te ontwikkelen software
- De eventuele relaties en interfaces van de software met andere producten
- Hoe de software gebruikt kan worden

De volledige uitwerkingen van het PMP en requirement specificatie zijn te vinden in de bijlages.

4.2 Ontwikkelingsfase

Tijdens de ontwikkelingsfase worden de producten gerealiseerd. De webservice, Passbook koffiekaart, medewerkers-applicatie en de companion app.

Deze fase wordt uitgevoerd met gebruik van scrum (zie hoofdstuk 6: Methoden). Hiermee wordt elke twee weken overlegd met de opdrachtgever, hierin wordt een demo gegeven en gecontroleerd of het project nog volgens schema verloopt.

4.3 Invoeringsfase

Gedurende de invoeringsfase worden de producten overgedragen aan TASS. Hierin worden eventuele onduidelijkheden nog duidelijk gemaakt en documentatie waar nodig aangepast. Ook zal de opdrachtgever tijdens deze fase bepalen of het geleverde product aan de eisen voldoet.

5 Onderzoek

5.1 Voorbereiding

5.1.1 Kader & Doelstelling

IT Beveiliging is een hot issue. De media besteedt hier steeds meer aandacht aan vanwege grootte hacks en inbraken. Het is daarom belangrijk om ervoor te zorgen dat alle gegevens van het bedrijf en haar klanten veilig zijn.

Er moet worden onderzocht welke technieken en methode de juiste beveiliging bieden, maar ook of er al eisen worden gesteld vanuit Apple en zo ja welke? Er zijn een veel technieken beschikbaar, maar welke zorgen ervoor dat alles werkbaar blijft en toch voldoende beveiliging bieden.

5.1.2 Vraagstelling

Hoofdvraag:

Hoe kan de toegang en verbinding met de Shakies webservice het beste beveiligd worden zodat alleen gemachtigde personen en kaarten toegang krijgen?

Deelvragen:

Om de hoofdvraag te kunnen beantwoorden zijn de volgende deelvragen opgesteld:

- Welke beveiligingsmogelijkheden zijn er beschikbaar binnen Ruby on Rails?
 - Bij deze vraag wordt gekeken welke beveiligingsmethoden en technieken worden ondersteund binnen Ruby on Rails.
- Aan welke mogelijkheden zijn kosten verbonden en hoe hoog zijn deze?
 - Bij deze vraag wordt gekeken of de beschikbare methodes en technieken geld kosten, en zo ja hoeveel.
- Welke eisen stelt Apple aan de beveiliging van de Passbookkaarten?
 - Hier wordt onderzocht welke eisen Apple stelt aan de beveiliging van de Passbookkaarten en de verbinding met de webservice.
- Met welke mogelijkheid/mogelijkheden is er al ervaring binnen TASS?
 - Hierbij wordt gekeken met welke methodes en technieken TASS al ervaring heeft en welke mogelijk al in gebruik zijn.

5.1.3 Materiaal en Strategie

Tijdens het onderzoek zal vooral gebruik worden gemaakt van digitale media, die opgezocht zal worden op het internet doormiddel van bureauonderzoek. Er word vooral naar digitale media gekeken omdat het onderwerp snel in beweging is waardoor boeken niet meer de juiste informatie hebben.

Er zal kritisch worden gekeken of de gevonden informatie ook nog ergens anders te vinden is. Daarnaast zal goed gelet worden op de datum van plaatsing. Hierdoor wordt de kwaliteit van een bron bekeken en vastgesteld of deze bruikbaar is voor het onderzoek.

5.2 Resultaten

De volledige uitwerking van het onderzoek is te vinden in de bijlage.

5.2.1 Welke beveiligingsmogelijkheden zijn er beschikbaar binnen Ruby on Rails?

Er zijn in totaal zes methode/technieken onderzocht:

- Gebruikersnaam en wachtwoord
- HTTP Basic Authentication
- Secure Sockets Layer - Transport Layer Security
- Certificaat gebaseerde verificatie
- Application programming interface Token
- Open Authorization

Allemaal hebben ze hun voor en nadelen. Deze worden afgewogen om tot een beslissing te komen. De kwesties die hieronder worden besproken spelen daar ook een rol in.

5.2.2 Aan welke mogelijkheden zijn kosten verbonden en hoe hoog zijn deze?

Alleen aan SSL/TLS zijn kosten verbonden. Dit komt doordat een geldig certificaat gekocht moet worden van een erkende certificaat autoriteit.

Er kan ook zelf een certificaat gemaakt worden, alleen word deze dan niet herkend als geldig door browsers en de Passbook applicatie.

5.2.3 Welke eisen stelt Apple aan de beveiliging van de Passbook kaarten?

Apple stelt een aantal eisen aan de Passbookkaart. Er moeten certificaten gebruikt worden die aangemaakt kunnen worden op de developer portal van Apple. Daarnaast verplicht Apple dat er een signeer algoritme gebruikt wordt. Het maakt niet uit in welke programmeertaal dit gedaan wordt als alle stappen maar worden doorlopen.

Voor de verbinding met de webservice stelt Apple een Application programming interface op. Hiervan kan niet worden afgeweken omdat dan de communicatie tussen de koffiekaart en de webservice niet meer mogelijk is.

Apple verplicht ook het gebruik van een geldig SSL/TLS-certificaat om een productie-omgeving te voorzien van een beveiligde verbinding.

5.2.4 Met welke mogelijkheid/mogelijkheden is er al ervaring binnen TASS?

Binnen TASS is ervaring met alle beveiligingsmethoden, maar er wordt momenteel gewerkt aan het Chimera framework. Binnen dit framework zitten verschillende technieken maar er is een sterke neiging naar OAuth2.0. Deze methode biedt het voordeel dat applicaties en gebruikers makkelijk de toegang onzegd kunnen worden.

5.3 Conclusie

Hoe kan de toegang en verbinding met de Shakies webservice het beste beveiligd worden zodat alleen gemachtigde personen en kaarten toegang krijgen?

Er is gebleken dat alle beveiligingsmethoden hun voor en nadelen hebben. Hierdoor zal de oplossing bestaan uit meerdere methoden. Voor alle verbindingen zal SSL/TLS verplicht zijn, hierdoor kunnen de verstuurd gegevens niet onderschept worden.

Aan de communicatie met de Passbookkaart valt er weinig te doen, Apple specificeert al de complete API en hoe deze beveiligd moet worden. Hieraan kan daarom ook niks veranderd worden.

Voor de communicatie met de medewerkers applicatie moet wel worden bepaalt wat er gebruikt gaat worden. Hiervoor zijn verschillende autorisatiemogelijkheden mogelijk. Er is gekozen om OAuth in combinatie met SSL te gebruiken. Door het gebruik van OAuth vult de gebruiker zijn of haar gegevens alleen in op een beveiligde omgeving waarna alleen nog tokens heen en weer gestuurd worden. Mocht een token gecompromitteerd worden, kan deze direct de toegang worden ontzegd zonder dat iemand anders hier last van heeft. Daarnaast wordt dit binnen het chimera framework ook gebruikt. Naast het gebruik van OAuth zou nog een API token gebruikt kunnen worden om alleen verzoeken met het juiste token toegang tot de webservice te geven.

De aanbeveling zal worden overleg met de klant waarna een beslissing wordt genomen.

6 Methoden

6.1 Scrum

Scrum is een agile software ontwikkelmethode. Er is voor scrum gekozen omdat TASS dit bijna bij alle projecten gebruikt en hierdoor veel ervaring met scrum aanwezig is, daarnaast zorgt scrum er ook voor dat het project overzichtelijk blijft. Er is hiervoor ook een aparte wiki gemaakt (genaamd sensei) waarin alles wordt uitgelegd, hierdoor zullen alle projecten scrum hetzelfde gebruiken. (TASS, 2013)

Het complete scrum proces bestaat uit 4 hoofd elementen:

Product backlog

Hierin bevindt zich wat er gedaan moet worden, vaak staat hier voor 2 sprints werk in. Een backlog item wordt ook wel een user story genoemd. Een user story definieert hoe een stuk programma zou moeten reageren, en door wie en waar het gedaan moet worden. In figuur 5-1 een voorbeeld.

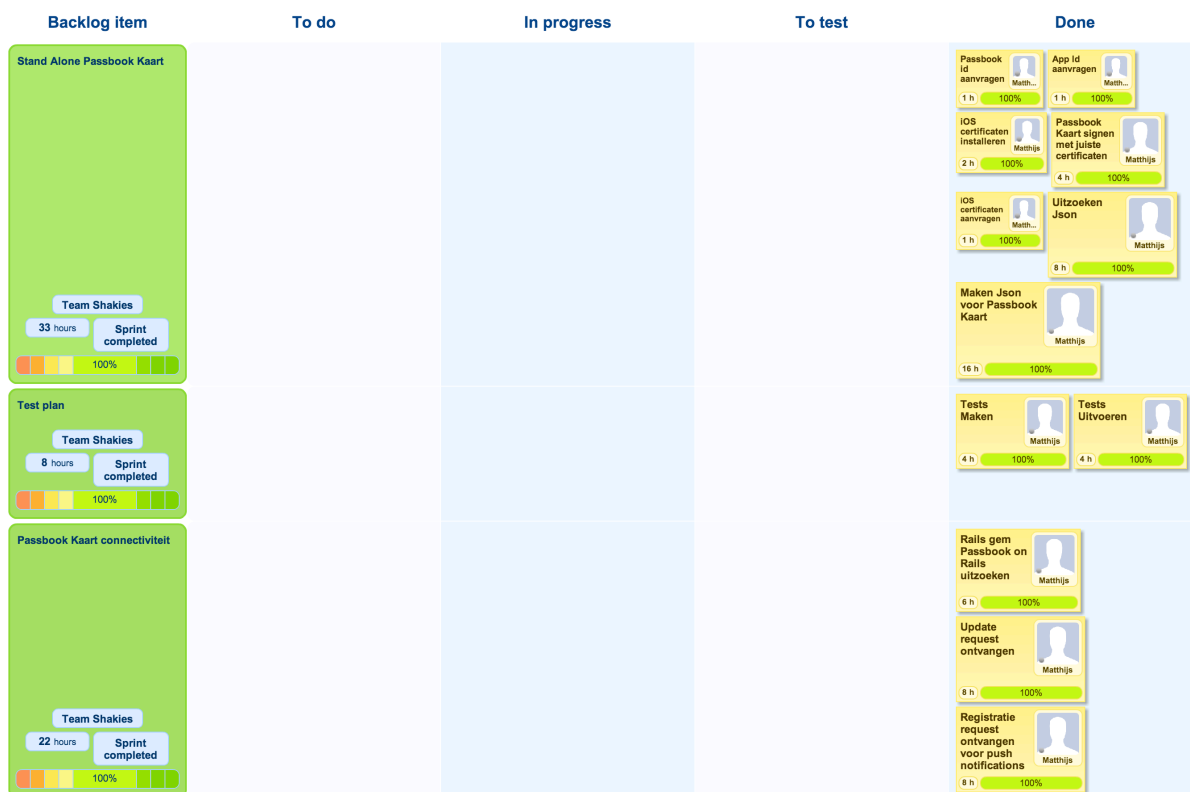
Feature	Passbook kaart barcode kunnen scannen met de front camera van de iPad	
ID	MW-1	
Story	Om	Een shakies koffiekaart te beheren
	Als een	Shakies Medewerker
	Wil ik	Een shakies koffiekaart scannen
	Scenario 1 Kaart word gescand met de front camera van de iPad	
	Gegeven	Klant heeft Passbook applicatie geopend en de shakies kaart geïnstalleerd
	En	Medewerker heeft de shakies applicatie geopend
	En	Medewerker heeft internet en de webservice is bereikbaar
	Wanneer	Klant laat zijn shakies koffiekaart zien
	Dan	Medewerker selecteert "Scan KoffieKaart"
	En	Medewerker houdt de QR code op de koffiekaart voor de camera.
	En	Medewerker ziet dat de iPad bezig is met het opzoeken van de klant gegevens.
	En	Medewerker krijg de klant gegevens te zien
	Scenario 2 Kaart kan niet worden gescand met de front camera van de iPad	
	Gegeven	Klant heeft Passbook applicatie geopend en de shakies koffiekaart geïnstalleerd
	En	Medewerker heeft de shakies applicatie geopend
	En	Medewerker heeft internet en de webservice is bereikbaar
	En	Klant laat zijn shakies koffiekaart zien
	En	Medewerker selecteert "Scan KoffieKaart"
	Wanneer	Medewerker houdt de QR code voor de camera.
	Dan	Kaart wordt niet herkend
	En	Medewerker krijgt een melding te zien dat de kaart niet herkent word

Figuur 6-1 User story

Sprint backlog

Voordat een sprint gestart wordt, wordt de sprint backlog gevuld met items uit de product backlog. Er wordt werk gepland voor de complete duur van de sprint en er worden taken toegewezen aan projectleden. Wanneer een project lid bezig is met een taak is dit te zien op het scrumbord.

In figuur 5-2 is een voorbeeld te zien van een scrumbord waarop alle taken zijn afgerond.



Figuur 6-2 Scrumbord waarop alle taken zijn afgerond (Scrumwise, 2013)

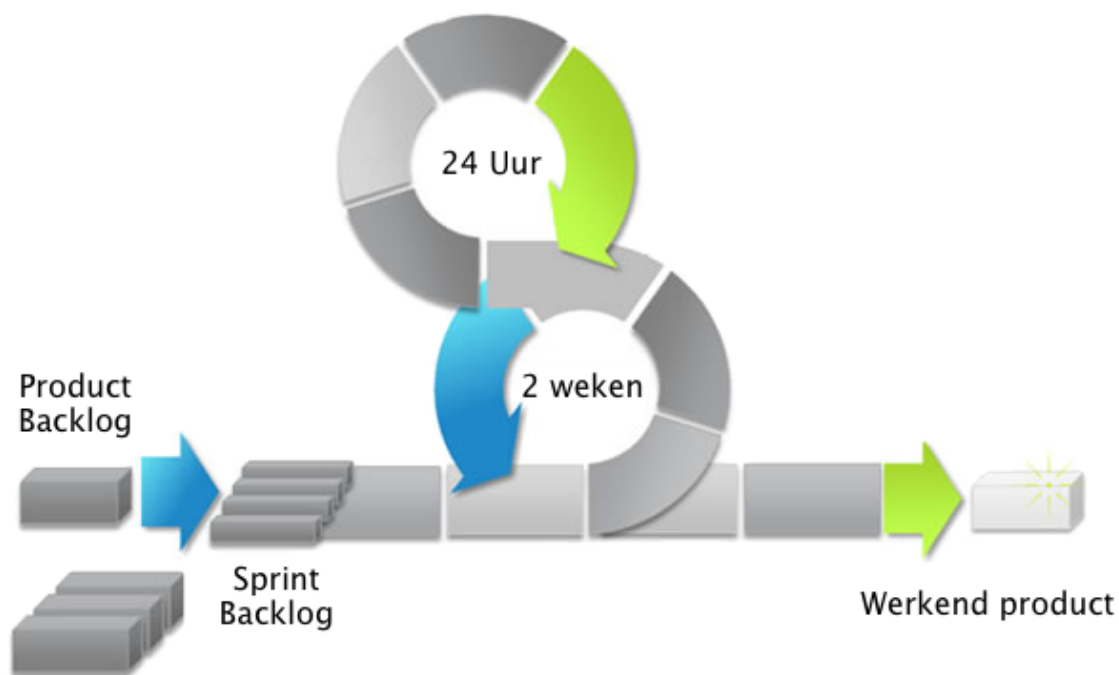
Sprint

Een sprint is een periode van meestal 1-4 weken. Tijdens de realisatie van shakies is er voor gekozen om elke sprint 2 weken te laten duren.

Om tot het eind product te komen zullen er meerdere sprints nodig zijn. Doordat de sprints relatief kort zijn kan er vaak overleg met de opdrachtgever plaatsvinden. Hierdoor wordt de kans verkleind dat het definitieve product niet aan de eisen van de opdrachtgever voldoet.

Standup meeting

Elke dag vindt er een standup plaats hierin vertelt elk teamlid wat er gisteren gedaan is, wat er vandaag gedaan gaat worden en of hierbij problemen zijn opgetreden. Tijdens de realisatie is de dagelijkse meeting niet gebruikt omdat het project door 1 persoon werd uitgevoerd.



Figuur 6-3 Scrum gevisualiseerd

6.1.1 Rollen

Voor scrum zijn er zijn een aantal rollen gedefinieerd:

Rol	Naam	Functie gedurende de afstudeer periode
Product Owner	Josee Zielman	Bedrijfsbegeleidster
Team member/ Scrum master	Matthijs van Driel	Afstudeerder
Stakeholder	Rob Knubben	Technisch begeleider

Product Owner

De product owner is verantwoordelijk voor het vullen van de product backlog. Ook bepaalt de product owner de prioriteit van de taken.

Team Member

De team member is verantwoordelijk voor het uitvoeren van de taken die zich in de sprint backlog bevinden.

Scrum Master

De scrum master houdt de planning bij, en controleert of door middel van een burndown chart of het team op schema ligt. Wanneer dit niet zo is probeert hij/zij er voor te zorgen dat het team terug op schema komt.

6.1.2 Sprint Planning

Aan het begin van elke sprint word de sprint backlog gevuld met taken, hierdoor word de sprint gepland. Het team kan planning poker spelen, waardoor duidelijk word of iedereen op één lijn zit. Hierdoor wordt een realistische uren schatting gemaakt van elke taak.

6.1.3 Einde sprint

Aan het einde van elke spint vind een retrospective, sprint review en een demo plaats.

Bij de sprint retospective word gekeken naar 3 dingen:

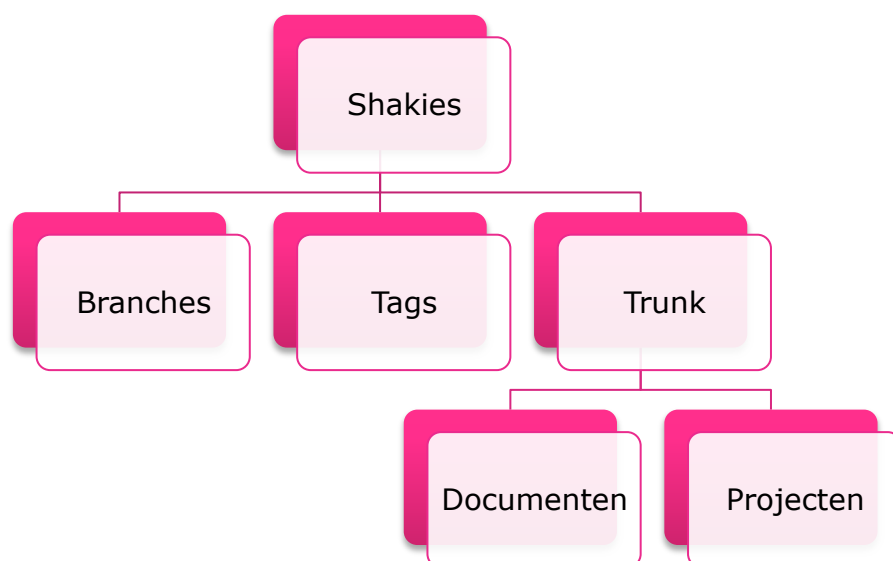
- Wat ging er goed?
 - Hier word opgeschreven wat er goed ging.
- Wat ging er niet goed?
 - Welke punten gingen niet goed? en waarom niet?
- Wat zou er beter kunnen?
 - Wat is er geleerd, en hoe kunnen we hiermee de volgende sprint beter laten verlopen.

6.2 Versie beheer

Voor versiebeheer zal subversion (SVN) gebruikt worden. SVN is een opensource versiebeheersysteem. Binnen TASS is hier ruimte ervaring mee, en worden de faciliteiten geboden om gebruik te maken van een SVN server. Hierop kunnen alle documenten en broncode worden beheerd. (Zennaware)

Het beheer word gedaan doormiddel van Cornerstone. Cornerstone is een SVN client voor Mac OSX.

De mappen structuur van SVN ziet er als volgt uitzien:



Figuur 6-4 Logo Cornerstone

Figuur 6-4 Structuur op SVN

7 Tools

Hier worden de meest gebruikte tools besproken.

7.1 Xcode

Voor het ontwikkelen van de iOS applicaties is gebruikt gemaakt van Xcode. Dit is de ontwikkel omgeving van Apple, waarin alle tools zijn verwerkt die nodig zijn voor het ontwikkelen van iOS-applicaties. Vanuit Xcode kan een applicatie worden gedeployed op een iOS-apparaat zoals de iPad, iPhone, iPod touch. Er al kennis was van Xcode, was het makkelijk om dit in gebruik te nemen. (Apple Inc.)



