



"Nothing is impossible"

greeting
voicemail 2.0

Mobiele Applicatie voor MTTM.nl
Voicemail-applicatie Greeting op basis van Salesforce en IVR Manager

12 juni 2013

Opdrachtgever:

MTTM.nl

in persoon dhr. D. de Boer en dhr. ing. C. Gorter

Opdrachtnemer:

T.R. Otter (474791)

Student Informatica (Bachelor of ICT)

Hogeschool INHOLLAND te Haarlem

Begeleider: drs. H. Drillenburg

12 juni 2013

Samenvatting

Vanuit de organisatie MTTM.nl is de vraag gerezen om een mobiele applicatie te ontwikkelen ten behoeve van haar klanten. Deze applicatie is een voicemail-applicatie die op de markt gebracht wordt onder de naam Greetingq.

De app heeft twee hoofdfunctionaliteiten: ten eerste kan de mobiele gebruiker zijn/haar voicemailbox beluisteren. Daarnaast kan de mobiele gebruiker instellingen van zijn voicemailbox aanpassen, zoals dat in de huidige situatie via IVRManager mogelijk is.

Eén van de eisen gesteld aan de applicatie is, dat de achterliggende architectuur uit te lezen is via het door MTTM.nl gebruikte CRM Salesforce.com. Om dit mogelijk te maken is er gebruik gemaakt van een koppeling tussen IVR Manager (telefoniecentrale die de voicemailgesprekken afvangt) en Heroku, een cloud-platform voor (o.a.) Java-applicaties.

Heroku handelt het opslaan van voicemails en het teruggeven van persoonlijke begroetingen af. Vanuit het Heroku-platform wordt de informatie gesynchroniseerd met het Salesforce.com platform.

Om het Greetingq-platform vanuit Salesforce.com te beheren, is gebruik gemaakt van een op maat gemaakte Salesforce.com pagina waarop de Greetingq-gebruikers en de bijbehorende voicemailprompts, voicemails en voicemailinstellingen te zien zijn. Deze Salesforce.com pagina is opgebouwd met de visuele vormgevingstaal VisualForce (onderdeel van Force.com framework).

Vanuit deze pagina wordt met behulp van APEX (Java-achtige programmeertaal, gebruikt voor Salesforce.com-toepassingen) informatie opgehaald uit de Database.com-database (met behulp van de REST-webservice van Database.com) waar de voicemaildata wordt bewaard en verwerkt.

Daarnaast is als proof-of-concept een hybride mobiele applicatie voor Android ontwikkeld met behulp van het Phonegap-framework, en de webprogrammeertalen HTML5, CSS3 en Javascript.

Deze mobiele applicatie voldoet, door het ontwikkelde Greetingq-platform op Heroku, aan de eisen van de opdrachtgever, in de hoedanigheid dat deze communiceert met IVR Manager en de informatie vanuit Salesforce.com te benaderen is.

Ook voldoet de mobiele applicatie aan de eisen van de opdrachtgever op het gebied van functionaliteiten die in de app aanwezig dienen te zijn.

Het Greetingq-platform (Heroku cloud-applicatie) is actief in gebruik genomen door klanten van MTTM.nl. De mobiele applicatie voldoet weliswaar aan de eisen van de opdrachtgever, maar niet aan alle wensen van de opdrachtgever is tijdens de ontwikkeling van dit proof-of-concept voldaan.

Advies aan de opdrachtgever is hierin het proof-of-concept op een aantal onderdelen (should-haves en nice-to-haves) te verbeteren alvorens de mobiele applicatie op de markt te brengen. Tevens adviseert de opdrachtnemer aan de opdrachtgever om vanuit dit proof-of-concept (de mobiele applicatie voor Android) ook een iOS-applicatie (en eventueel Windows Phone en BlackBerry) te laten ontwikkelen om zodoende het grootste deel van de markt aan te spreken.