Figuur 7-1 Logo Xcode

7.2 Rubymine

Voor het ontwikkelen van de Ruby on Rails webservice is gekozen om Rubymine te gebruiken. Deze keuze is gevallen nadat er gekeken is naar verschillende integrated development environments (IDE). Daaruit bleek dat Rubymine beter werkte dan de Eclipse variant Aptana. Rubymine heeft een betere integratie met SVN en de database waardoor er geen verschillende tools nodig waren om deze te beheren en kon dit direct vanuit Rubymine gebeuren.



Figuur 7-2 Logo Rubymine

(Jetbrains)

7.3 Scrumwise

Scrumwise is een online scrum tool. Hierbinnen kan een backlog aangemaakt worden, waarna deze items aan sprints kunnen worden toegewezen. Binnen elk backlog item kunnen taken worden aangemaakt, zodat een groot backlog item verdeeld kan worden in kleinere taken. Bij elke taak wordt een urenregistratie bij gehouden. Hiermee wordt duidelijk hoe ver een taak is en of er teveel of te weinig uren voor een taak gepland zijn. Ook heeft scrumewise een taskboard, waarin kunnen de taken gemakkelijke van To do naar In progress en van To test naar Done worden gesleept. Alle uren worden aan het eind van iedere dag verwerkt in de burndown chart. (Scrumwise)



Figuur 7-3 Logo Scrumwise

7.4 Visual Paradigm for UML

Visual Paradigm for UML (VP) is een software pakket voor het maken van UML diagrammen. Er is gekozen om de diagrammen in VP te maken omdat hier al eerder ervaring mee is opgedaan tijdens de lessen op Fontys. Daarom worden er ook licenties beschikbaar gesteld aan studenten. (Visual Paradigm)



Figuur 7-4 Logo Visual Paradigm

8 Ontwikkelfase

8.1 Sprint 1

In sprint 1 is begonnen met het daadwerkelijke realiseren van het product. Er is tijdens deze sprint een start gemaakt met de koffiekaart en de webservice.

8.1.1 Koffiekaart

Er is –op de developer website van Apple– begonnen met het aanmaken van alle certificaten, passtypeid's, en app-ID's die nodig zijn voor het ontwikkelen.

De Passbook-koffiekaart is oorspronkelijk gemaakt zonder server connectiviteit, zodat eerst gefocust kon worden op het design van de koffiekaart. Hierdoor kon er snel met de klant gecommuniceerd worden wat ze van het design vonden en wat er nog zou moeten veranderen. Het gekozen design is te zien in figuur 7-1.

Om de koffiekaart daadwerkelijk toe te voegen aan de Passbook applicatie moet de koffiekaart gesigneerd worden. Dit gebeurt door middel van een tweetal certificaten en een algoritme van Apple. Hierna kan de koffiekaart worden toegevoegd op elk apparaat met een Passbook applicatie.



Figuur 8-1 Koffiekaart

8.1.2 Webservice

De webservice wordt gemaakt in Ruby on Rails, een eis die vanuit de klant wordt gesteld. Deze keuze is gemaakt omdat binnen TASS meerdere projecten hiermee werken, hiervoor worden binnen kort ook ontwikkelomgevingen opgezet die dienen voor het testen en draaien van applicaties en hun databases.

Om de communicatie tussen de webservice en de koffiekaart te realiseren, moet de webservice de API van Apple implementeren. Toen hieraan begonnen is, is er gekeken of hiervoor niet al een gem (een gem is een software library of framework) beschikbaar is voor Ruby on Rails. Er zijn drie gems gevonden die de Passbook api implementeren. Er is gekozen voor Passbook-ruby, die niet in beta verkeerde, en na het testen bleek dat de gem goed werkt. Daarnaast biedt het meer functionaliteit dan alleen het implementeren van de API. De gem ondersteunt ook het aanmaken en signeren van Passbook kaarten. Ook kunnen er verschillende kaarten en templates worden aangemaakt met verschillende lay-outs en functies. Hierdoor kunnen er meerdere type kaarten worden aangemaakt binnen het hetzelfde systeem. Zoals een kaart die dient als vliegticket of als kortingskaart. Een ander voordeel van deze gem is dat er niks naar de harde schijf wordt geschreven, maar in plaats daarvan wordt de kaart dynamisch aangemaakt wanneer deze wordt opgevraagd.

8.2 Sprint 2

In sprint 2 wordt verder gegaan met de communicatie tussen de koffiekaart en de webservice. Het doel is om ervoor te zorgen dat de kaart zich kon registreren en updaten.

8.2.1 Communicatie

Het eerste doel was om ervoor te zorgen dat de koffiekaart zich kon registreren bij de webservice. Wanneer een koffiekaart zich registreert worden de volgende gegevens opgeslagen in de database:

- Push token: Met een Push Token kunnen er push-notificaties gestuurd worden naar de koffiekaart, waardoor een update kan worden gestart vanuit de server wanneer er iets gewijzigd wordt wat relevant is voor de betreffende koffiekaart. Bijvoorbeeld het wijzigen van het tegoed.
- Datum van de laatste verandering: Dit wordt gebruikt om te controleren of de Passbook applicatie al beschik over de laatste versie van de koffiekaart. Wanneer dit het geval is, wordt verder niks gedownload, wat de gebruiker batterijduur en dataverbruik bespaart. Beschikt Passbook niet over een oudere versie dan wordt de kaart geüpdatet.

Omdat de Passbook-ruby gem deze Application programming interface (API) implementeert hoefde hier weinig aangedaan worden. Tijdens het testen van de API bleek echter een fout te zitten in het controleren van de datum, waardoor er altijd een nieuwe koffiekaart gedownload werd terwijl dit niet nodig was. De gem is opensource, dus is deze aangepast zodat de datumcontrole wel goed werkt. Door deze aanpassing werkt de gem nu zoals Apple dit dicteert in de API.

Om een koffiekaart te kunnen updaten vanaf de webservice, moet er gebruik worden gemaakt van de Apple Push Notification Service (APNS). Omdat de Push Token van de koffiekaart al beschikbaar is binnen de database, kan er een connectie naar de APNS worden geopend. Hierna worden alle push tokens doorgestuurd naar de servers van Apple, waarna zij deze zullen doorsturen naar de betreffende koffiekaart wanneer deze online is.

Normaalgesproken wordt er niet gegarandeerd dat push notificaties aankomen, maar voor notificaties voor Passbook wordt dit wel gedaan. De push notificaties zelf bevatten geen informatie, maar vertelt de Passbook applicatie dat er een update beschikbaar is. Er wordt dan aan de webservice gevraagd wat de updates zijn, waardoor geen informatie verloren gaat. (Apple Inc)

8.2.2 Medewerkers Applicatie

De medewerkers-applicatie (MWA) moet een iPad-applicatie worden, dit is ook een eis vanuit de klant.

De koffiekaart beschikt over een tweedimensionale streepjescode: QR-code. Deze QR-code moet door de medewerkers-applicatie kunnen worden uitgelezen. Dit zou moeten gebeuren door de camera aan de voorkant van de iPad. De klant wil de mogelijkheid hebben om de iPad vast te kunnen zetten in een toonbank, hierdoor zou de camera aan de achterkant van de iPad niet bruikbaar zou zijn. Doordat de camera aan de voorkant een lagere resolutie heeft, is deze eerst getest met applicaties die beschikbaar zijn binnen de App Store. De camera bleek gewoon verschillende QR-codes te kunnen uitlezen, dus het gebruik van de camera aan de voorkant leverde geen bezwaar op.

Voor het scannen van de QR-code is gebruik gemaakt van ZBar. Dit is een open source software suite beschikbaar voor meerdere platformen waaronder iOS voor de iPad. ZBar ondersteunt het scannen van QR-codes vanaf plaatjes, maar ook van live video. Dat maakt het ideaal voor de

toepassing van de medewerkers-applicatie. Omdat ZBar meer ondersteunt dan alleen QR codes, is het scannen van alle andere barcodes uitgeschakeld. Hierdoor wordt de QR code sneller herkent en gescand. (Brown, 2011)

De Graphical User Interface (GUI) is ook ontwikkeld, waardoor de klant kon zien hoe de flow van de applicatie zou worden voordat deze 100% was uitgewerkt.

8.2.3 Webservice

Om ervoor te zorgen dat de medewerkers-applicatie de gegevens van de gescande koffiekaart kan opvragen, is er een API gemaakt. Deze is specifiek voor de medewerkers-applicatie en maakt gebruik van JavaScript Object Notation (JSON) voor het oversturen van informatie. Via deze API kan de applicatie het gescande serienummer van de koffiekaart doorsturen naar de webservice, waarna deze kan antwoorden met de klantgegevens en het resterend tegoed van de koffiekaart. De medewerker kan zien of er nog tegoed op de kaart zit en de klant iets mag bestellen.

De medewerker kan vervolgens het tegoed aanpassen, waarna het zal worden terug gestuurd naar de webservice, waar het wordt opgeslagen in de database. Tegelijkertijd zal er een pushnotificatie worden gestuurd naar de koffiekaart waardoor het nieuwe tegoed ook daar zichtbaar wordt.

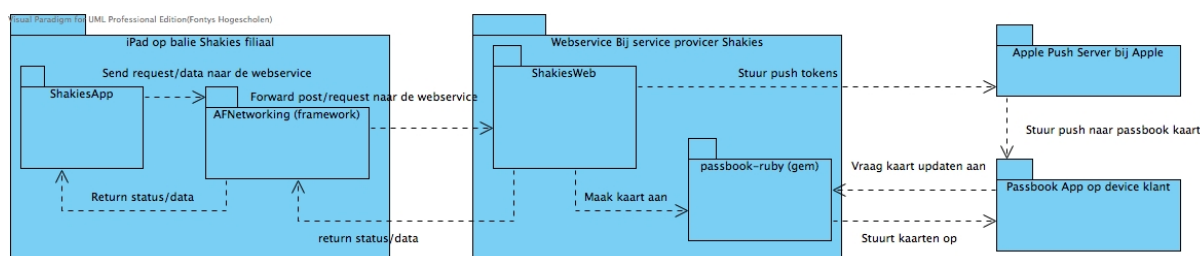
8.3 Sprint 3

Gedurende sprint 3 is er wat meer tijd gemaakt voor documentatie, terwijl de laatste onderdelen van de webservice en de medewerkers-applicatie worden afgemaakt.

8.3.1 Documentatie

Communicatie

Er is een communicatiediagram gemaakt om de klant duidelijk te maken welke applicaties en componenten met elkaar communiceren. Op figuur 7-2 is de communicatie te zien tussen de medewerkers-applicatie (links), de webservice (midden), de koffiekaart (rechts onder) en de 'Apple Push Notification Service' (rechts boven). Binnen elk blok zijn de verschillende componenten van de applicaties te zien, en welke informatie ze naar elkaar doorsturen.



Figuur 8-2 Communicatiediagram

Userstories

In het requirements document bevinden zich de userstories, hiervan zijn er twee volledig uitgewerkt en verbeterd waar nodig. De userstories zullen in sprint vier gebruikt worden gebruikt voor het opzetten van een aantal tests.

Tutorial

Er is er een tutorial gemaakt waarin wordt uitgelegd hoe alle benodigde certificaten/App-ID's en passtypes moeten worden aangemaakt. Ook wordt behandeld hoe al deze dingen samen komen in de Passbook-ruby gem en wat hierin moet gebeuren zodat de gem zonder problemen werkt.

Medewerkers-applicatie

Aan de uitwerking van de medewerkers applicatie -specifiek de GUI- zijn de shakies kleuren en plaatjes toegevoegd. Daarnaast zijn er een aantal verduidelijkingen toegevoegd waardoor het voor de medewerkers makkelijker wordt om de applicatie in gebruik te nemen. Op het hoofdscherm (Figuur 7-4) is een voorbeeld van een koffiekaart geplaatst en op het scan scherm (Figuur 7-5) is dezelfde kaart te zien met daarop een voorbeeld wat van de kaart binnen het vierkant moet vallen voor een optimale scan.

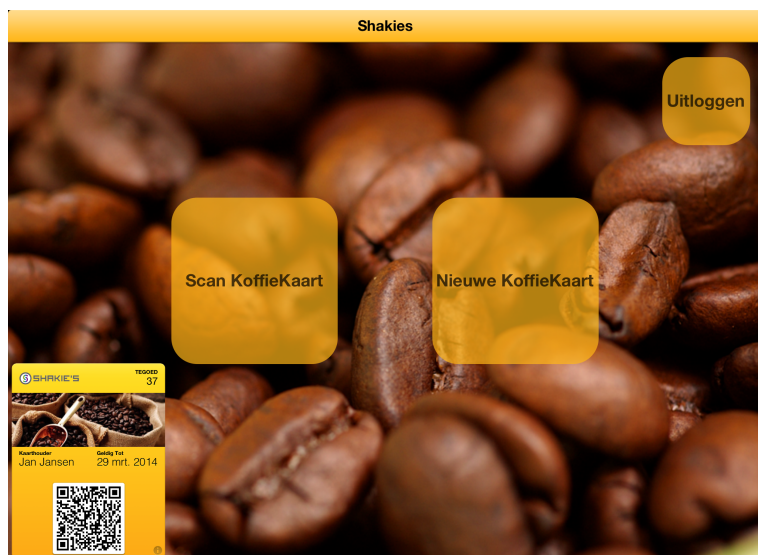
Ook is er een icon gemaakt voor de applicatie zodat deze herkenbaar is op het thuis scherm van de iPad (Figuur 7-3).

Na het laten testen van de applicatie door TASS-werknemers kwam naar voren dat de feedback die gebruiker kreeg duidelijker kon. Dit is opgelost door meer

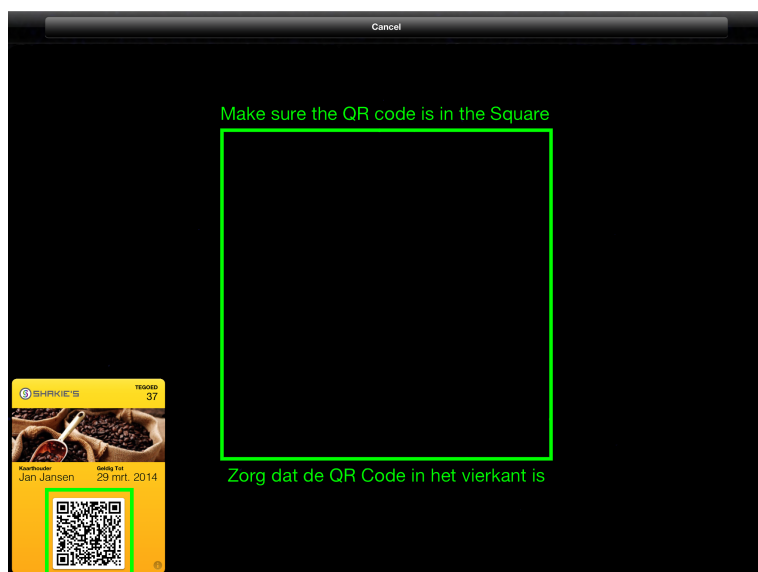


**Figuur 8-3
Applicatie Icon**

waarschuwingen te geven wanneer iets fout gaat en dit ook visueel door middel van kleuren duidelijk te maken op het scherm, waardoor direct duidelijk wordt waar het probleem zich heeft voorgedaan.



Figuur 8-4 Hoofdscherm medewerkers-applicatie



Figuur 8-5 QR Scanscherm

8.3.2 Webservice

De klant wilde dat de koffiekaartgegevens ook zichtbaar zijn buiten de iPad-applicatie. Hiervoor is daarom een simpele website gemaakt, met daarop een overzicht van alle koffiekaarten. Het is nu ook mogelijk om via de website de gegevens van klanten en tegoeden aan te passen.

8.4 Sprint 4

Sprint 4 zal gericht worden op het beveiligen en testen van de verschillende applicaties.

8.4.1 Beveiliging

Wat moet er beveiligd worden?

- Toegang tot de webservice
- Toegang tot de medewerkers applicatie API
- Toegang tot de web interface van de webservice

Uit het onderzoek is gebleken dat OAuth gebruikt gaat worden voor de beveiliging van de webservice, dit komt overeen met de wensen van de klant. Om dit te kunnen realiseren moeten er gebruikersaccounts zijn en moet er een OAuth provider worden opgezet. Er is wederom gezocht naar bruikbare gems. Deze zijn gevonden in Devise en Doorkeeper.

Devise

Devise is een veelgebruikte authenticatie-oplossing met ondersteuning voor veel belangrijke functies. Het versleutelen van wachtwoorden voor het opslaan in de database, de mogelijkheid tot het versturen van e-mails naar gebruikers voordat hun account geactiveerd wordt en het blokkeren van accounts na het te vaak te proberen in te loggen met foutieve gegevens zijn een aantal voorbeelden waarvan gebruik wordt gemaakt op de webservice. (Plataformatec)

Doorkeeper

Doorkeeper is een OAuth2.0 provider gebaseerd op versie 22 van de OAuth specificatie. Doormiddel van Doorkeeper kan er gemakkelijk de OAuth functionaliteit worden toegevoegd aan een applicatie en na enig configureren kan Doorkeeper gebruik maken van de gebruikers die zich bevinden binnen. Zit zorgt voor de authenticatie en Doorkeeper voor de toegangstokens voor de applicaties. (Doorkeeper)

Op de website wordt alleen gebruik gemaakt van Devise omdat de gebruiker direct op de server werkt. Wanneer OAuth hier gebruikt zou worden voegt dit weinig toe aan de beveiliging. De medewerkers-applicatie moet daarin tegen gebruik maken van OAuth. De medewerker moet in de applicatie zijn gebruikersnaam en wachtwoord invullen en op inloggen klikken. Dan worden de gegevens opgestuurd naar de webservice, waar Doorkeeper dit zal doorgeven aan Devise. Wanneer Devise aangeeft dat de gebruiker met succes geauthentiseerd is, zal Doorkeeper een antwoord terugsturen naar de medewerkers-applicatie. In dit antwoord zit een toegangstoken, een ververs token en een tijd in seconden. Wanneer de tijd verlopen is zal de toegangstoken niet langer geldig zijn en moet er gebruik worden gemaakt van de het ververs token om opnieuw toegang te verkrijgen. Wanneer een gebruiker zelf uitlogt of langere tijd inactief is, zal het hele proces opnieuw worden doorlopen.

HTTPS

Er zal nog een HTTPS-certificaat worden aangevraagd, maar doordat dit niet kan op een interne domeinnaam moet er overlegd worden met de IT-afdeling hoe dit opgelost kan worden. Dan zal de volgende sprint plaats vinden, waarna het in sprint 5 of 6 geïmplementeerd kan worden binnen de applicaties.

8.4.2 Calabash – Tests

Calabash is een Behaviour Driven Development (BDD) test library en is gebaseerd op Cucumber. De tests worden geschreven in Gherkin, een taal die iedereen eenvoudig kan lezen. Doordat er tijdens sprint 0 al userstories gemaakt zijn, kunnen deze makkelijk worden omgezet in tests voor Calabash. Een test (feature) bestaat uit meerdere scenario's. Hieronder een voorbeeld van het login scenario:

Scenario: Login

Given I am on the Welcome Screen
Then I wait to see a navigation bar titled "Inloggen"
And take picture
Then I enter "matthijs.van.driel@tass.nl" into the "Gebruikersnaam" input field
Then I enter "geheim123" into the "Wachtwoord" input field
And take picture
Then I touch the "Inloggen" button
Then I wait to see a navigation bar titled "Shakies"

Elke regel is een stap in de test die vertelt wat er moet gebeuren. Bijvoorbeeld de 2d^e regel van het login scenario is het wachten tot het login scherm verschijnt, hierna wordt een screenshot genomen. De screenshots worden gemaakt zodat wanneer alle tests afgelopen zijn, geverifieerd kan worden of alles op het scherm is zoals het moet zijn.