Inhoudsopgave

Samenvatting.....	3
Voorwoord	8
1. Inleiding.....	9
1.1. Over MTTM.nl	9
1.2. Aanleiding en belang	10
1.3. Doelstelling.....	10
1.4. Eisen en wensen van opdrachtgever	11
1.5. Probleemstelling en onderzoeksvragen	11
1.6. Indeling rapport.....	12
2. Aanpak.....	14
2.1. Literatuuronderzoek	14
2.2. Overige onderzoeksmethoden	14
3. Aanwezige webservices binnen IVR Manager.....	16
3.1. Beschrijving IVR Manager	16
3.2. Webservices van IVR Manager.....	18
3.3. IVR Manager en Greeting	20
3.4. Alternatief voor IVR Manager (Asterisk)	21
4. Aanwezige webservices binnen Salesforce.com	22
4.1. Beschrijving CRM-pakket Salesforce.com	22
4.2. Gebruik van Salesforce.com binnen MTTM.nl	23
4.3. Webservices van Salesforce.com	24
4.3.1. Salesforce.com REST API	24
4.3.2. Salesforce.com SOAP API	25
4.4. Beperkingen van Salesforce.com/Database.com	26
4.5. Overwegingen met betrekking tot implementatie.....	27
5. Voordelen en nadelen van een native applicatie	28
5.1. Voordelen van native ten opzichte van web	28
5.1.1. Native functionaliteit: Notificaties	28
5.1.2. Native functionaliteit: Contacts	29
5.1.3. Snelheid	29
5.1.5. Vormgeving.....	29

5.1.6.	Conclusie ten aanzien van de voordelen.....	29
5.2.	Nadelen van native ten opzichte van web	30
5.2.1.	Ontwikkeltijd.....	30
5.2.2.	Kosten.....	30
5.2.3.	Platform-specifiek.....	30
5.3.	Conclusie ten aanzien van voordelen en nadelen van native app.....	30
6.	Borging van de continuïteit van de applicatie.....	32
6.1.	Veranderingen met betrekking tot Salesforce	32
6.2.	Vastgestelde maatregelen voor borging van de continuïteit van de applicatie.....	32
7.	Ontwerp	34
7.1.	Keuze IVR Manager.....	34
7.2.	Keuze Heroku-platform.....	34
7.3.	Keuze hybrid-applicatie	35
8.	Gebruikte technologieën	36
8.1.	UML voor functioneel en technisch ontwerp	36
8.2.	XML voor uitwisseling tussen IVR Manager en Heroku.....	36
8.3.	Ontwikkeling voor Heroku	37
8.3.1.	Heroku Java (+Git Heroku)	38
8.3.2.	Heroku PostgreSQL database	39
8.3.3.	Eclipse	40
8.4.	Ontwikkeling van mobiele applicatie voor Android	40
8.5.	Ontwikkeling voor CRM Salesforce.com	42
9.	Architectuurbeschrijving	43
9.1.	Overzicht van de te realiseren infrastructuur	43
9.2.	IVR Manager / Massxess.....	44
9.3.	Heroku platform	48
9.4.	Heroku Postgres Cloud Database	49
9.5.	Database.com	50
9.6.	Salesforce.com.....	51
10.	Proof of Concept / Implementatie.....	52
10.1.	Opbouw Mobiele Applicatie Greeting	52
10.2.	Home-scherm / Menu	52

10.3	Voicemailberichten	53
10.3.1.	Berichten bekijken en beluisteren	53
10.3.2.	Berichten downloaden.....	54
10.3.3.	Berichten verwijderen	55
10.3.4.	Berichten filteren	55
10.4	Contactpersonen	56
10.4.1.	Contactpersonen bekijken	56
10.4.2.	Contactpersonen toevoegen.....	56
10.4.3.	Contactpersonen wijzigen.....	57
10.4.4.	Contactpersonen verwijderen	58
10.5	Begroetingen	58
10.5.1.	Begroetingen bekijken en beluisteren	58
10.5.2.	Begroetingen toevoegen.....	59
10.5.3.	Begroetingen verwijderen	60
10.5.4.	Begroetingen downloaden.....	60
10.5.5.	Begroetingen uploaden	61
11.	Validatie Proof of Concept naar eisen en wensen opdrachtgever	62
11.1.	Validatie van must-haves	62
11.1.1.	Koppeling met telefoonboek	62
11.1.2.	Geluidsbestanden afspelen.....	62
11.1.3.	Koppeling met CRM Salesforce.com van MTTM.nl	62
11.1.4.	IVR Manager als infrastructuur	63
11.2.	Validatie van should-haves	63
11.3.	Validatie van nice-to-haves	64
11.4.	Validatie van continuïteitsmaatregelen	65
11.5.	Tussentijdse conclusie	66
12.	Conclusie en aanbevelingen	67
13.	Evaluatie	69
13.1.	Projectaanpak.....	69
13.2.	Projectproces.....	69
13.3.	Onderzoek	69
13.4.	Gerealiseerd product	70

13.5. Reflectie op functioneren opdrachtnemer.....	70
Bijlagen.....	71
Bijlage A. Gesprek met Massxess-engineer Edwin op 7 maart 2013	72
Bijlage B. Samenvatting samenwerkingsbijeenkomst Massxess	73
Bijlage C. Overleg met MTTM.nl over eisen en wensen Greeting	74
Bronvermelding	75

Voorwoord

Graag wil ik David de Boer en ing. Casper Gorter, opdrachtgevers en bedrijfsbegeleiders namens MTTM.nl bedanken voor de kansen en mogelijkheden om zowel het ontwerp en de realisatie van de mobiele voicemail-applicatie Greeting te mogen verzorgen, als het bijbehorende onderzoek uit te mogen voeren.

Daarnaast bedank ik drs. Harald Drillenburg, voor de ondersteuning als afstudeerbegeleider vanuit Hogeschool INHOLLAND te Haarlem.

Tevens wil ik ing. Frank Boon, Software Engineer bij MTTM.nl, bedanken voor de technische ondersteuning tijdens het project Greeting.

Tenslotte (*last but definitely not least*) bedank ik mijn lieve verloofde Layla, die mij tijdens het afstuderen heeft gesteund, en die mij heeft geholpen om door te gaan.

vrijdag 7 juni 2013
Thijs Otter

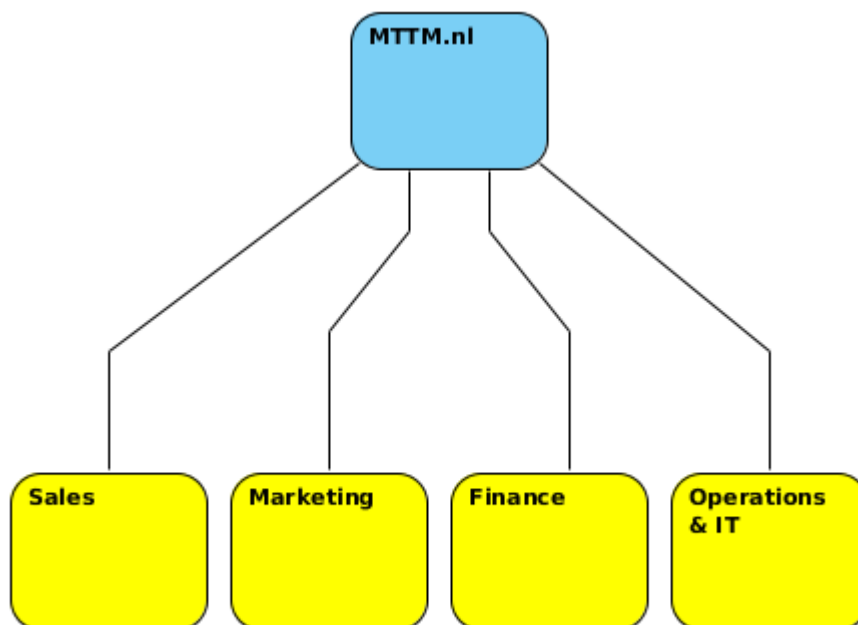
1. Inleiding

Vanuit de telecom-organisatie MTTM.nl (Message To The Moon) is de vraag gerezen om een mobiele applicatie te ontwikkelen ten behoeve van haar klanten. Deze applicatie is een voicemail-applicatie die op de markt gebracht wordt onder de naam Greeting*. Aan opdrachtnemer is gevraagd om deze applicatie te ontwikkelen en hiervoor ook een onderzoek in te stellen. In dit rapport wordt nader ingegaan op de resultaten van het onderzoek.

1.1. Over MTTM.nl

MTTM.nl is een full-service telecom-aanbieder, gespecialiseerd in 0800-, 0900- en 088-servicenummers voor de zakelijke markt. Daarnaast biedt MTTM.nl ook andere zakelijke telecomdiensten aan zoals internetbellen met Hosted VoIP**, en zakelijke ADSL aansluitingen.

MTTM.nl telt ongeveer veertig medewerkers, verdeeld over de afdelingen Sales, Marketing, Operations & IT en Finance. De opdrachtnemer is gedurende 20 weken werkzaam op de afdeling Operations & IT.



Figuur 1.1 Organigram MTTM.nl

*Greeting komt voort uit het Engelse woord 'greeting' (wat begroeting betekent), en kwam oorspronkelijk op de markt als de voicemaildienst van de start-up Greeting B.V. van studenten aan de TU Delft. Greeting B.V. is in 2012 overgenomen door de MTTM Group en wordt door MTTM.nl verder ontwikkeld.