Door het uitvoeren van alle tests wordt de complete flow van de applicatie getest, van het invullen van gegevens op de iPad tot een antwoord krijgen van de webservice. Wanneer een onderdeel zijn werk niet goed doet dan zal de test falen. (LessPainful) (Krukow, Stu, & Harmon)

8.4.3 Companion app

Gedurende de sprint werd steeds duidelijker dat alles eerder klaar zou zijn dan verwacht. Dat er tijd over zou zijn is besproken met de opdrachtgever. Toen is het idee geopperd om naast de medewerkers-applicatie een tweede iOS-applicatie te maken. Binnen de applicatie zou de klant zijn koffiekaarten moeten kunnen zien en deze kunnen opwaarderen zonder langs te hoeven gaan bij een Shakies filiaal. Wanneer een klant nog geen koffiekaart heeft, kan hij of zij er een aanschaffen binnen deze applicatie. De app wordt ook wel een companion app genoemd, omdat hij de functionaliteit van de koffiekaart uitbreidt. Er zal tijdens de volgende sprint worden onderzocht of dit mogelijk is, en wat hiervoor de benodigheden zijn.

8.5 Sprint 5

Sprint 5 richt zich op het verder documenteren en testen van de applicatie, daarnaast wordt er gekeken naar de companion app.

8.5.1 Documentatie

De documentatie wordt in deze sprint gericht op Devise, Doorkeeper en AFNetworking. Dit zijn frameworks en gems die gebruikt worden. In de documentatie wordt kort uitgelegd hoe de frameworks werken en hoe ze geconfigureerd moeten worden.

Over Devise is te lezen hoe de gem geïnstalleerd moet worden en welke functies er wel en niet gebruikt worden. Ook wordt er uitgelegd hoe de administrator functionaliteit wordt toegevoegd. Door het gebruik van een administratorrol kunnen gewone gebruikers geen gegevens verwijderen uit de applicatie, terwijl de administrator dit wel kan.

Bij Doorkeeper moet gebruik worden gemaakt van de gebruikersaccounts binnen Devise. Hiervoor moet er specifiek worden verwezen naar Devise voor de gebruikersauthenticatie.

AFNetworking is een netwerk framework voor iOS en Mac OSX. Door het gebruik van het framework wordt er een veel werk uit handen genomen. Hierdoor kunnen applicaties sneller ontwikkeld worden zonder veel zorgen over netwerkverbindingen. Er wordt ook gebruik gemaakt van AFOAuth2Client, een plug-in voor AFNetworking die OAuth functionaliteit toevoegt. Hierdoor kan eenvoudig elk verzoek worden voorzien van een toegangstoken.

8.5.2 Calabash Tests

In de vorige sprint is een begin gemaakt met Calabash, de tests zijn in deze sprint uitgebreid. Er worden nu ook foute gegevens ingevuld waarna gecontroleerd wordt of de gebruiker een foutmelding krijgt. Wanneer dit het geval is, is de test geslaagd.

De tests worden voornamelijk gedaan op de iOS iPad simulator, zodat de tests sneller klaar zijn. Wanneer dit op een echte iPad gedaan wordt duurt de test 3,5 keer langer dan op de simulator. Dit heeft vooral te maken met de vertraging over het netwerk.

8.5.3 Companion App

Om binnen de companion app de koffiekaart uit te lezen moet gebruik worden gemaakt van het PassKit framework. Door middel van dit framework kan worden gecontroleerd of de Passbook-applicatie beschikt over een koffiekaart. Wanneer dit zo is, kan de koffiekaart worden uitgelezen.

Voor het aankopen en opwaarderen van de koffiekaart moet betaald worden, net zoals bij een Shakies filiaal. Om de betaling te kunnen verwerken is gekozen voor in App-aankopen van de Apple app store. Er is hiervoor gekozen omdat TASS ook hier meer over wilde weten.

8.6 Sprint 6

Tijdens sprint 6 is een begin gemaakt aan de companion app. Daarnaast is de beveiliging verbeterd door het gebruik van een ssl-certificaat.

8.6.1 Compantion App

Dankzij het PassKit framework is het door Apple erg eenvoudig gemaakt om de gegevens uit de koffiekaart te halen en deze weer te geven in de app.

Voor de in App-aankopen moet meer gebeuren. Producten moeten worden aangemaakt in iTunes connect. Dit is de beheertool voor ontwikkelaars, waarin al hun producten worden aangemaakt.

Voor het aanmaken van het product binnen iTunes connect moeten allerlei gegevens over het product worden ingevuld, zoals de prijs en de omschrijving in elke taal waarin de app beschikbaar wordt. Vervolgens kan binnen de iOS-applicatie de gegevens van het product worden opgevraagd door middel van het StoreKit framework.

Nadat de code hiervoor gemaakt was, werd deze getest. Bij het testen bleek dat de App store het gebruikte product-ID terugstuurde als niet geldig. Na enig uitzoekwerk bleek dat kwam doordat TASS nog niet akkoord was gegaan met het 'iOS paid applications' contract.

Dit is doorgestuurd naar de personen binnen TASS die hiermee akkoord moeten gaan en wordt momenteel uitgezocht. Volgens de planning wordt hier in sprint 7 verder mee gegaan, Mocht het dan nog niet geregeld zijn of zou TASS besluiten het contract niet af te sluiten, dan moet er met de opdrachtgever worden overlegd wat de volgende stap is.

8.6.2 HTTPS

Omdat de server waarop de webservice draait een interne server is was er een probleem met het aanvragen van het SSL-certificaat. Deze kan niet worden aangemaakt voor een interne domein naam. Na overleg met de IT-afdeling is besloten om een subdomein van een externe domeinnaam te laten verwijzen naar het interne IP-adres van de server. Hierdoor kan er binnen kantoor worden getest met een geldig certificaat zonder dat de server beschikbaar is vanaf het internet.

De koffiekaart maakt nu verbinding over een SSL-verbinding, waardoor de koffiekaart worden geïnstalleerd op een iPhone of iPod touch die niet de ontwikkelingsopties aan heeft staan. Dit is dan ook uitgeprobeerd met een aantal iPhone's van TASS-medewerkers. Uit de test bleek dat alle functies naar behoren werkten.

Hierna is ook de medewerkers-applicatie aangepast zodat deze ook al het verkeer verstuurd over HTTPS, waardoor de beveiliging weer is toegenomen.



Figuur 8-3 Hoofdschem companion app

8.7 Sprint 7

Sprint 7 is op dit moment nog niet begonnen, maar zal gericht worden op het verbeteren van de GUI van de webservice en het verder werken van de companion app.

8.7.1 GUI Webservice

De GUI van de webservice is erg basic (Figuur 7-4). De bedoeling is om deze gebruikersvriendelijker en overzichtelijker te maken. Dit wordt mede gerealiseerd door het gebruik van Bootstrap, een framework waardoor een website goed werkt op verschillende schermformaten.

Listing customers

Logged in as **matthijs** @tass.nl [Edit profile](#) | [Logout](#)
[User Admin](#)

Signed in successfully.

Card ID	Firstname	Lastname	Email	Balance	Created	Updated			
202				20	2013-05-30 08:24:02 UTC	2013-05-30 08:24:02 UTC	Show	Edit	Destroy
203				19	2013-05-30 08:24:54 UTC	2013-05-30 08:27:01 UTC	Show	Edit	Destroy
204				18	2013-05-30 12:01:48 UTC	2013-05-30 12:03:39 UTC	Show	Edit	Destroy
205				15	2013-05-30 12:58:07 UTC	2013-05-30 13:00:14 UTC	Show	Edit	Destroy
201				999999	2013-05-29 13:54:49 UTC	2013-05-30 14:21:34 UTC	Show	Edit	Destroy
199				20	2013-05-28 20:27:33 UTC	2013-06-02 15:23:09 UTC	Show	Edit	Destroy
112				7	2013-04-18 15:20:36 UTC	2013-05-27 08:19:06 UTC	Show	Edit	Destroy
195				15	2013-05-27 09:08:33 UTC	2013-05-27 09:11:58 UTC	Show	Edit	Destroy
200				13	2013-05-29 13:51:21 UTC	2013-05-29 13:53:23 UTC	Show	Edit	Destroy

[New Customer](#)

Figuur 8-4 Huidige GUI webservice (gegevens zijn afgeschermd)

8.7.2 Compantion App

Als TASS akkoord gaat met de contracten van Apple, zal het werk aan de compantion app worden voorgezet. Het zal dan mogelijk worden voor klanten om direct hun koffiekaart te kunnen opwaarderen en voor nieuwe klanten om een nieuwe koffiekaart te kunnen aanschaffen vanaf hun smartphone.

Wanneer TASS besluit niet akkoord te gaan met de contracten zal overlegt worden met de opdrachtgever wat de andere opties zijn.

9 Invoeringsfase

De invoeringsfase zal plaatsvinden in Sprint 8. Deze zal in tegenstelling tot alle andere sprints maar één week duren. Gedurende deze sprint zal al het gemaakte werk worden overgedragen aan TASS. Waar nodig zal nog extra documentatie worden gemaakt of de bestaande documentatie zal verduidelijkt worden, zodat iemand anders gemakkelijk hiermee verder kan gaan. Dit zal gebeuren in overleg met de opdrachtgever. Verder zal nog niet gecommenterende code voorzien worden van commentaar.

Wanneer de opdrachtgever met alles akkoord is zal het klant acceptatie formulier worden ingevuld.

Het product zal worden gebruikt als demonstratie opstelling voor passbook. Hiermee kan TASS laten zien wat de mogelijkheden zijn van passbook aan klanten. Er zijn nog ideeën om passbook te gebruiken bij #devdate (een evenement voor software ontwikkelaars dat word georganiseerd door TASS), maar wanneer dit gebeurd en of het gaat gebeuren is nog niet bekend.

10 Conclusie(s) en aanbeveling(en)

Gedurende dit project is een complete Passbook oplossing gemaakt. Dit is gedaan om te zien wat alle mogelijkheden van Passbook zijn zodat TASS –de opdrachtgever– kan beoordelen of het gebruikt kan worden in toekomstige projecten.

Met dit project is bewezen dat Passbook een goed alternatief is voor de papieren koffiekaart. Hij is makkelijk in gebruik en biedt een aantal voordelen over de papieren versie. Zo kunnen klanten hun kaart niet kwijtraken. De kaart kan worden opgeladen en kan op meerdere apparaten geïnstalleerd worden. Hierdoor heeft de klant zijn koffiekaart altijd bij zich.

Om uit te vinden wat een goede en werkbare beveiliging is voor het project is een onderzoek gedaan. De hoofdvraag uit het onderzoek luidt: *“Hoe kan de toegang en verbinding met de Shakes webservice het beste beveiligd worden zodat alleen gemachtigde personen en kaarten toegang krijgen?”*

Deze vraag is beantwoord door het beantwoorden van de deelvragen en het uitvoeren van bureauonderzoek. Hieruit is gebleken dat elke beveiligingstechniek zijn voordelen en nadelen heeft. Voor de communicatie van de eigen applicaties met de webservice is in overleg met de opdrachtgever gekozen voor OAuth2.0 in combinatie met SSL/TLS. Waarbij OAuth zorgt voor de autorisatie van de gebruikers bij de webservice en SSL/TLS wordt gebruikt om de verbinding en communicatie tussen de verschillende applicaties te beveiligen. Beide technieken bewezen zijn op hun veiligheid en worden daarom ook veel gebruikt. Ook heeft TASS hier al ruime ervaring mee waardoor toekomstige ontwikkelaars met het product verder kunnen werken.

Voor de beveiliging van de koffiekaart heeft Apple al besloten, hieraan kon daarom niks worden veranderd of worden toegevoegd. Was dit wel gedaan, dan zou de koffiekaart geen toegang meer kunnen krijgen tot de webservice of zou de Passbook-applicatie weigeren verbinding te maken.

Om het scrumproces bij te houden is gebruik gemaakt van Scrumwise. Scrumwise mist functionaliteit en dit maakt het moeilijk om er optimaal gebruik van te maken. Ook kan er niks achteraf worden aangepast wanneer er een fout is gemaakt. Ik zou daarom TASS willen aanraden scrumwise niet te gebruiken voor toekomstige projecten.

TASS kan in de toekomst de bestaande passbook oplossing nog verder uitbreiden met onder andere een uitgebreider beheer systeem en meer type kaarten. Hiermee wordt het product bruikbaar voor een groter publiek. Daarnaast kan TASS het product verkopen aan verschillende organisaties die een klantenkaart, kortingsbonen en dergelijken uitgeven.

Evaluatie

Tijdens mijn afstudeerperiode heb ik het erg naar mijn zin gehad, alle studenten helpen elkaar als er om wordt gevraagd en er was een goede samenwerking. Ook was het geen probleem om bij andere medewerkers aan te koppen voor hulp wanneer dit nodig is. Dit alles draagt bij aan een leuke afstudeerperiode.

De afstudeeropdracht die ik heb gekozen is mij goed bevallen en hierdoor heb ik een aantal nieuwe dingen geleerd, zoals de programmeertaal Ruby en het Rails framework. Dit was in het begin een uitdaging, maar door het volgen van verschillende tutorial's en dingen opzoeken op internet is dit goed verlopen. Binnen Ruby on Rails is nog veel meer te leren dan dat nu aanbod is gekomen maar ik heb nu wel een goede basis om op verder te bouwen. Ook heb ik me weer verder verdiept in Objectieve-C met de iOS Software Development Kit (SDK) door het gebruik van Passbook.

Door de kennis die opgedaan had met Ruby on Rails kwamen verschillende medewerkers langs om hierover vragen te stellen. Dit omdat ze hiervan nog geen kennis hadden, en voor hun een aantal dingen niet duidelijk waren. Deze heb ik toen uitgelegd.

Van het onderzoek heb ik ook een veel geleerd, ik weet nu beter wat de voor,- en nadelen zijn van de verschillende beveiligingstechnieken en hoe een aantal hiervan moeten worden toegepast. Zoals OAuth, dit heb ik gebruikt bij de communicatie tussen de medewerkers-applicatie en de webservice.

Tijdens sprint 4 werk duidelijk dat het project eerder klaar zou zijn dan oorspronkelijke gepland. Ik heb toen met de opdrachtgever gesproken wat er nog meer bij gemaakt zou kunnen worden. Dit is de companion app geworden. Op het moment van schrijven is nog niet duidelijk of deze afgemaakt kan worden omdat TASS nog akkoord moet gaan met de contracten van Apple. Ik hoop dat dit op tijd gebeurt zodat ik het nog kan afronden.

Het project is goed verlopen, er hebben zich verder geen problemen voorgedaan. Dit is mede te danken aan het goed methodisch handelen doormiddel van het tien stappen plan in combinatie met scrum. Scrum vind ik een goede manier van werken, het proces blijft veel overzichtelijker dan wanneer de waterval methode gebruikt wordt omdat er in korte sprints gewerkt wordt. Na elke sprint vind er een retrospectieve plaats waarin duidelijk word wat er goed ging, wat minder en wat er verbeterd kan worden.

Literatuurlijst

Apple Inc. (sd). *Passbook Programming Guide: About Passbook*. Opgeroepen op 3 10, 2013, van Apple Developer: https://developer.apple.com/library/ios/#documentation/UserExperience/Conceptual/PassKit_PG/Chapters/Introduction.html

Apple Inc. (sd). *Xcode 4 Downloads and Resources - Apple Developer*. Opgehaald van Apple Developer: <https://developer.apple.com/xcode/>

Brown, J. (2011, 5 15). *ZBar bar code reader*. Opgeroepen op 4 16, 2013, van ZBar bar code reader: <http://zbar.sourceforge.net/>

Doorkeeper. (sd). *Doorkeeper OAuth 2 Provider*. Opgehaald van Doorkeeper OAuth 2 Provider: <https://doorkeeper-provider.herokuapp.com/>

Jetbrains. (sd). *Ruby on Rails IDE :: JetBrains RubyMine*. Opgehaald van JetBrains: <http://www.jetbrains.com/ruby/>

Krukow, K., Stu, & Harmon, T. (sd). Opgeroepen op 4 20, 2013, van <https://github.com/calabash/calabash-ios>

LessPainful. (sd). *Calaba.sh - Automated Acceptance Testing for iOS and Android Apps*. Opgeroepen op 3 25, 2013, van Calaba.sh: <http://calaba.sh/>

Plataformatec. (sd). *plataformatec/devise · GitHub*. Opgeroepen op 4 20, 2013, van Github: <https://github.com/plataformatec/devise>

Scrumwise. (sd). *Scrum tools | Scrumwise - The Scrum tool you've always wanted*. Opgehaald van Scrumwise: <https://www.scrumwise.com/>

Scrumwise. (2013, 5 2). *Scrumwise*. Opgeroepen op 5 2, 2013, van Scrumwise: <https://www.scrumwise.com/scrum/>

TASS B.V. . (sd). *Afstuderen - TASS NL*. Opgeroepen op 4 5, 2013, van TASS NL: http://www.tass.nl/media/220043/tass_afstudeeropdrachtenboek_jan2013-jun2013.pdf

TASS. (2013, 3 28). *Intranet TASS - Organisatie*. Opgeroepen op 4 5, 2013, van Intranet TASS: <https://www.onderwegmettass.nl/Subwebs/Organisation/main.php>

TASS. (2013, 2 1). *Sensei*. Opgeroepen op 3 1, 2013, van Sensei: <https://trac.tassprojects.nl/sensei>

TASS. (n.d., n.d. n.d.). *TASS - TASS NL*. Opgeroepen op 3 10, 2013, van TASS: <http://tass.nl/nl-nl/over-tass/tass/>

Visual Paradigm. (sd). *UML CASE tool for software development*. Opgehaald van Visual Paradigm: <http://www.visual-paradigm.com/product/vpum/>

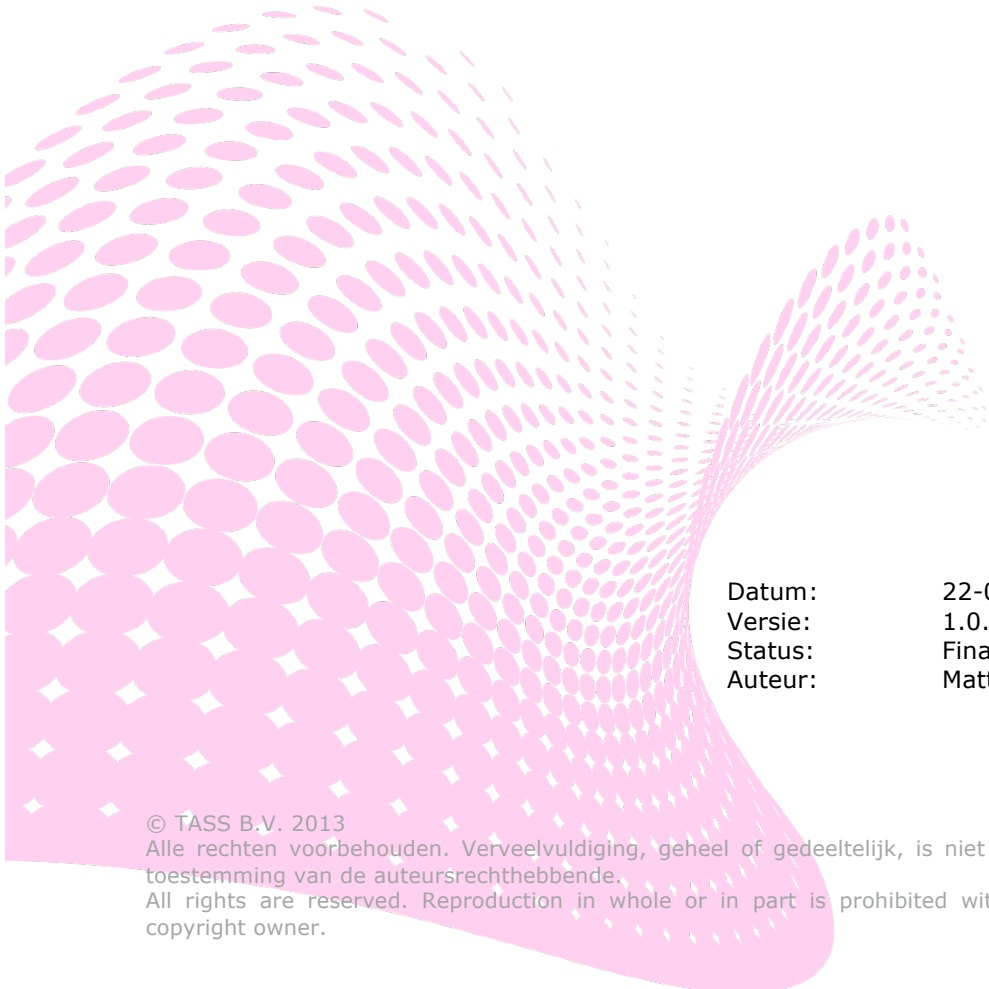
Wikipedia. (2013, 4 23). *Application programming interface - Wikipedia*. Opgeroepen op 3 9, 2013, van Wikipedia: http://nl.wikipedia.org/wiki/Application_programming_interface

Zennaware. (sd). *Zennaware » Home of Cornerstone Subversion Client for Mac OS X*. Opgeroepen op 3 1, 2013, van Cornerstone 2: <http://www.zennaware.com/cornerstone/index.php>

Bijlage I: Project Management Plan (PMP)

Project Management Plan

Shakies passbook koffiekaart



Datum:	22-03-2013
Versie:	1.0.0
Status:	Final
Auteur:	Matthijs van Driel

© TASS B.V. 2013

Alle rechten voorbehouden. Verveelvuldiging, geheel of gedeeltelijk, is niet toegestaan dan met schriftelijke toestemming van de auteursrechthebbende.

All rights are reserved. Reproduction in whole or in part is prohibited without the written consent of the copyright owner.

Dit document is gepubliceerd door:

TASS B.V.
Eindhoven, Nederland

Commentaar en suggesties kunnen worden gestuurd naar:

TASS B.V.
Postbus 80060
5600 KA EINDHOVEN
Nederland
tel: +31 (0)40 2503200
fax: +31 (0)40 2503201

1	Introductie	1
1.1	Doel	1
1.2	Afkortingen en Definities	1
1.3	Distributielijst	1
1.4	Document Geschiedenis	1
2	Project Beschrijving	2
2.1	Doel	2
2.2	Scope	2
2.2.1	Medewerker App	2
2.2.2	Webservice	2
2.2.3	KoffieKaart	2
2.2.4	Companion app	2
2.2.5	Onderzoek	3
2.3	Fasering	3
2.3.1	Oriënterende Fase	3
2.3.2	Ontwikkelings- en oplossingsfase	3
2.3.3	Invoeringsfase/Afrondingsfase	4
2.4	Milestones	4
2.5	Werkproducten / Releases	4
2.6	Budget	5
2.7	Metrics	5
2.8	Rapportage	5
3	Organisatie	6
3.1	Organisatieschema	6
3.2	Stuurgroep	6
3.3	Project Team	7
3.4	Communicatie en Externe Interfaces	7
4	Randvoorwaarden	8
4.1	Aannames en Afhankelijkheden	8
4.2	Standaarden	8
4.3	Apparatuur, Infrastructuur en Tools	8
4.4	Risico's	9
5	Software Test Plan	10
5.1	Scope	10
5.1.1	Unittests	10
5.1.2	Systeemtest	10
5.1.3	Acceptatietest	10
5.2	Aanpak	10
5.2.1	Test suite	10
5.3	Testomgeving	10
5.4	Testrapportage	10
6	Configuratie Management Plan	11
6.1	Organisatie	11

6.2 Configuratie Beheer	11
6.3 Change Management	11
7 Kwaliteit	12
7.1 Proces	12
7.2 Reviews	12
7.3 Testrapportages	12

1 Introductie

1.1 Doel

Dit plan beschrijft:

- Hoe het project is georganiseerd.
- Wat het globale schedule is en welke werkproducten wanneer worden geleverd.
- Op welke aannames en randvoorwaarden het plan is gebaseerd.
- Hoe de kwaliteit wordt gewaarborgd.
- Afspraken over acceptatie en onderhoud.

1.2 Afkortingen en Definities

SVN Subversion

PMP Project Management Plan

1.3 Distributielijst

Naam	Organisatie	Functie
Josee Zielman	TASS Technology Solutions	People Manager
Rob Knubben	TASS Technology Solutions	Technisch begeleider
Rene M.J.W. van der Heijden	Fontys Hogeschool ICT	Docent begeleider
Matthijs van Driel	TASS Technology Solutions	Afstudeerder

1.4 Document Geschiedenis

Versie	Status	Datum	Auteur	Wijzigingen
0.1.0	Draft	14-02-2013	Matthijs van Driel	- Initiële versie
0.2.0	Concept	20-02-2013	Matthijs van Driel	- Feedback bedrijfsbegeleider verwerkt - Concept versie
0.3.0	Concept	22-02-2013	Matthijs van Driel	- Feedback docentbegeleider verwerkt

2 Project Beschrijving

2.1 Doel

Passbook is een nieuwe manier om instapkaarten, klantenkaarten, kortingsbonnen, bioscoopkaartjes ect bij je te hebben. Passbook is geïntroduceerd met de nieuwe iOS versie 6.0 en is beschikbaar op de iPhone en iPod touch. Er zijn al een hoop bedrijven die gebruik maken van deze functionaliteit, dit gaat van vliegmaatschappijen met vliegtickets tot kortingsbonnen bij de Dixons.

TASS is geïnteresseerd in de functionaliteiten en werking van Passbook. Hoe werkt het, en wat is er mogelijk?

Ook wil TASS meer ervaring opdoen het met maken van applicaties voor iOS want er is op dit gebied weinig ervaring. Daarnaast moet er een werkende applicatie worden opgeleverd.

TASS zelf is de klant van deze opdracht en het zal intern uitgevoerd worden. Het resultaat wordt gebruikt als indicatie of de passbook functionaliteit gebruikt kan worden in toekomstige projecten.

2.2 Scope

De volgende applicaties worden gerealiseerd tijdens het project:

2.2.1 Medewerker App

De applicatie voor de shakies medewerker wordt gemaakt voor iOS6 voor gebruik op een iPad. Deze applicatie zal passbook koffiekaart kunnen scannen, en kunnen communiceren met de webservice voor het aanmaken/beheren/verwijderen van koffiekaarten.

2.2.2 Webservice

De webservice wordt gemaakt in Ruby on rails. Ook wordt er gebruikt gemaakt van het TASS Chimera framework (mits er een werkbare versie beschikbaar is). Dit framework is/word ontwikkelt zodat er niet bij ieder project het wiel opnieuw hoeft worden uitgevonden.

Als er blijkt dat het gebruik van het chimera framework te veel tijd kost, kan er in overleg worden afgesproken om het niet te gebruiken.

De webservice zal verantwoordelijk worden voor de connectie met de database, medewerkers app, en de koffiekaart.

2.2.3 KoffieKaart

De passbook koffiekaart bestaat uit een speciaal zip bestand, deze wordt gegenereerd door de webservice. Wanneer deze binnen de passbook applicatie op de telefoon is toegevoegd zal de koffiekaart contact kunnen zoeken met de webservice zodat deze geüpdatet kan worden.

Op de koffieKaart zal informatie over de vervaldatum, saldo, eigenaar te zien zijn.

2.2.4 Companion app

De companion app bestaat uit een iPhone/iPod touch applicatie. Binnen deze applicatie zal de informatie van de passbook kaart te vinden zijn, en kunnen klanten tegoed aan hun koffie kaart(en) toevoegen. Ook zal er een nieuwe koffiekaart kunnen worde aan geschaft.

2.2.5 Onderzoek

Om de webservice te beveiligen zal er een onderzoek gedaan worden naar de verschillende beveiligingsmogelijkheden.

Hoofdvraag van het onderzoek:

Hoe kan de toegang en verbinding met de Shakies webservice het beste beveiligd worden zodat alleen gemachtigde personen en kaarten toegang krijgen?

Deelvragen:

- Welke beveiligingsmogelijkheden zijn er beschikbaar binnen Ruby on Rails?
- Aan welke mogelijkheden zijn kosten verbonden en hoe hoog zijn deze?
- Welke eisen stelt Apple aan de beveiliging van de passbook kaarten?
- Met welke mogelijkheid/mogelijkheden is er al ervaring binnen TASS?

Methode

De meeste deelvragen zullen beantwoord worden door het doen van bureau onderzoek. Om uit te vinden wat er binnen TASS gebruikt wordt en waarom, zal een interview gedaan worden.

Meer informatie over het onderzoek en de voorbereiding hierop is te vinden in het onderzoek voorbereidingsdocument.

2.3 Fasering

2.3.1 Oriënterende Fase

In deze fase wordt de opdracht uitgebreid onderzocht en kennis gemaakt met het bedrijf. Met deze informatie wordt het project management plan (PMP) opgesteld.

Na het opstellen van het PMP wordt er begonnen aan het onderzoek van de verschillende beveiligingsmogelijkheden.

Resultaten:

- Project management plan
- Onderzoeksrapport
- Backlog

2.3.2 Ontwikkelings- en oplossingsfase

Deze fase wordt uitgevoerd met behulp van scrum. Dit betekent dat er elke 2 weken een sprint zal plaatsvinden, waarin een user story(s) wordt gerealiseerd en getest.

Aan het begin wordt er gezorgd dat de userstories goed zijn ingevuld in de backlog, met daarin de benodigde taken.

Aan het einde van een sprint wordt het sprintrapport opgesteld, uitkomsten van de tests gedocumenteerd, benodigde documentatie gemaakt en wordt er een demo gegeven van het gerealiseerde werk.

Resultaten:

- Prototypes
- Demo
- Sprintrapport
- Test plan
- Acceptatie test
- Backlog
- Documentatie

2.3.3 Invoeringsfase/Afrondingsfase

In deze fase word alles afgrond en word het project overgedragen aan TASS.

Resultaten:

- Scriptie
- Demo
- Presentatie

2.4 Milestones

Onderstaande tabel geeft een overzicht van de project milestones zoals afgesproken met de stuurgroep:

Milestone	Datum
Start Project	04-02-2013
PMP	08-03-2013
Onderzoek	08-03-2013
KoffieKaart	22-03-2013
Webservice Passbook	26-04-2013
Webservice Medewerker Applicatie	10-06-2013
Medewerker Applicatie	10-06-2013
Afronding realisatie	10-06-2013
Project Sluiting	21-06-2013

2.5 Werkproducten / Releases

Release Type bepaald de flow in het product Release en Product Acceptatie proces.

- Een Formele Levering: klant/technisch begeleider Product Acceptatie.
- Een Tussentijdse Levering: vereist onderlinge afspraak team – klant.

Onderstaande tabel geeft een overzicht de werkproducten die aan de klant worden geleverd:

ReleaseType	Werkproduct / Release	Datum
Tussentijds	Project Management Plan	04-03-2013
Tussentijds	Onderzoeksrapport	04-03-2013
Tussentijds	Backlog	15-03-2013
Tussentijds	Test cases	
Tussentijds	Passbook kaart	15-03-2013
Tussentijds	Webservice: Genereren passbook kaart	29-03-2013
Tussentijds	Webservice: Communicatie passbook kaart (updates, push notifications)	26-04-2013

Tussentijds	iOS Medewerker applicatie – Scannen passbook kaart	03-05-2013
Tussentijds	Webservice: Api voor Medewerkers applicatie	10-06-2013
Tussentijds	iOS Medewerker applicatie – Communicatie webservice	10-06-2013
Tussentijds	Tussentijdse release aan het einde van een sprint	Elke 2 weken
Formeel	Oplevering	21-05-2013

2.6 Budget

De totale tijdsduur van het project is vijf maanden. Voor het project zijn er 85 dagen / 680 uur beschikbaar.

Wanneer er kosten gemaakt moeten worden om iets aan te schaffen zal dit met de projectbegeleider worden overlegd.

2.7 Metrics

Tijdens het project zullen de volgende metrics worden verzameld:

Metric	Definitie	Frequentie
Milestone status	Zie evaluatie rapport template	Periodiek
Effort	Zie evaluatie rapport template	Wekelijks
Aantal risico's	Zie evaluatie rapport template	Wekelijks
Test Cases geïmplementeerd	Zie evaluatie rapport template	Wekelijks
Test Cases passed	Zie evaluatie rapport template	Wekelijks
Sprint rapportage	Document met afgeronde users story's, sprint evaluatie van de afgelopen sprint	Tweewekelijks
Burndownchart	Een grafische representatie van het werk uitgezet tegen de tijd	Tweewekelijks

2.8 Rapportage

De voortgang van het project zal op wekelijkse basis aan de projectbegeleider worden gerapporteerd via de project rapportage.

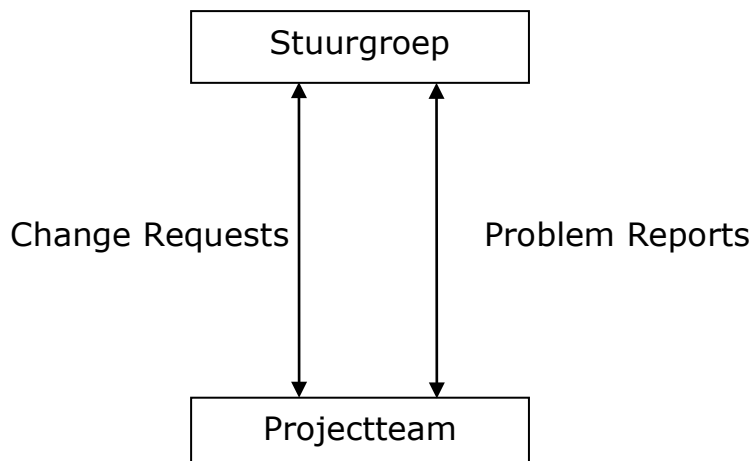
Daarnaast zal tweewekelijks de voortgang van het project worden besproken met de stuurgroep

De volgende rapporten zullen tijdens het project worden gemaakt:

- Onderzoeksrapport
- Sprint rapport
- Test rapport van de acceptatie test

3 Organisatie

3.1 Organisatieschema



3.2 Stuurgroep

De stuurgroep bestaat uit de volgende personen:

Rol	Naam	Functie in het project
Voorzitter	Josee Zielman	Proces coach, Klant
Lid	Matthijs van Driel	
Technisch begeleider	Rob Knubben	
Vergaderfrequentie: Elke 2 weken, Voorzitter en leden wekelijks		

De stuurgroep is verantwoordelijk voor:

- Periodieke voortgangscntrole in termen van tijd, geld en functionaliteit.
- Beslissingen over wijzigingen die de software requirements specificatie of het project management plan beïnvloeden.
- Organisatorische zaken met betrekking tot resources, communicatie, milestones, leveringen, etc.
- Prioriteitsstelling binnen het project.
- Evaluatie van de resultaten van milestone checks, audits en de project evaluatie.

3.3 Project Team

Naam	Periode	Niveau
Matthijs van Driel	4-02-2013 – 21-06-2013	HBO

3.4 Communicatie en Externe Interfaces

Wekelijks vind er een voortgangsgesprek plaats met de Project begeleider en tweewekelijks met de project begeleider en de technische begeleider.
Intern wordt met behulp van Scrum gecommuniceerd. Er zullen geen externe interfaces zijn.

4 Randvoorwaarden

4.1 Aannames en Afhankelijkheden

Binnen dit project zijn er enkele aannames gedaan.

- De afstudeerder kan zijn gehele werktijd besteden aan het project
- De afstudeerder heeft begeleiding vanuit het bedrijf en school
- De benodigde software en hardware zijn aanwezig voor het uitvoeren van het project

Tijdens de realisatie periode van het project:

- Toegang tot ontwikkelaars portal van Apple
- SDK beschikbaar voor iOS om 2D barcodes te scannen

4.2 Standaarden

De realisatie vindt plaats volgende de Agile Scrum richtlijnen van Sensei die op het intranet van TASS gevonden kunnen worden.

Daarnaast word gewerkt volgens de IOS 9001 standaard.

Bij tijdens het project zullen de volgende richtlijnen worden aangehouden.

- Tijdens het programmeren word gebruikt gemaakt van camcel case en snake case.
- Voor het maken van documenten word gebruikt gemaakt van de TASS templates waar mogelijk.

4.3 Apparatuur, Infrastructuur en Tools

Naam	Aantal	Toelichting
Computer	1x	PC om op te werken
SVN server	1x	SVN server beheerd door TASS
Ruby on Rails server	1x	Deze zou tijdens ontwikkeling kunnen draaien op de pc.
Tablet (iPad)	1x	
Mac computer	1x	Apple van computer
Xcode met iOS SDK	1x	Hiervoor zal een mac computer nodig zijn.

4.4 Risico's

Risico	Actie	Kans	Impact
Langdurige ziekte / afwezigheid	In overleg met de klant wordt er naar de planning gekeken, er worden nieuwe afspraken gemaakt over het product.	Laag	Hoog
Veranderende eisen	Aan de hand van de impact van deze verandering zal worden beslist welke acties worden genomen.	Gemiddeld	Hoog
Deadline van een taak of back log item niet halen	De niet gehaalde taak zal moeten worden ingehaald, wanneer er een grootte taak niet gehaald word zal er moeten worden overlegt meet de stuurgroep en klant.	Gemiddeld	Gemiddeld / Hoog
Hardware defect	Hardware zal moeten worden vervangen	Gemiddeld	Laag

5 Software Test Plan

5.1 Scope

Het testen heeft betrekking op de werking en functionaliteit van de "koffiekaart", webservice en medewerkers applicatie. Het systeem zal zowel getest moeten worden op unit-, integratie- en systeemniveau. Deze tests dienen om de werking van het systeem te testen, zo kan gecontroleerd worden of het systeem aan de functionele eisen voldoet.

5.1.1 Unittests

Unit tests zijn de eerste tests die doorlopen moeten worden. Een unit test gaat over een zo klein mogelijk onderdeel van het systeem, een unit. Bij object oriented programming is dit een klasse of een methode.

5.1.2 Systeemtest

Integratie tests zijn tests van een aantal units samen. Hiermee kan de functionaliteit voor een user story getest worden. Voordat deze tests kunnen worden uitgevoerd moet elke losse unit al getest zijn. Na de eerste user story getest te hebben, kunnen steeds units worden toegevoegd voor de volgende user story, tot het systeem compleet is.

5.1.3 Acceptatietest

Na elke sprint zal een acceptatietest worden uitgevoerd door de product owner. Hierin zullen de tot dan toe opgeleverde functies getest worden.

5.2 Aanpak

5.2.1 Test suite

De test suite is een verzameling van alle hierboven beschreven tests. Deze test suite zal worden uitgevoerd aan het einde van de sprint.

5.3 Testomgeving

De software zal getest worden op OSX 10.8. Voor het testen zal nog onderzoek moeten worden gedaan naar een geschikt test framework.

Voor het bijhouden van de problemen in de code zal gebruik worden gemaakt van TRAC, de standaardomgeving binnen TASS. Hiermee kunnen tickets worden aangemaakt waar bugs, feature request of een ticket voor een ander probleem.

5.4 Testrapportage

Een testrapport beschrijft de volgende punten:

- Uitgevoerde tests
- Hoe zijn de tests uitgevoerd
- Resultaten van de test

6 Configuratie Management Plan

6.1 Organisatie

Er gebruikt gemaakt worden van Subversion (SVN). De svn server zal worden beheerd door TASS. Alle projectleden zijn zelf verantwoordelijk voor het gebruik van de svn server.

6.2 Configuratie Beheer

De producten die onder configuratiebeheer vallen zijn:

- Project documenten
- Broncode
- Stage documenten

Conventie TASS documenten

- Geschreven in kleine letters.
- Alle spaties worden vervangen door lage streepjes `_'.
- Heeft de volgende indeling: <naam>_<versie>.<extentie>

6.3 Change Management

Voor documenten word het volgende formaat gebruikt naam_document_v_y_z

- De X word verhoogd met 1 wanneer het document klaar is voor release.
- De Y word verhoogd wanneer er nieuwe informatie word toegevoegd, verwijderd of gewijzigd word in het document.
- De Z word verhoogd wanneer er kleine aanpassingen worden gedaan, zoals spelfouten, zinsopbouw.

Programmeer code word bijgehouden in SVN. Hierin zullen de Commit berichten worden gebruikt om aan te geven wat er veranderd is. Een bericht zal er als volgt uit zien:

- PROJECT/APPLICATIE – KLASS – Wat er veranderd is

Er kan van dit bericht worden afgeweken wanneer het niet van toepassing is op commit. Zoals bij het verplaatsen van mappen of aanpassen van de SVN Structuur.

7 Kwaliteit

7.1 Proces

Het project wordt uitgevoerd volgens de guidelines en eisen zoals beschreven in de TASS Handboek Projecten.

7.2 Reviews

Aan het einde van elke sprint zal een meeting plaatsvinden met de klant en technische begeleider. In deze meeting wordt de sprint gereviewed en zal een kleine demonstratie plaatsvinden van de producten die ontwikkeld zijn tijdens de sprint. De feedback die tijdens deze meetings wordt gegeven zal verwerkt worden in het product.