** Hosted VoIP: een vorm van internettelefonie waarbij een traditionele telefoon- of ISDN-centrale is vervangen door een centrale voor internettelefonie die niet meer op locatie van het bedrijf staat, maar nu door de telecomleverancier wordt geëxploiteerd. Via een internetverbinding worden de gesprekken afgeleverd op locatie van het bedrijf.

MTTM.nl (ZakelijkeNummers.nl B.V.) is onderdeel van de MTTM Group, een holdingmaatschappij waarbinnen nog meerdere ondernemingen actief zijn¹. De belangrijkste hierin zijn: ZakelijkeTelefonie.nl B.V., ConnectMeNow, Bedrijfsinfo.nl, TapeCALL.net en Teledeals.nl. De focus van de MTTM Group ligt op telecom.

1.2. Aanleiding en belang

MTTM.nl heeft verschillende producten voor haar klanten beschikbaar gemaakt waarmee de aangeboden telecom-diensten kunnen worden beheerd en uitgebreid. Eén van deze producten is Greeting, een voicemaildienst waarmee persoonlijke boodschappen (voicemailprompts) kunnen worden ingesteld die de beller hoort wanneer men de voicemailbox van deze Greeting-gebruiker inspreekt.

Omdat het beluisteren van de voicemailbox op de mobiele telefoon doorgaans een tijdrovend en inefficiënt proces is, en hierdoor vaak wordt gemeden, is het van belang dat de gebruiker gestimuleerd wordt om het af luisterproces te doorlopen.

Om hierop in te spelen heeft MTTM.nl voorgesteld om een mobiele applicatie te ontwikkelen waarin de voicemailbox beluisterd en visueel geopend kan worden, een zogeheten 'Visual Voicemail'-applicatie.

1.3. Doelstelling

De doelstelling van dit project is het ontwikkelen van een mobiele 'Visual Voicemail'-applicatie op basis van het door MTTM.nl gebruikte CRM Salesforce.com.

De app heeft twee hoofdfunctionaliteiten: ten eerste kan de mobiele gebruiker (die klant is van MTTM.nl) zijn/haar voicemailbox beluisteren en bekijken met behulp van een visuele weergave. Daarnaast kan de mobiele gebruiker instellingen van de voicemaildienst aanpassen.

In de huidige situatie is het enkel mogelijk om de voicemail via e-mail te ontvangen en om de instellingen van de voicemaildienst in het IVR-script van IVR Manager aan te passen.

IVR staat voor Interactive Voice Response, een geautomatiseerd systeem waarmee een interactief menu en/of een routeringsflow kan worden opgezet in een telefoongesprek. IVR Manager is een product van telecom-leverancier Massxess, waarmee de flow van een telefoongesprek (met behulp van een IVR-script) naar wens kan worden ingesteld.

Met de instellingen van de voicemaildienst worden voornamelijk de begroeting (en), de algemene gegevens van de Greeting-gebruiker bedoeld. Hierbij valt te denken aan: welke begroeting er wordt afgespeeld wanneer de voicemailbox wordt gebeld, of naar welk e-mailadres de ingesproken voicemails gemaïld worden.

1.4. Eisen en wensen van opdrachtgever

Vanuit de directie van MTTM.nl zijn de volgende eisen (must-haves) gedefiniëerd voor de (mobiele) applicatie:

- 1. De mobiele applicatie dient gebruik te maken van een koppeling met het telefoonboek (Contacts) om telefoonnummers van binnengekomen voicemailberichten te herkennen.*
- 2. De (mobiele) applicatie dient gebruik te maken van de functionaliteit om geluidsbestanden af te spelen zodat de gebruiker van de mobiele applicatie de voicemailberichten kan afluisteren.*
- 3. De mobiele applicatie dient gekoppeld te zijn met het Salesforce.com CRM van MTTM.nl.*
- 4. De Greeting-dienst dient te werken met IVR Manager (van Massxess) als onderliggende infrastructuur.*

Tevens zijn de volgende wensen (should-haves) gedefiniëerd:

- 5. De mobiele applicatie dient tijdens installatie een doorschakeling in te stellen voor het Greeting-nummer.*
- 6. De mobiele applicatie dient te beschikken over de mogelijkheid om contacten in Greeting te importeren vanuit LinkedIn en Outlook.*

Tenslotte zijn een aantal nice-to-haves gedefiniëerd:

- 7. De mobiele applicatie dient gebruik te maken van de functionaliteit van het opnemen van geluidsberichten ten behoeve van het inspreken van een nieuwe voicemailprompt (persoonlijke begroeting).*
- 8. De mobiele applicatie dient gebruik te maken van notificaties om de gebruiker op de hoogte te stellen van inkomende voicemailberichten.*

De resultaten van het onderzoek leiden tot een beschrijving van de infrastructuur waarin de eisen en wensen van de opdrachtgever zoveel mogelijk dienen te worden meegenomen.