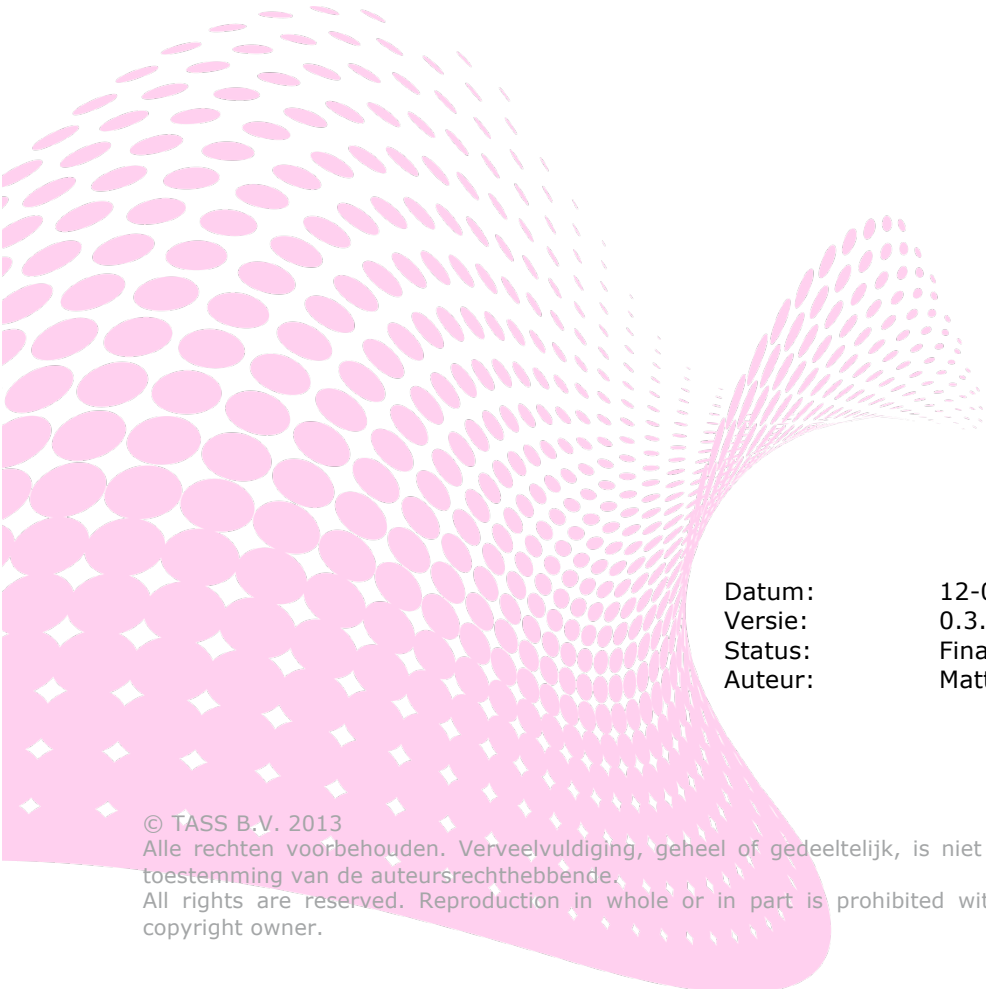
7.3 Testrapportages

De testrapportages zullen dienen als controlemiddel, zodat duidelijk is dat alle vereiste tests zijn uitgevoerd voordat een release heeft plaatsgevonden. Deze testrapportages zullen verder informatie bevatten over de performance van het systeem, zodat het duidelijk is of de nieuwe versie verbeteringen heeft doorgevoerd in de snelheid van het systeem. Verder zal een aanpak beschreven worden voor het oplossen van de bugs die naar voren komen uit de tests.

Bijlage II: Requirements document

Requirements Specificatie

Shakies Passbook



Datum:	12-03-2013
Versie:	0.3.0
Status:	Final
Auteur:	Matthijs van Driel

© TASS B.V. 2013

Alle rechten voorbehouden. Verveelvuldiging, geheel of gedeeltelijk, is niet toegestaan dan met schriftelijke toestemming van de auteursrechthebbende.

All rights are reserved. Reproduction in whole or in part is prohibited without the written consent of the copyright owner.

Dit document is gepubliceerd door:

TASS B.V.
Eindhoven, Nederland

Commentaar en suggesties kunnen worden gestuurd naar:

TASS B.V.
Postbus 80060
5600 KA EINDHOVEN
Nederland
tel: +31 (0)40 2503200
fax: +31 (0)40 2503201

1	Introductie	1
1.1	Doel	1
1.2	Afkortingen en Definities	1
1.3	Referenties	1
1.4	Distributielijst	1
1.5	Document Geschiedenis	1
2	Product Beschrijving	2
2.1	Scope	2
2.2	Perspectief	3
2.3	Functionaliteit	3
2.4	Beperkingen	3
2.5	Gebruikers	3
2.6	Aannames en Afhankelijkheden	3
3	Specifieke Requirements	4
3.1	Functionele Requirements	4
3.1.1	Passbook	4
3.1.2	Webservice	5
3.1.3	Medewerkers applicatie	7
3.2	Externe Interface Requirements	9
3.2.1	Gebruikers Interfaces	9
3.2.1.1	Webservice	9
3.2.1.2	Medewerkers Applicatie	9
3.2.1.3	Passbook	14
3.2.2	Software Interfaces	14
3.2.2.1	Sequentiediagram: Koop KoffieKaart	15
3.2.2.2	Sequentiediagram: Koop Koffie	16
3.2.2.3	Sequentiediagram: Verwijder Kaart	17
3.2.2.4	Sequentiediagram: Klant stop Push notificaties	18
3.3	Performance Requirements	19
3.4	Design Constraints	19
3.4.1	Standaarden	19
3.4.2	Software Beperkingen	19
3.5	Niet-Functionele Requirements	19

1 Introductie

1.1 Doel

Dit document beschrijft:

- De functionele en niet-functionele requirements voor de te ontwikkelen software
- De eventuele relaties en interfaces van de software met andere producten
- Hoe de software gebruikt kan worden

1.2 Afkortingen en Definities

Afk.	Afkorting
PB	Passbook
WS	Webservice
MA	Medewerkers Applicatie (App)

1.3 Referenties

[PMP]	Titel	Project Management Plan Shakies passbook koffiekaart
	Auteur	Matthijs van Driel
	Versie	0.2.0
	Datum	27-03-2013

1.4 Distributielijst

Naam	Organisatie	Functie
Josee Zielman	TASS Technology Solutions	People Manager
Rob Knubben	TASS Technology Solutions	Technisch begeleider
Rene M.J.W. van der Heijden	Fontys Hogeschool ICT	Docent begeleider
Matthijs van Driel	TASS Technology Solutions	Afstudeerder

1.5 Document Geschiedenis

Versie	Status	Datum	Auteur	Wijzigingen
0.1.0	Status	05-03-2013	Matthijs van Driel	- Initiële versie
0.2.0	Draft	07-03-2013	Matthijs van Driel	- Saldo vervangen voor tegoed - Feedback Rob verwerkt
0.3.0	Draft	12-03-2013	Matthijs van Driel	- Sequentiediagrammen toegevoegd - Screen flow toegevoegd

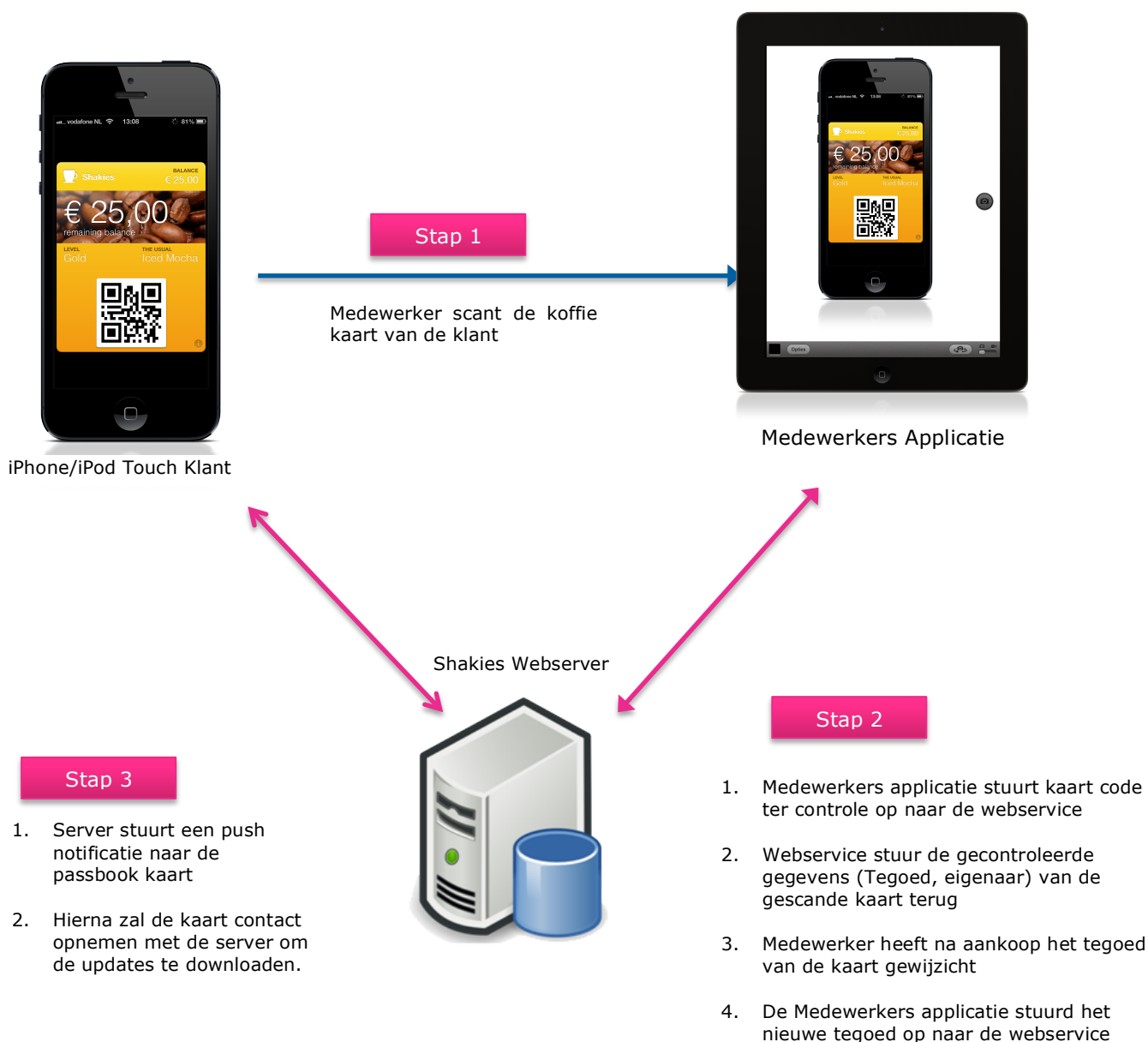
2 Product Beschrijving

2.1 Scope

Dit document beschrijft de requirements voor de shakies passbook applicaties.

- De koffiekaart is een digitale strippen kaart. Deze passbook kaart laat het tegoed zien, en kan worden geüpdatet via de webservice.
- De webservice applicatie kan passbook kaarten aanmaken, en verzorgd de verbinding met de passbook kaart en de medewerkers applicatie.
- De medewerker app beheert de kaarten, na het scannen (van een 2D barcode) en herkennen van de kaart, kan het tegoed veranderd worden. Ook kunnen er nieuwe kaarten worden aangemaakt.

Alle producten zijn bedoelt om samen de papieren koffiekaart te vervangen voor een digitale variant. Daarnaast is het ook bedoeld voor het opdoen en delen van kennis.



2.2 Perspectief

Dit product word een individueel product, en is niet gerelateerd tot een ander product.

2.3 Functionaliteit

Verschillende gebruikers kunnen het volgende doen:

- Koffie afrekenen met de passbook kaart
- Passbook kaart bekijken
- Nieuwe passbook kaarten aanmaken
- Bestaande passbook kaarten scannen
- Bestaande passbook kaarten beheren

2.4 Beperkingen

Dit product heeft de volgende beperkingen:

- De medewerkers applicatie moet worden gemaakt voor de iPad.
- Apple legt strikt vast hoe de communicatie geregeld moet zijn.
- Apple legt strikt vast hoe de passbook kaart er uit moet zien en welke 2D barcodes beschikbaar zijn.

2.5 Gebruikers

Dit product kent verschillende gebruikers. De rollen zijn als volgt gedefinieerd:

Koffiekaart klant

De gebruiker die een koffiekaart koopt en deze als volgt beschikbaar heeft binnen de iOS passbook applicatie

Shakies medewerker:

De medewerker scant de koffiekaart, waarna deze beheerd kan worden.

2.6 Aannames en Afhankelijkheden

De volgende aannames en afhankelijkheden zijn van toepassing op dit product:

- De front camera van de iPad is goed genoeg voor het scannen van 2D barcodes

3 Specifieke Requirements

3.1 Functionele Requirements

ID	Beschrijving/ Feature
PB-1	De shakies (Passbook) kaart beschikbaar binnen de passbook applicatie met duidelijke weergave van het tegoed en 2D barcode
PB-2	Klant krijg bij aankoop koffie push notificatie over update van saldo
PB-3	Klant moet koffie kaart kunnen verwijderen uit passbook
PB-4	Klant moet de koffie kaart kunnen toevoegen via E-mail, Download of iOS App
PB-5	Notificatie laten zien wanneer een klant bij een shakies locatie komt
WS-1	Genereren van een Passbook kaart
WS-2	Klanten moeten hun kaart kunnen updaten
WS-3	Server moet kaart kunnen updaten via Push notificaties
WS-4	De klant moet de passbook kaart kunnen installeren
WS-5	Medewerkers applicatie moet de kaart kunnen beheren via een API
MA-1	Barcode scannen vanaf een passbook kaart
MA-2	Tegoed van Passbook kaart kunnen bekijken
MA-3	Tegoed van Passbook kaart kunnen veranderen
MA-4	Nieuwe Passbook kaart kunnen aanmaken

3.1.1 Passbook

Feature		
ID	PB-1	
Story	Om	De shakies kaart te zien
	Als een	Klant
	Wil ik	De passbook applicatie openen en de shakies kaart selecteren
	Scenario 1 De shakies passbook kaart zien	
	Gegeven	Ik ben een applicatie gebruiker
	En	Open de passbook applicatie
	En	Selecteer de shakies kaart
	Wanneer	Ik wil mijn shakies tegoed zien
	Dan	Word de shakies kaart weergegeven. Hier op is het tegoed, en de 2D barcode te zien
	Scenario 2 Er is geen shakies passbook kaart	
	Gegeven	Ik ben een applicatie gebruiker
	En	Open de passbook applicatie
	En	Kan geen shakies kaart vinden
	Wanneer	Ik heb de passbook applicatie open, maar kan geen shakies kaart vinden
	Dan	Er moet eerst een shakies kaart worden toegevoegd

Feature Klant moet koffie kaart kunnen verwijderen uit passbook		
ID	PB-3	
Story	Om	Een koffie kaart te verwijderen
	Als een	Klant
	Wil ik	De shakies koffie kaart verwijderen uit passbook
	Scenario 1 Kaart verwijderen	
	Gegeven	Passbook applicatie is geopend
	En	Shakies koffie kaart is geselecteerd
	En	De achterkant van de shakies koffie kaart is zichtbaar
	Wanneer	Ik klik op het prullenbak icoontje links boven
	Dan	Word ik gevraagd om de verwijder actie te bevestigen
	En	Klik ik op verwijder
	En	De kaart is verwijderd
	Scenario 2 Kaart verwijderen, cancel	
	Gegeven	Passbook applicatie is geopend
	En	Shakies koffie kaart is geselecteerd
	En	De achterkant van de shakies koffie kaart is zichtbaar
	Wanneer	Ik klik op het prullenbak icoontje links boven
	Dan	Word ik gevraagd om de verwijder actie te bevestigen
	En	Klik ik op cancel
	En	De kaart is niet verwijderd

3.1.2 Webservice

Feature Genereren van een Passbook kaart		
ID	WS-1	
Story	Om	Een shakies kaart te genereren
	Als een	Shakies Medewerker
	Wil ik	Een nieuwe shakies kaart aanmaken
	Scenario 1 Shakies kaart kan worden gegenereerd	
	Gegeven	Ik ben een shakies medewerker
	En	Medewerkers applicatie is opgestart
	En	Er is een internet verbinding
	En	De medewerker vult alle gegevens (Naam en-mail adres) van de klant juist in.
	Wanneer	De medewerker klikt op opslaan
	Dan	Word de shakies kaart gegenereerd en aangeboden aan de koffiokaart klant
	Scenario 2 Shakies kaart kan niet worden gegenereerd	
	Gegeven	Ik ben een shakies medewerker
	En	Medewerkers applicatie is opgestart
	En	Er is een internet verbinding
	En	De medewerker vult alle gegevens (Naam en-mail adres) van de klant juist in
	Wanneer	De medewerker klikt op opslaan
	Dan	Medewerker krijgt een Error te zien wat er is fout gegaan
	En	Medewerker moet de fout aanpassen en het opnieuw proberen

Feature	Communicatie van de Passbook kaart (gebruiker refreshed de kaart)	
ID	WS-2	
Story	Om	De kaart te updaten
	Als een	Koffiekaart klant
	Wil ik	Mijn koffie kaart updaten
	Scenario 1 Shakies kaart kan worden geüpdatet	
	Gegeven	Ik ben een koffiekaart klant
	En	De passbook applicatie is geopend
	En	De shakies kaart is geselecteerd
	En	Er is een internet verbinding
	En	Webservice is bereikbaar
	Wanneer	De klant een pull to refresh actie uitvoert op de kaart
	Dan	Word de kaart geüpdatet vanaf de webservice
	Scenario 2 Shakies kaart kan niet worden geupdate	
	Gegeven	I ben een koffiekaart klant
	En	De passbook applicatie is geopend
	En	De shakies kaart is geselecteerd
	En	Er is een internet verbinding
	En	Webservice is niet bereikbaar
	Wanneer	De klant een "pull to refresh" actie uitvoert op de kaart
	Dan	De kaart kan de webservice niet bereiken
	En	Kaart word niet geüpdatet, gebruiker krijgt een error te zien dat de kaart niet geüpdatet kan worden

Feature	De klant moet de passbook kaart kunnen installeren	
ID	WS-4	
Story	Om	Een shakies kaart te kunnen gebruiken
	Als een	Koffiekaart klant
	Wil ik	Een shakies kaart installeren
	Scenario 1 E-mail installatie	
	Gegeven	De webservice heeft een kaart gegenereerd en opgestuurd
	En	De klant heeft de email met de kaart geopend
	Wanneer	De gebruiker klikt op de kaart in de email
	Dan	iOS vraagt of de klant de kaart wil toevoegen aan de Passbook applicatie
	En	Klant kiest "Toevoegen"
	En	De kaart word toegevoegd aan de passbook applicatie
	Scenario 2 Download link installatie	
	Gegeven	De webservice heeft een kaart gegenereerd en een download link opgestuurd
	En	De klant heeft de email met de kaart geopend
	En	De klant heeft internet
	En	De webservice is bereikbaar
	Wanneer	De klant klikt op de download link
	Dan	Word safari geopend
	En	iOS vraagt of de klant de kaart wil toevoegen aan de Passbook applicatie

En	Klant kiest "Toevoegen"
En	De kaart word toegevoegd aan de passbook applicatie
Scenario 3 E-mail installatie, cancel	
Gegeven	De webservice heeft een kaart gegenereerd en opgestuurd
En	De klant heeft de email met de kaart geopend
Wanneer	De gebruiker klikt op de kaart in de email
Dan	iOS vraagt of de klant de kaart wil toevoegen aan de Passbook applicatie
En	Klant kiest "Cancel"
En	De kaart word niet toegevoegd aan de passbook applicatie

3.1.3 Medewerkers applicatie

Feature Passbook kaart barcode kunnen scannen met de front camera van de iPad		
ID	MW-1	
Story	Om	Een shakies kaart te beheren
	Als een	Shakies Medewerker
	Wil ik	Een shakies kaart scannen
	Scenario 1 Kaart word gescand met de front camera van de iPad	
	Gegeven	Klant heeft passbook applicatie geopend en de shakies kaart geïnstalleerd
	En	Medewerker heeft de shakies applicatie geopend
	En	Medewerker heeft internet en de webservice is bereikbaar
	Wanneer	Klant laat zijn shakies kaart zien
	Dan	Medewerker selecteert de scan optie
	En	Medewerker scant de kaart.
	En	Medewerker krijg te zien dat het scannen is gelukt
	Scenario 2 Kaart kan niet worden gescand met de front camera van de iPad	
Gegeven	Klant heeft passbook applicatie geopend en de shakies kaart geïnstalleerd	
En	Medewerker heeft de shakies applicatie geopend	
En	Medewerker heeft internet en de webservice is bereikbaar	
En	Klant laat zijn shakies kaart zien	
En	Medewerker heeft de scan optie geselecteerd	
Wanneer	Medewerker scant de kaart.	
Dan	Kaart word niet herkent	
En	Medewerker krijgt een melding te zien dat de kaart niet herkent word	

Feature Tegoed van Passbook kaart kunnen bekijken		
ID	MW-2	
Story	Om	Te controleren of de klant nog tegoed heeft
	Als een	Shakies medewerker
	Wil ik	Het tegoed van de kaart kunnen zien

Scenario 1 Tegoed bekijken	
Gegeven	MW-1 moet gelukt zijn
En	Medewerker heeft het kaart overzicht scherm open
Wanneer	De medewerker kijkt in het overzicht scherm van de kaart
Dan	Kan de medewerker het tegoed zien

Feature Tegoed van Passbook kaart kunnen veranderen		
ID	MW-3	
Story	Om	Het tegoed te veranderen
	Als een	Shakies Medewerker
	Wil ik	Het nieuw gekochte tegoed invoeren
Scenario 1 Tegoed verhogen		
Gegeven	MW-1 moet gelukt zijn	
Wanneer	Selecteer ik wijzig tegoed	
Dan	Ik verhoog het tegoed met 1	
En	WS- word uitgevoerd	
Scenario 2 Tegoed verlagen		
Gegeven	MW-1 moet gelukt zijn	
Wanneer	Selecteer ik wijzig tegoed	
Dan	Ik verlaag het tegoed met 1	

Feature Nieuwe Passbook kaart kunnen aanmaken		
ID	MW-4	
Story	Om	Een nieuwe kaart te maken
	Als een	Shakies Medewerker
	Wil ik	Een nieuwe shakies koffiokaart maken
Scenario 1 Medewerker maakt nieuwe shakies kaart		
Gegeven	Medewerkers applicatie is opgestart	
En	Er is internet beschikbaar en de webservice is bereikbaar	
Wanneer	Ik kies voor nieuwe kaart, Vul ik alle gegevens (Naam, E-mail adres) van de klant in, klik ik op opslaan	
Dan	Word er een nieuwe kaart gemaakt	
En	Opgestuurd naar de klant	
Scenario 2 Medewerker maakt nieuwe shakies kaart met verkeerde gegevens		
Gegeven	Medewerkers applicatie is opgestart	
En	Er is internet beschikbaar en de webservice is bereikbaar	
Wanneer	Ik kies voor nieuwe kaart, Vul ik een getal in het E-mail adres veld, klik ik op opslaan	
Dan	Word er een error getoond dat het E-mail adres verkeerd is ingevuld.	
En	Medewerker moet een geldig E-mail adres invullen	

3.2 Externe Interface Requirements

3.2.1 Gebruikers Interfaces

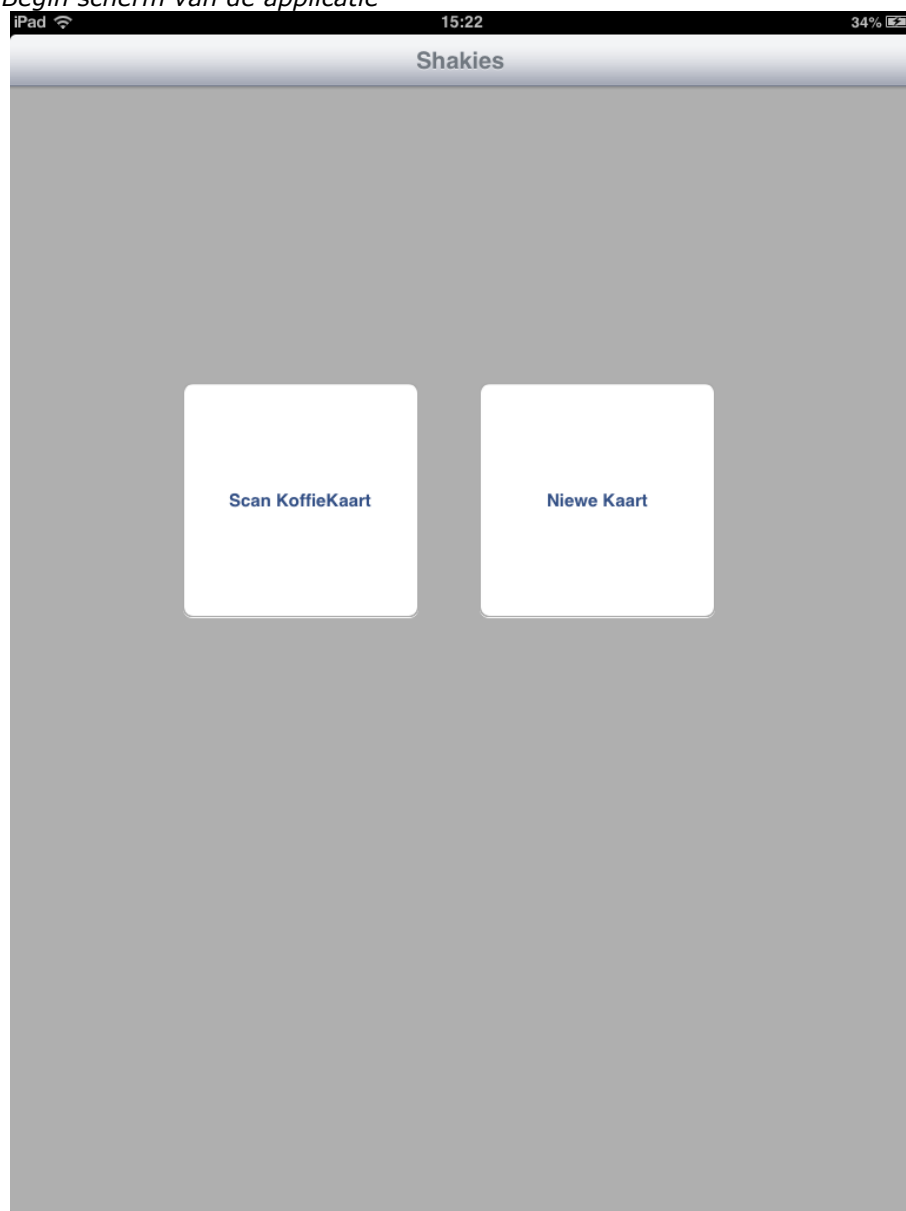
3.2.1.1 Webservice

De webservice zal geen GUI (Grafical user interface) krijgen.

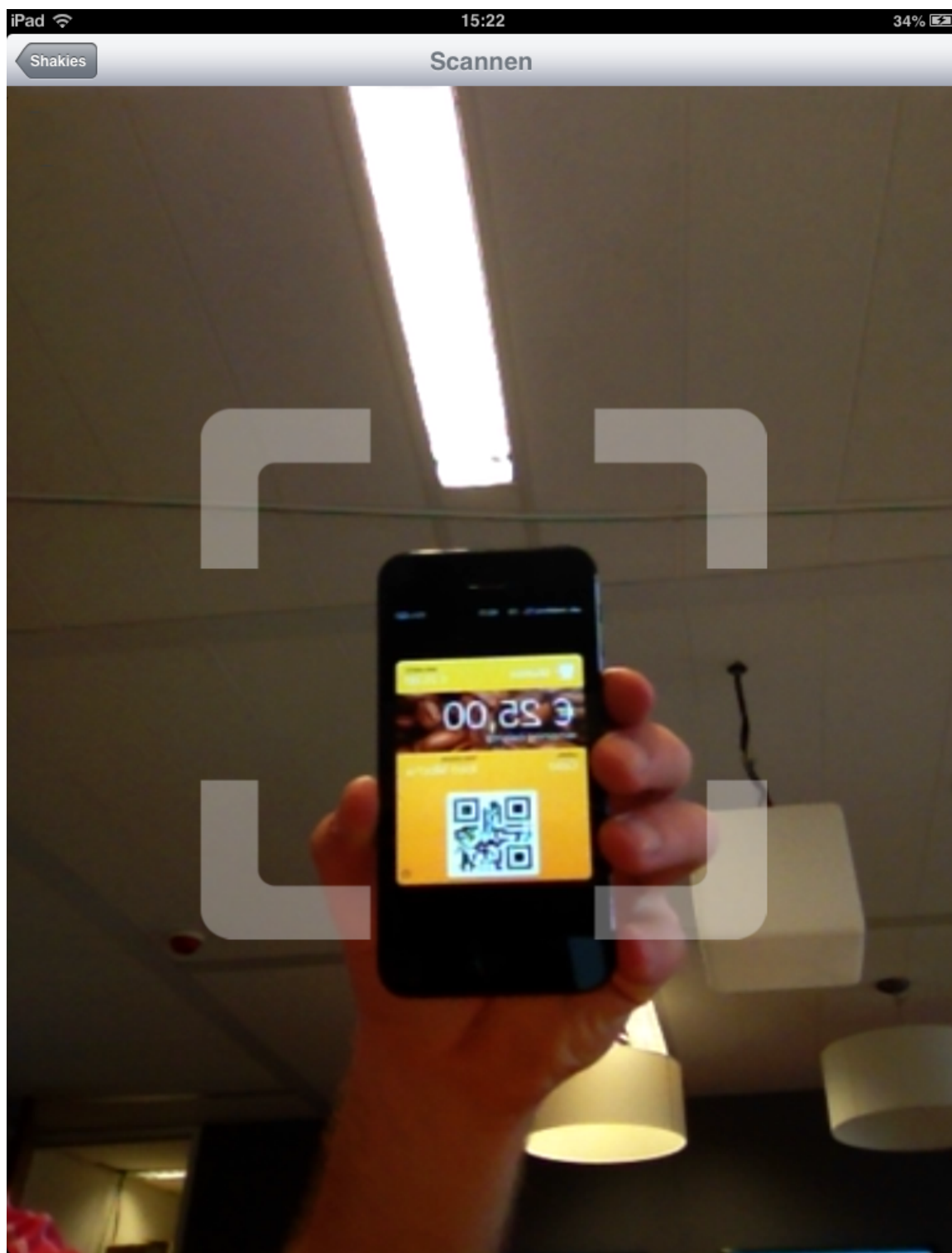
3.2.1.2 Medewerkers Applicatie

De medewerkers applicatie zal ongeveer de volgende interface krijgen op de iPad

Begin scherm van de applicatie




Hier moet de medewerker kiezen wat hij/zei wil doen.



Scan scherm

De medewerker moet zorgen dat de barcode van de koffiekaart goed te zien is door de camera en dat het beeld scherp is zodat de applicatie de informatie uit de barcode kan uitlezen.

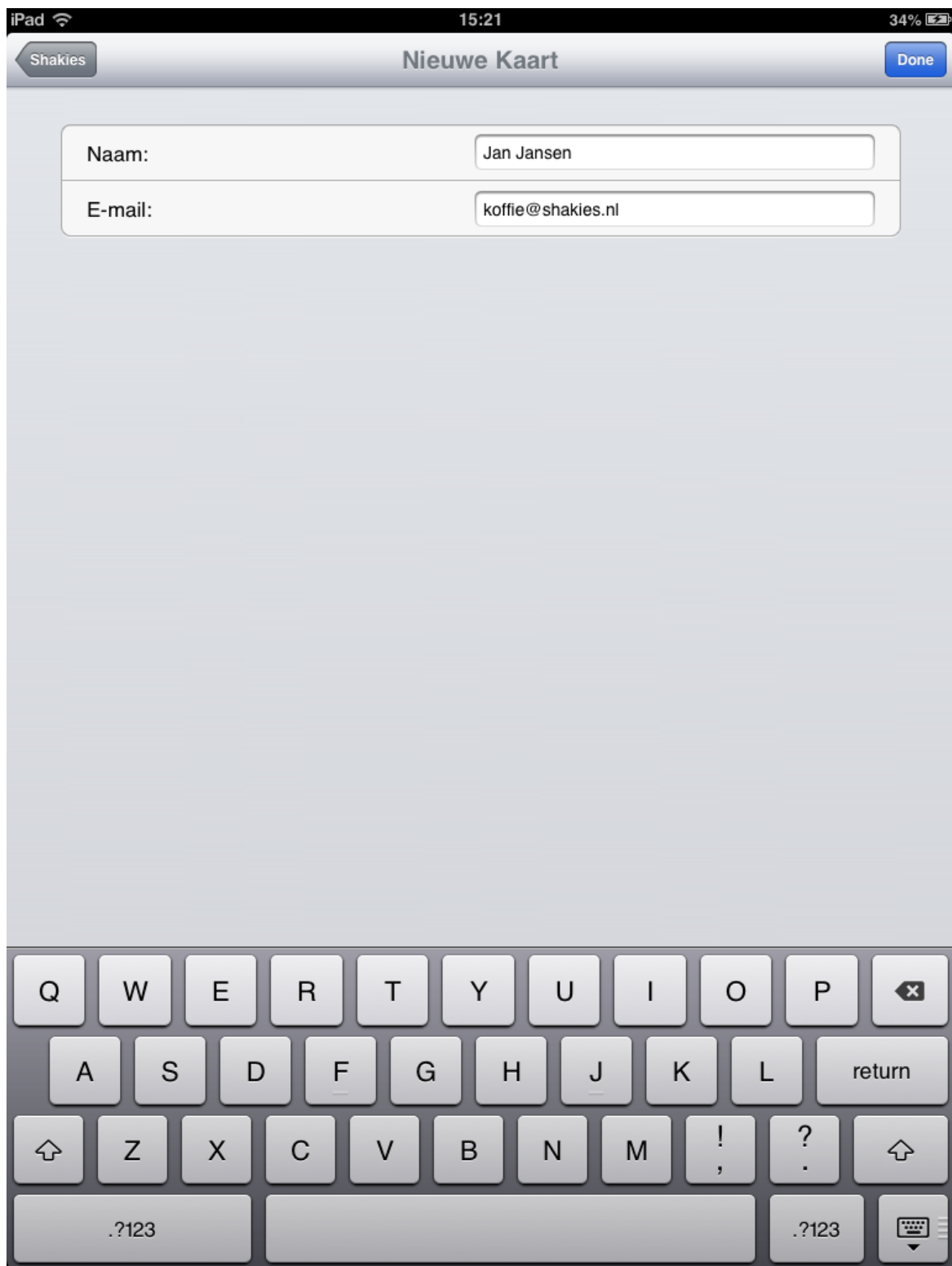
Kaart herkent



Naam:	Jan Jansen
Email:	jan_jansen@tass.nl
Tegoed:	20

Dit scherm krijgt de medewerker te zien wanneer een barcode is herkend.
Op het scherm is te zien van wie de kaart is, en wat het huidige tegoed is.

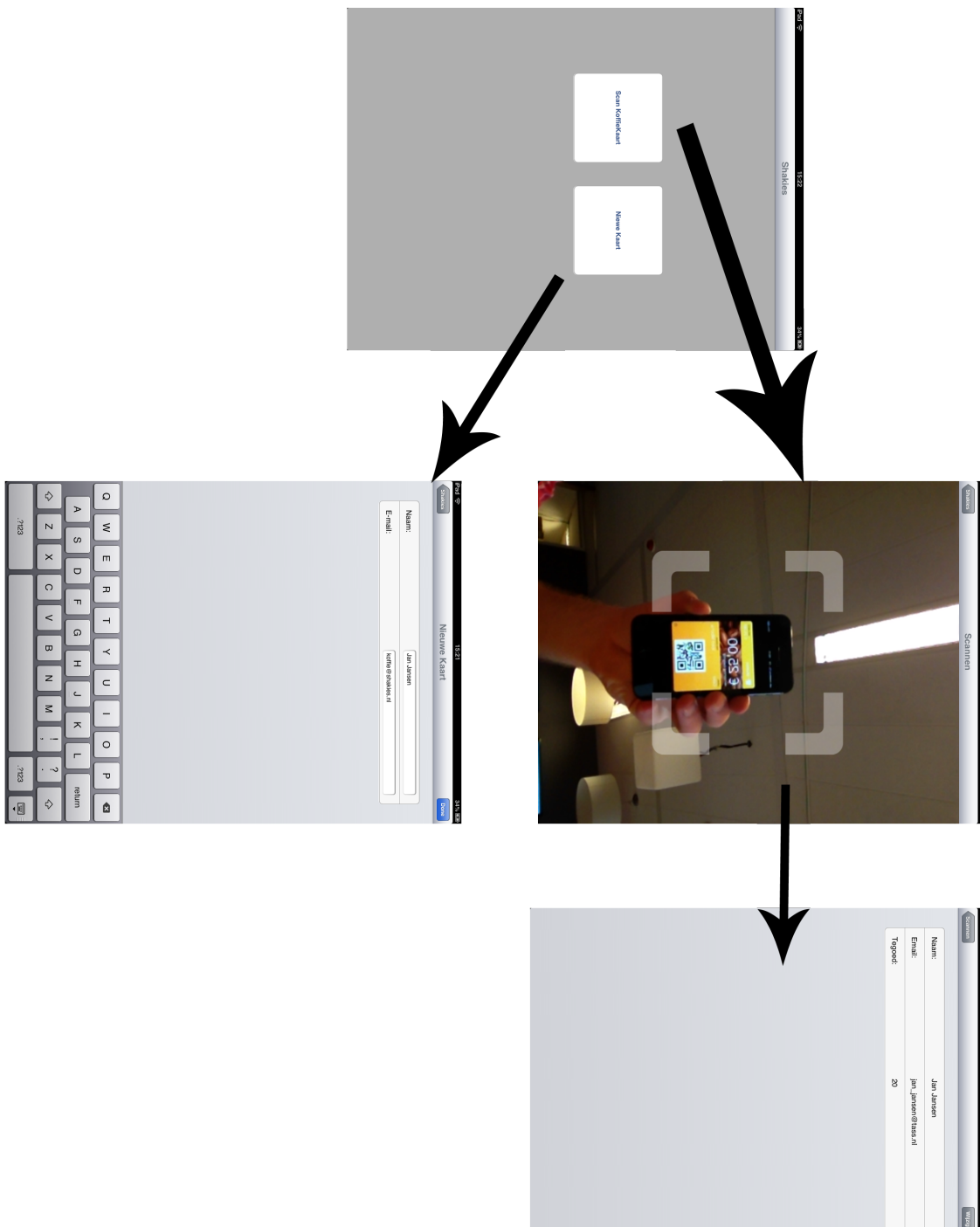
Aanmaken nieuwe kaart



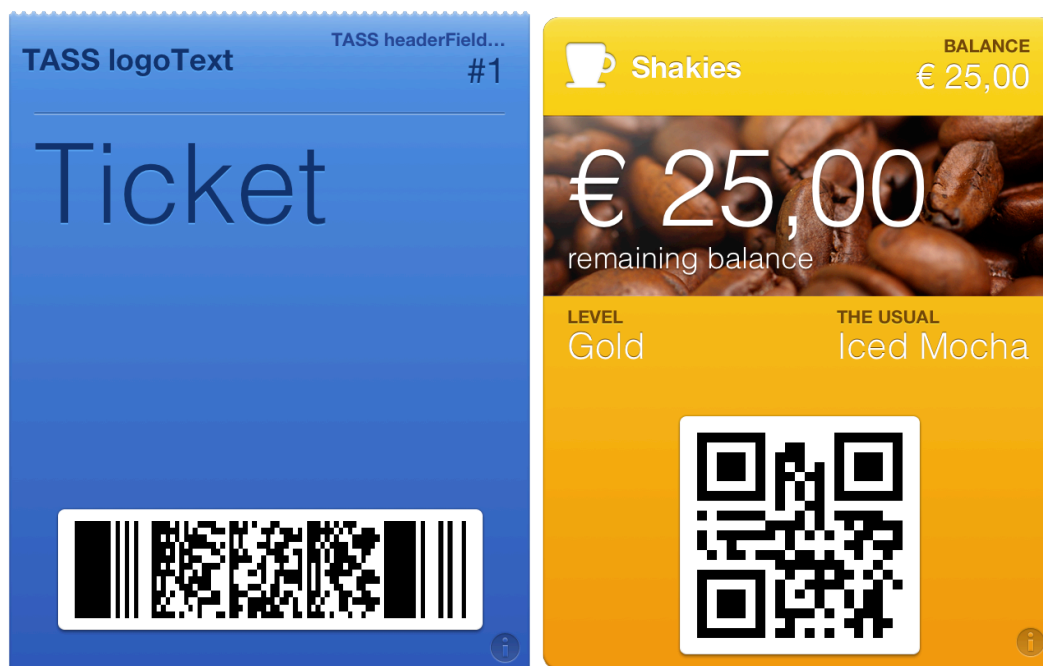
The screenshot shows an iPad interface for creating a new card. At the top, the status bar displays 'iPad', signal strength, time '15:21', and battery level '34%'. Below the status bar is a navigation bar with a 'Shakies' button on the left, the title 'Nieuwe Kaart' in the center, and a 'Done' button on the right. The main content area contains two text input fields. The first field is labeled 'Naam:' and contains the text 'Jan Jansen'. The second field is labeled 'E-mail:' and contains the text 'koffie@shakies.nl'. A virtual QWERTY keyboard is visible at the bottom of the screen.

Hier moet de medewerker de goede gegevens invullen waarna er een bericht naar de webservice.

3.2.1.2.1 Screen Flow



3.2.1.3 Passbook



Kaart 1

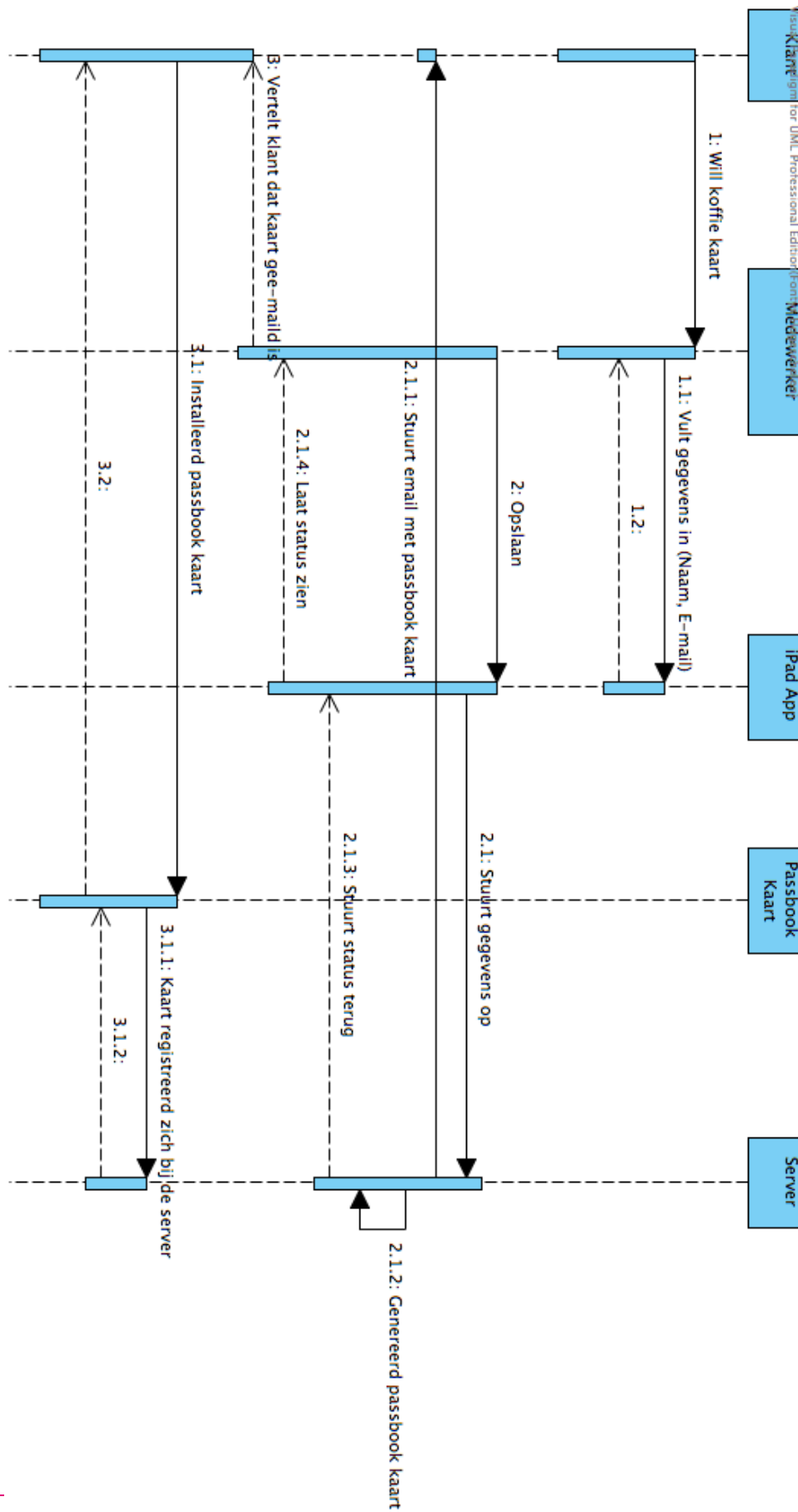
Dit zijn 2 voorbeelden van mogelijke passbook kaarten, waarbij de koffiekaart meer zal lijken op kaart 2.

Kaart 2

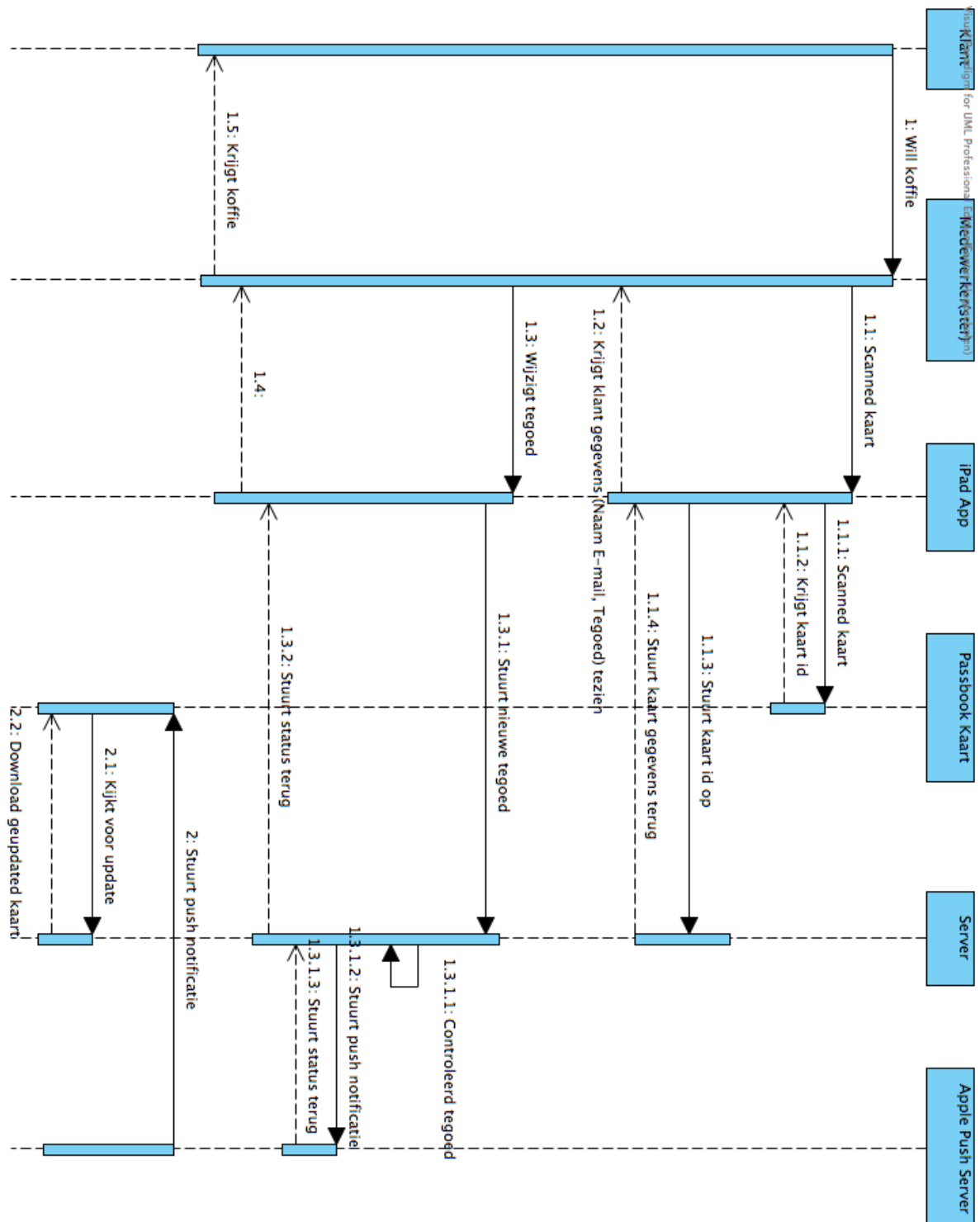
3.2.2 Software Interfaces

- Medewerkers applicatie met de webservice API.
- Webservice met database (waarschijnlijk postgres)
- 2D Barcode scan library

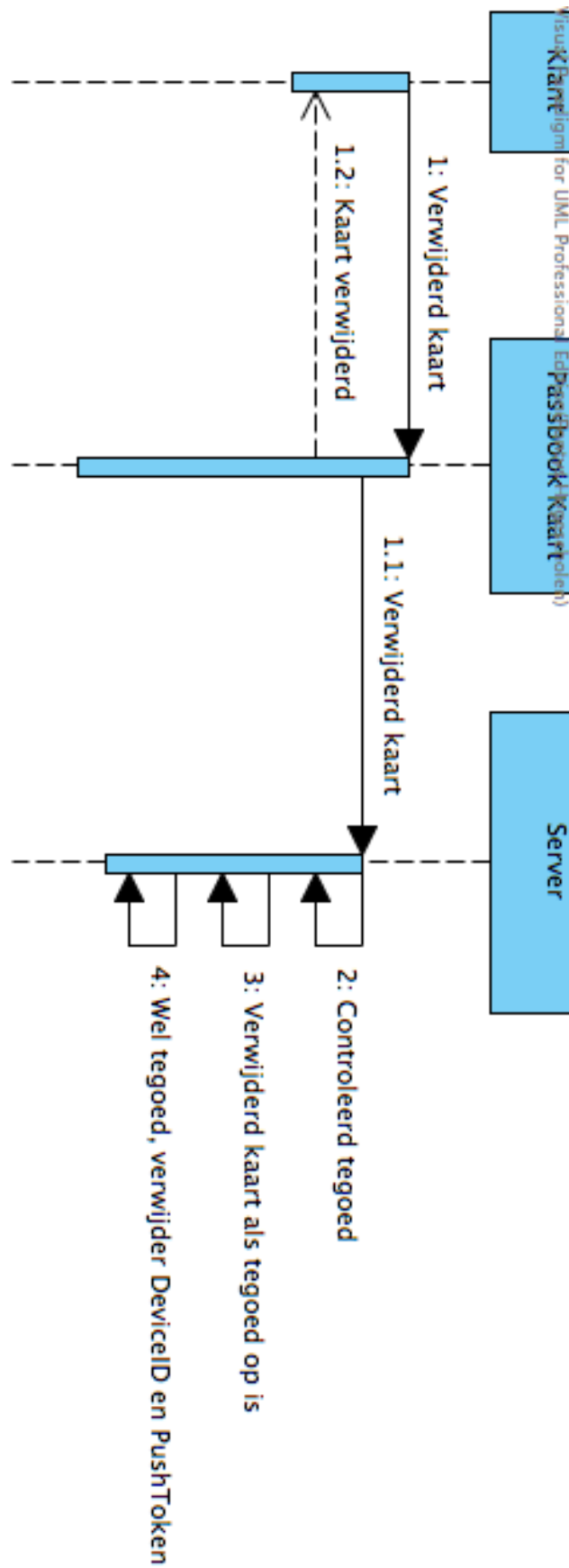
3.2.2.1 Sequentiedagram: Koop KoffieKaart



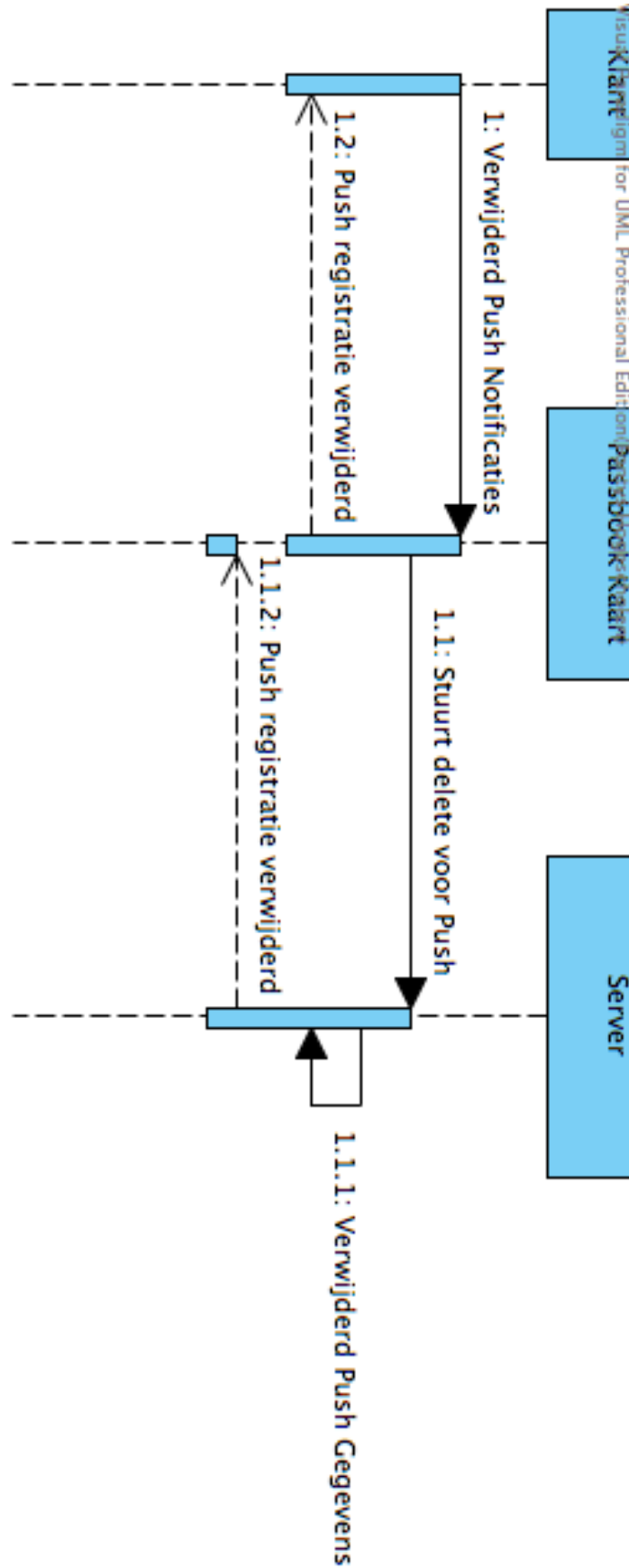
3.2.2.2 Sequentiedagram: Koop Koffie



3.2.2.3 Sequentiediagram: Verwijder Kaart



3.2.2.4 Sequentiedigram: Klant stop Push notificaties



3.3 Performance Requirements

Het maken van een nieuw koffiekaart in de medewerkers applicatie, zal na het opslaan van de gegevens direct worden gemaakt en opgestuurd naar de gebruiker (klant). Wanneer de klant zijn kaart ontvang is afhankelijk van de klant zijn E-mail en Internet providers, en de drukte op de server. In 99% van de gevallen zal dit hele proces binnen 2 minuten afgrond zijn, waarbij de meest tijd zit aan de klant zijn kant.

3.4 Design Constraints

3.4.1 Standaarden

- Er zal voldaan moeten worden aan de eisen die Apple stelt over de verbinding met de webservice.

3.4.2 Software Beperkingen

- Er zal gebruikt gemaakt worden van een Ruby on Rails server image. Binnen deze image zijn al de volgende dingen geconfigureerd.
 - o Database (Postgres)
 - o Ruby on Rails
 - o Apache

3.5 Niet-Functionele Requirements

Betrouwbaarheid	Het systeem mag minder dan 1 keer per maand uitvallen.
Beschikbaarheid	99% beschikbaarheid tussen werktijden.
Beveiliging	Voor alle verbindingen moet een HTTPS verbinding gebruikt worden.

Bijlage III: Onderzoek

Beveiliging Webservice API

Het beveiligen van de Shakies webservice



Datum:	15-03-2013
Versie:	1.0
Status:	Final
Auteur:	Matthijs van Driel

© TASS B.V. 2013

Alle rechten voorbehouden. Vervielfuldiging, geheel of gedeeltelijk, is niet toegestaan dan met schriftelijke toestemming van de auteursrechthebbende.

All rights are reserved. Reproduction in whole or in part is prohibited without the written consent of the copyright owner.

Dit document is gepubliceerd door:

TASS B.V.
Eindhoven, Nederland

Commentaar en suggesties kunnen worden gestuurd naar:

TASS B.V.
Postbus 80060
5600 KA EINDHOVEN
Nederland
tel: +31 (0)40 2503200
fax: +31 (0)40 2503201

1 Verklarende woordenlijst	1
2 Hoofdvraag	2
3 Welke beveiligingsmogelijkheden zijn er beschikbaar binnen Ruby on Rails?	3
3.1 Gebruikersnaam en wachtwoord	3
3.2 HTTP Basic Authentication	3
3.3 Secure Sockets Layer/Transport Layer Security	4
3.4 Certificaat gebaseerde verificatie	5
3.5 Application Programming Interface Token	5
3.6 Open Authorization	6
4 Aan welke mogelijkheden zijn kosten verbonden en hoe hoog zijn deze?	7
5 Welke eisen stelt Apple aan de beveiliging van de passbook kaarten?	8
5.1 De Passbookkaart	8
5.1.1 Manifest.json	8
5.1.2 Signature	8
5.2 Communicatie met de Passbookkaart	8
6 Met welke mogelijkheid/mogelijkheden is er al ervaring binnen TASS?	9
7 Conclusie	10
8 Literatuurlijst	11

1 Verklarende woordenlijst

Begrip	Verklaring
Ruby	Is een programmeer taal. Een programmeertaal is een taal waarin opdrachten voor de computer worden uitgeschreven.
Ruby on Rails	Is een web framework gemaakt boven op Ruby. Met dit framework kunnen websites gemaakt worden
Webserver	Een server waarop een website of API draait.
Passbook	Is een techniek van Apple om papieren kaarten/bonnen/tickets te vervangen voor een digitale variant.
API	Een Application Programming Interface (API) is een verzameling van definities, waardoor een computerprogramma kan communiceren met een ander programma of onderdeel. (Wikipedia, 2013)
Hash	Is een wiskundige berekeningsmethode die wordt toegepast op de sleutel van een record en die een uniek adres oplevert. Het is een techniek die één of meerdere velden van een record omzet in een compacter veld (hash code techniek (Betekenis/definitie van))
Base64	Base64 is een manier om binaire code te converteren naar ASCII-tekens. (Base64 - Wikipedia)
Reverse Engineering	Het ontrafelen van de werking van een apparaat of computerprogramma zonder de beschikking te hebben over de bouwplannen. (Wikipedia, 2013)

2 Hoofdvraag

In dit onderzoek wordt de volgende vraag beantwoord: "Hoe kan de toegang en verbinding met de Shakies webservice het beste beveiligd worden zodat alleen gemachtigde personen en kaarten toegang krijgen?"

3 Welke beveiligingsmogelijkheden zijn er beschikbaar binnen Ruby on Rails?

3.1 Gebruikersnaam en wachtwoord

De meest gebruikte beveiligingsmethode bestaat uit een gebruikersnaam en wachtwoord. De gegevens worden ingevuld op een formulier en worden vervolgens naar een server gestuurd. De server controleert de combinatie van de twee, als deze klopt wordt er toegang gegeven.

Zolang de gebruiker zijn/haar gegevens niet afgeeft en een voldoende lang wachtwoord gebruikt is dit een goede manier van beveiligen.

Het nadeel is dat gegevens zonder SSL(zie Secure Sockets Layer) makkelijk te onderscheppen zijn. Als een hacker deze gegevens heeft, kan deze direct toegang kan worden verkregen.

3.2 HTTP Basic Authentication

HTTP Basic maakt gebruik van de headers in een HTTP Request. Hierdoor hoeven er geen sessies of cookies gemaakt te worden. In de header van het HTTP bericht worden de gebruikersnaam en het wachtwoord opgestuurd. Dit gebeurt als volgt:

1. Gebruikersnaam en wachtwoord worden samengevoegd als volgt:
gebruikersnaam:wachtwoord
2. Hierna worden deze gecodeerd door middel van Base64¹ (dit is geen encryptie!)
3. Het resultaat zal dan worden meegestuurd in de HTTP header:
Authorization: Basic Z2VicVpa2Vyc25hYW06d2FjaHdvd3Jk

Met deze informatie kan de server controleren of de gegevens kloppen en dan toegang verlenen of een error terug sturen. Het grootste voordeel van HTTP basic is dat het makkelijk te gebruiken is en door bijna elke server ondersteund wordt.

Het voordeel is ook gelijk het grootste nadeel. Omdat alles in de http-header zonder encryptie als platte tekst verstuurd wordt, zou een hacker erg eenvoudig de gegevens kunnen onderscheppen en toegang te krijgen. Wanneer dit gebruikt wordt in combinatie met Secure Sockets Layer (SSL) of Transport Layer Security (TLS) maakt het direct een stuk veiliger. (Wikipedia, 2013)(AviD, 2010)

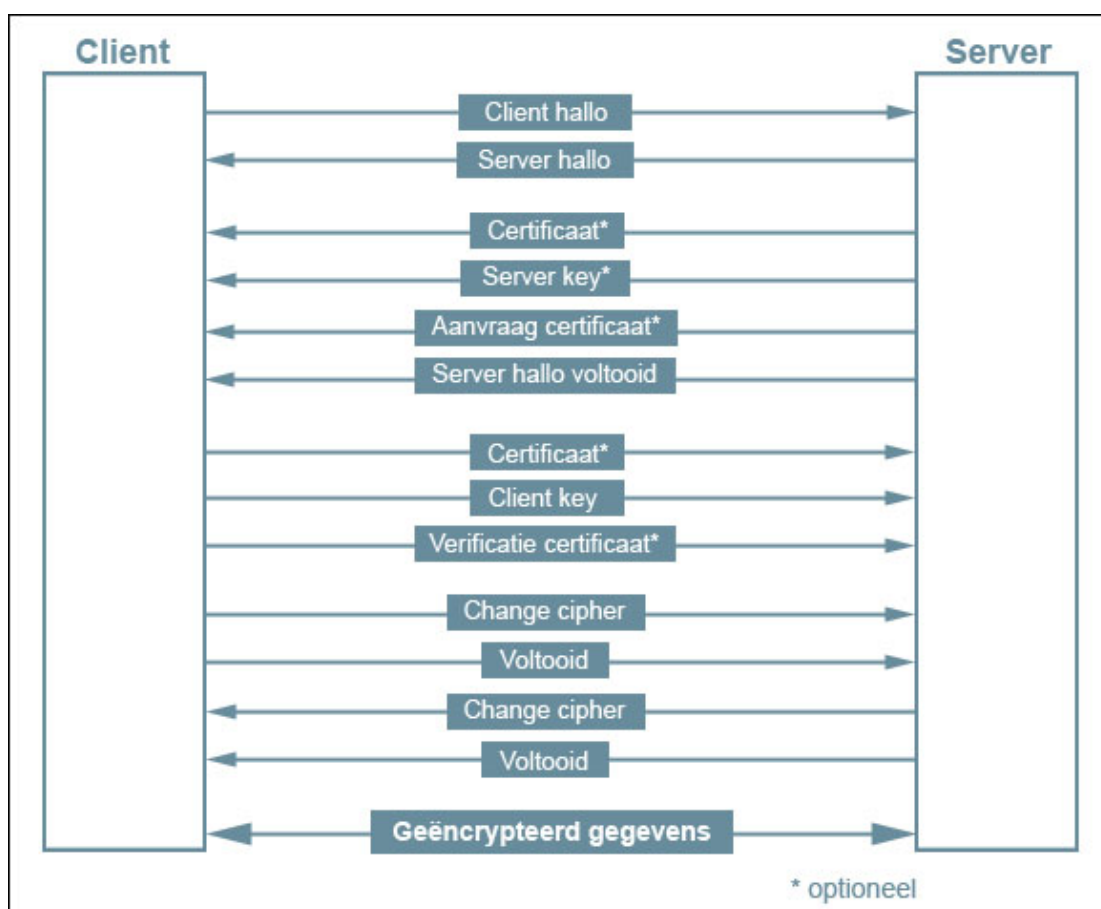
¹ Base64 is een manier om binaire code te converteren naar ASCII-tekens.

3.3 Secure Sockets Layer/Transport Layer Security

Secure Sockets Layer (SSL) en Transport Layer Security (TLS) worden gebruikt voor het beveiligen van de communicatie door middel van encryptie en het controleren of de server waarmee verbonden wordt ook echt de goede server is. TLS is de opvolger van SSL, maar de afkorting SSL is blijven hangen.

Wanneer een client verbinding maakt met een server, wordt er tussen de client en server een geheime sleutel afgesproken. Nadat deze twee overeen komen kan er veilig worden gecommuniceerd, want vanaf dat moment wordt de data geëncrypt met een sleutel die alleen de client en server weten.

Verbinding tussen cliënt en server geïllustreerd:



(Jeffreytedjosukmono, 2006)

Beiden maken gebruik van asymmetrische cryptografie² voor authenticatie en symmetrische encryptie³ voor het beveiligen van de data. Daarnaast wordt ook "Message authentication code" gebruikt, dit is een klein stukje informatie om te controleren of het bericht intact is. Zo kan het bericht onderweg niet worden aangepast.

² Bij asymmetrische cryptografie wordt gebruikgemaakt van twee aparte sleutels: één wordt gebruikt om de informatie te coderen (vercijferen) of te ondertekenen, de tweede sleutel om de informatie weer te decoderen (ontcijferen) of de identiteit van de afzender te verifiëren. (Wikipedia, 2013)

³ Bij symmetrische cryptografie wordt dezelfde cryptografische sleutel gebruikt voor zowel het vercijferen van informatie als voor het ontcijferen ervan. (Wikipedia, 2013)

3.4 Certificaat gebaseerde verificatie

Bij certificaat gebaseerde verificatie verwacht de server een certificaat van de client die verbinding maakt met de server. Wanneer de client het juiste certificaat heeft, zal de server toegang geven tot de webservice.

Het voordeel is dat gebruikers geen gebruikersnaam en wachtwoord nodig hebben om toegang te krijgen. Dit omdat het certificaat de server vertelt, wie de persoon of applicatie is die probeert verbinding te maken. Omdat het certificaat wordt uitgegeven via een certificaatautoriteit is er gecontroleerd of de persoon echt is wie hij zegt dat die is.

Het grootste nadeel hiervan is dat het certificaat gestolen kan worden door een hacker of virus, die direct toegang KUNNEN krijgen via het geldige certificaat.

Wanneer dit in een publieke applicatie gebruikt wordt, zou de applicatie gereverse engineer⁴ kunnen worden. Hierdoor kunnen hackers ook toegang krijgen met hetzelfde resultaat. (cryptography - Certificate based authentication vs Username and Password authentication - IT Security Stack Exchange, 2011)

3.5 Application Programming Interface Token

Application Programming Interface (API) Token is een token die wordt meegestuurd naar de server op het moment van verbinden. Er wordt dan gecontroleerd of de token bestaat, als dit het geval is wordt toegang gegeven tot de server. Vaak wordt er naast deze token een gebruikersnaam en wachtwoord gebruikt om aparte gebruikers binnen één token te kunnen onderscheiden. Deze is zelf vaak een hash⁵, die gegenereerd wordt op waardes die de maker vaststelt.

Er zijn veel websites en programma's die gebruik maken van deze techniek, omdat het makkelijk te maken en te gebruiken is. Hierdoor kan de maker makkelijk bepaalde applicaties of gebruikers toegang geven of deze weer afnemen zonder dat het andere gebruikers beïnvloed.

Het grote nadeel ook hier is dat wanneer de communicatie gebeurt over HTTP zonder encryptie., een hacker het verkeer kan afluisteren en alle benodigde gegevens kan verkrijgen.

⁴ "Reverse engineering is het onderzoeken van een product (meestal een stuk software of een communicatieprotocol) om daaruit af te leiden wat de eisen zijn waaraan het product probeert te voldoen of om de precieze interne werking ervan te achterhalen." (Reverse engineering - Wikipedia, 2013)

⁵ "Een hashfunctie of klutsfunctie is een functie in de informatica die invoer uit een breed domein van waarden omzet in een (meestal) kleiner bereik, meestal een deelverzameling van de gehele getallen". (Wikipedia, 2013)

3.6 Open Authorization

Open Authorization beter bekend als OAuth is een standaard voor autorisatie. Hiermee kan een applicatie toegang krijgen tot een service, nadat de gebruiker hiervoor toestemming heeft gegeven op de website van de service aanbieder.

Bijvoorbeeld:

1. Gebruiker heeft een applicatie voor Twitter.
2. In plaats van zijn/haar gebruikersnaam en wachtwoord in te vullen, wordt de website van Twitter geopend.
3. Hier logt de gebruiker in.
4. Dan word er gevraagd of de applicatie toegang mag hebben.
5. Gebruiker staat dit toe.
6. De applicatie heeft toegang tot de Twittergegevens van de gebruiker.

Dit word ook wel een *“Three-Legged OAuth”* genoemd omdat er drie personen of applicaties in het spel zijn. De gebruiker, een Twitter applicatie en de Twitter server.

De gebruiker vult zijn gebruikersnaam en wachtwoord alleen in op de website van de service aanbieder (Twitter). Hierdoor hebben externe applicaties geen toegang tot de login gegevens. Hierna, net zoals bij API-tokens, word er gebruik gemaakt van tokens. Daarom kunnen applicaties direct de toegang ontzegt worden zonder dat een gebruiker zijn/haar gegevens hoeft te veranderen en zonder dat andere toepassingen hier last van hebben. Ook wordt het voor hacker een stuk moeilijker om de gegevens te onderscheppen.

Er bestaat ook nog een *“Two-Legged OAuth”* Hierin word het toestemming vragen aan de gebruiker over geslagen en word er een van te voren geautoriseerde token door gegeven. Denk hierbij bijvoorbeeld aan een widget op een website waar de gebruiker al is ingelogd.

Ook OAuth zal gebruik moeten maken van een SSL-verbinding om er voor te zorgen dat de tokens niet onderschept kunnen worden.

(OAuth Community Site) (OAuth Community Site) (OAuth - Wikipedia, the free encyclopedia, 2013) (RFC6749 - The OAuth 2.0 Authorization Framework, 2012)

4 Aan welke mogelijkheden zijn kosten verbonden en hoe hoog zijn deze?

Alleen aan SSL en Client certificaten (CC) zijn kosten verbonden. Alle andere besproken technieken zijn open en kunnen worden geïmplementeerd.

Bij SSL en Client certificaten zijn certificaten nodig. Wanneer deze zelf gemaakt worden zullen de hedendaagse browsers deze niet herkennen als een geldig certificaat. En zullen gebruikers worden afgeraden om verder te gaan.

Door een geldig certificaat te kopen zal dit niet gebeuren.

Er zijn verschillende typen certificaten. Één is de goedkoopste.

1. Domeinvalidatie: Alleen de gegevens van de domeinhouder worden gecontroleerd
2. Organisatievalidatie: Alle bedrijfsgegevens worden gecontroleerd en opgenomen in het certificaat
3. Extended validatie: Hier worden alle bedrijfsgegevens gecontroleerd, Uittreksels van de kamer van koophandel worden opgevraagd, waarmee aanvrager wordt opgebeld om te controleren of hij/zij het echt heeft aangevraagd.

Met een extended validatie word de balk in een browser ook groen gekleurd.

Afhankelijk van welk type certificaat en welke aanbieder kan het tussen de 15 en 2000 euro⁶ kosten.

⁶ Prijzen kunnen veranderen en zijn afhankelijk van de aanbieders.

5 Welke eisen stelt Apple aan de beveiliging van de passbook kaarten?

Apple stelt op twee gebieden eisen aan de beveiliging.

5.1 De Passbookkaart

De Passbook kaart is een zip bestand dat een aantal dingen bevat:

- Pass.json
- Plaatjes/Logo's
- Manifest.json
- Signature

Pass.json en de plaatjes en logo's verzorgen de functionaliteit en het uiterlijk van de Passbook kaart. Manifest.json en de signature zijn er voor de beveiliging.

5.1.1 Manifest.json

Hierin staan alle SHA1-hashes van alle bestanden in de Passbookkaart. Met uitzondering van de manifest.json en de signature.

5.1.2 Signature

De signature file is een 'PFCS #7 detached signature' van de manifest.json en twee certificaten die van Apple komen. Eén certificaat is persoonlijk en uniek van een ontwikkelaar, het tweede is er één van Apple genaamd het '*Apple Worldwide Developer Relations Certification Authority*'

Nu aan alle eisen is voldaan kan de map gezippt worden met de .pkpass extensie. Als alles klopt zal een iPhone of iPod Touch de kaart herkennen als een geldige Passbookkaart.

5.2 Communicatie met de Passbookkaart

De communicatie van de Passbookkaart en een webservice is ook door Apple vastgelegd. Zo moet er gebruik worden gemaakt van SSL en er moet gecontroleerd worden op de volgende items:

- Kaart ID
- Serienummer
- Autorisatie token

Met het kaart-id en het serienummer moet er gecontroleerd worden of de autorisatie token geldig is en klopt. Als dit niet het geval is, moet er direct worden gestopt met de communicatie.
(Apple Inc.) (Apple Inc.) (Apple Inc.)

6 Met welke mogelijkheid/mogelijkheden is er al ervaring binnen TASS?

Binnen TASS is ervaring met alle genoemde mogelijkheden.

Maar er wordt binnen TASS gewerkt aan het chimera project. In dit project wordt een framework gerealiseerd waarin zaken zoals beveiliging al zitten ingebakken. Hierdoor hoeft het wiel niet telkens opnieuw te worden uitgevonden. Binnen het framework is gekozen om OAuth in combinatie met gebruikersnaam/wachtwoord te gebruiken.

De redenen hiervoor zijn:

1. Alle grote spelers hebben een OAuth provider: Google Facebook, Twitter, Microsoft, Github, etc. het is een goed idee om hierbij aan te haken want de pool van support is hierdoor velen malen groter dan met andere technologieën.
2. Door het Consumer - Provider principe kunnen wij de autorisatie encapsuleren en zelfs uitbesteden aan een derde partij.
3. Het is een open standaard, dat wil zeggen dat er zijn geen kosten aan zijn verbonden om de specificatie te implementeren.
4. Er zitten honderden man-jaren werk in het uitdenken, specificeren, standaardiseren, implementeren en verifiëren van OAuth, dat kunnen wij als TASS niet nadoen.
5. Er zijn ook open source implementaties van de consumers en de providers beschikbaar, wij kunnen deze opnieuw gebruiken.
6. Er is goed nagedacht over de derde partij ontwikkelaar permissie. Ook de strategieën over ontwikkelaar permissies kunnen wij overnemen.

(Hu, 2013)

7 Conclusie

Hoe kan de toegang en verbinding met de Shakies webservice het beste beveiligd worden zodat alleen gemachtigde personen en kaarten toegang krijgen?

Er is gebleken dat alle beveiligingsmethoden hun voor en nadelen hebben. Hierdoor zal de oplossing bestaan uit meerdere methoden. Voor alle verbindingen zal SSL/TLS verplicht zijn, hierdoor kunnen de verstuurde gegevens niet onderschept worden.

Aan de communicatie met de Passbookkaart valt er weinig te doen, Apple specificeert al de complete API en hoe deze beveiligd moet worden. Hieraan kan daarom ook niks veranderd worden.

Voor de communicatie met de medewerkers applicatie moet wel worden bepaalt wat er gebruikt gaat worden. Hiervoor zijn verschillende autorisatiemogelijkheden mogelijk. Er is gekozen om OAuth in combinatie met SSL te gebruiken. Door het gebruik van OAuth vult de gebruiker zijn of haar gegevens alleen in op een beveiligde omgeving waarna alleen nog tokens heen en weer gestuurd worden. Mocht een token gecompromitteerd worden, kan deze direct de toegang worden ontzegd zonder dat iemand anders hier last van heeft. Daarnaast wordt dit binnen het chimera framework ook gebruikt. Naast het gebruik van OAuth zou nog een API token gebruikt kunnen worden om alleen verzoeken met het juiste token toegang tot de webservice te geven.

De aanbeveling zal worden overleg met de klant waarna een beslissing wordt genomen.

8 Literatuurlijst

Apple Inc. (sd). *Getting Started with Passbook on iOS 6*. Opgeroepen op 3 5, 2013, van Apple Developer: https://developer.apple.com/passbook/Getting_Started_with_Passbook.pdf

Apple Inc. (sd). *Passbook for Developers - Apple Developer*. Opgeroepen op 3 5, 2013, van Apple Developer: <https://developer.apple.com/passbook/>

Apple Inc. (sd). *Passbook Programming Guide: About Passbook*. Opgeroepen op 3 5, 2013, van Developer Apple: https://developer.apple.com/library/ios/#documentation/UserExperience/Conceptual/PassKit_PG/Chapters/Introduction.html

AviD. (2010, 12 5). *appsec - Is BASIC-Auth secure if done over HTTPS?* Opgeroepen op 03 1, 2013, van stackexchange: <http://security.stackexchange.com/questions/988/is-basic-auth-secure-if-done-over-https>

Base64 - Wikipedia. (sd). Opgehaald van Wikipedia: <http://nl.wikipedia.org/wiki/Base64>

cryptography - Certificate based authentication vs Username and Password authentication - IT Security Stack Exchange. (2011, 5 6). Opgeroepen op 2 25, 2013, van Stackexchange: <http://security.stackexchange.com/questions/3605/certificate-based-authentication-vs-username-and-password-authentication>

hash code techniek (Betekenis/definitie van). (sd). Opgeroepen op 3 5, 2013, van Encyclo: <http://www.encyclo.nl/begrip/hash%20code%20techniek>

Hu, J. (2013, 3 15). *OAuth - TASS Wiki*. Opgeroepen op 3 16, 2013, van TASS Wiki: <https://www.onderwegmettass.nl/mediawiki/index.php/OAuth>

Jeffreytedjosukmono. (2006, 11 24). *File:Client server handshake.jpg - Wikipedia*. Opgeroepen op 3 3, 2013, van Wikimedia: http://commons.wikimedia.org/wiki/File:Client_server_handshake.jpg

OAuth - Wikipedia, the free encyclopedia. (2013, 5 13). Opgeroepen op 2 19, 2013, van Wikipedia, the free encyclopedia: <http://en.wikipedia.org/wiki/OAuth>

OAuth Community Site. (sd). *OAuth 2.0 - OAuth*. Opgeroepen op 2 19, 2013, van OAuth: <http://oauth.net/2/>

OAuth Community Site. (sd). *OAuth Community Site*. Opgeroepen op 2 18, 2013, van OAuth Community Site: <http://oauth.net>

Reverse engineering - Wikipedia - Wikipedia. (2013, 3 7). Opgeroepen op 3 6, 2013, van Wikipedia: http://nl.wikipedia.org/wiki/Reverse_engineering

RFC6749 - The OAuth 2.0 Authorization Framework. (2012, 10). Opgeroepen op 2 20, 2013, van ETF Tools: <http://tools.ietf.org/html/rfc6749>

Wikipedia. (2013, 4 23). *Application programming interface - Wikipedia*. Opgeroepen op 3 9, 2013, van Wikipedia: http://nl.wikipedia.org/wiki/Application_programming_interface

Wikipedia. (2013, 4 17). *Asymmetrische cryptografie - Wikipedia*. Opgeroepen op 2 23, 2013, van Wikipedia: http://nl.wikipedia.org/wiki/Asymmetrische_cryptografie

Wikipedia. (2013, 5 28). *Basic access authentication* - *Wikipedia*. Opgeroepen op 2 12, 2013, van Wikipedia: http://en.wikipedia.org/wiki/Basic_access_authentication

Wikipedia. (2013, 3 12). *Hashfunctie* - *Wikipedia*. Opgeroepen op 3 25, 2013, van Wikipedia: <http://nl.wikipedia.org/wiki/Hashfunctie>

Wikipedia. (2013, 3 7). *Reverse engineering* - *Wikipedia*. Opgeroepen op 3 9, 2013, van Wikipedia: http://nl.wikipedia.org/wiki/Reverse_engineering

Wikipedia. (2013, 3 11). *Symmetrische cryptografie*. Opgeroepen op 2 24, 2013, van Wikipedia: http://nl.wikipedia.org/wiki/Symmetrische_cryptografie