1.5. Probleemstelling en onderzoeksvragen

In hoofdstuk 1.2 (Aanleiding en belang) en hoofdstuk 1.3 (Doelstelling) is naar voren gekomen dat er een mobiele applicatie dient te worden ontwikkeld en dat voor het implementeren van de juiste functionaliteiten, verder onderzoek vereist is. Hierbij is tot de volgende probleemstelling gekomen:

“Hoe kan een mobiele 'visual voicemail' applicatie worden ontwikkeld volgens de eisen en wensen van de opdrachtgever?”

Aan de hand van deze probleemstelling zijn enkele deelvragen bepaald om de verschillende invalshoeken van de probleemstelling nader uit te kunnen werken. De deelvragen zijn geformuleerd tot de volgende:

1. In hoeverre zijn er API's en/of webservices binnen Salesforce aanwezig om data en/of instellingen van de voicemail-dienst Greeting op te halen en op te slaan?

2. Wat zijn de voordelen en nadelen van een native applicatie ten opzichte van een web-app, daarbij rekening houdende met de gewenste functionaliteiten van de voicemail-dienst Greeting?
3. Hoe kan de applicatie zodanig worden opgebouwd dat de continuïteit wordt gewaarborgd, gelet op frequente veranderingen in de Force.com cloud-software?
4. Hoe is de mobiele voicemail-applicatie (Proof of Concept – Android-applicatie) uiteindelijk opgebouwd?
5. Voldoet het 'proof-of-concept' aan de eisen en wensen van de opdrachtgever?

De centrale probleemstelling en de eerstgenoemde drie deelvragen (deelvraag 1 t/m 3) zullen door middel van een literatuuronderzoek worden onderzocht.

Tenslotte zal de vierde deelvraag (Beschrijving Proof of Concept) worden uiteengezet aan de hand van een infrastructuurbeschrijving en een beschrijving van het ontworpen en gerealiseerde bedrijfsproduct.

Uit de vijfde deelvraag volgt een analyse van het gerealiseerde proof-of-concept, waarin wordt vergeleken of de functionaliteit overeenkomt met de gewenste functionaliteiten van de opdrachtgever: de eisen en wensen, zoals beschreven in paragraaf 1.4 (eisen en wensen van de opdrachtgever).

1.6. Indeling rapport

In hoofdstuk 2 (Aanpak) wordt besproken hoe het onderzoek is opgezet en tevens is uitgevoerd. In de daaropvolgende kernhoofdstukken wordt nader uiteengezet welke resultaten er uit het onderzoek voort zijn gevloeid.

In hoofdstuk 3 (IVR Manager) worden zowel het product IVR Manager als de aanwezige webservices binnen IVR Manager beschreven. Ook wordt beschreven in hoeverre IVR Manager geschikt is om de (mobiele) applicatie naar eisen en wensen van de opdrachtgever te kunnen realiseren.

In hoofdstuk 4 (Aanwezige webservices binnen Salesforce.com) wordt uiteengezet wat het CRM Salesforce.com behelst en in hoeverre het CRM gebruikt kan worden om de gewenste functionaliteit op te leveren voor de mobiele applicatie.

In hoofdstuk 5 (Voordelen en nadelen van een native applicatie) wordt, met behulp van de onderzoeksresultaten, toegewerkt naar een conclusie omtrent de voordelen/nadelen van een native mobiele applicatie ten opzichte van een mobiele webapplicatie en of er voor deze specifieke bedrijfssituatie gekozen dient te worden voor een native mobiele applicatie of een mobiele web-applicatie.

In hoofdstuk 6 (Borging van de continuïteit van de applicatie) wordt ingegaan op de relevante veranderingen op het Salesforce-platform die van belang kunnen zijn voor de continuïteit van de Greeting-applicatie.

In hoofdstuk 7 (Ontwerp) worden de belangrijkste keuzes tijdens het ontwikkelproces van het proof-of-concept toegelicht. Tevens wordt uiteengezet welke beslissingen hebben geleid tot het uiteindelijke proof-of-concept.

Hoofdstuk 8 (Gebruikte technologieën) omvat een beschrijving van de methoden en technieken die zijn toegepast voor, tijdens en na de ontwikkeling van het bedrijfsproduct (mobiele app).

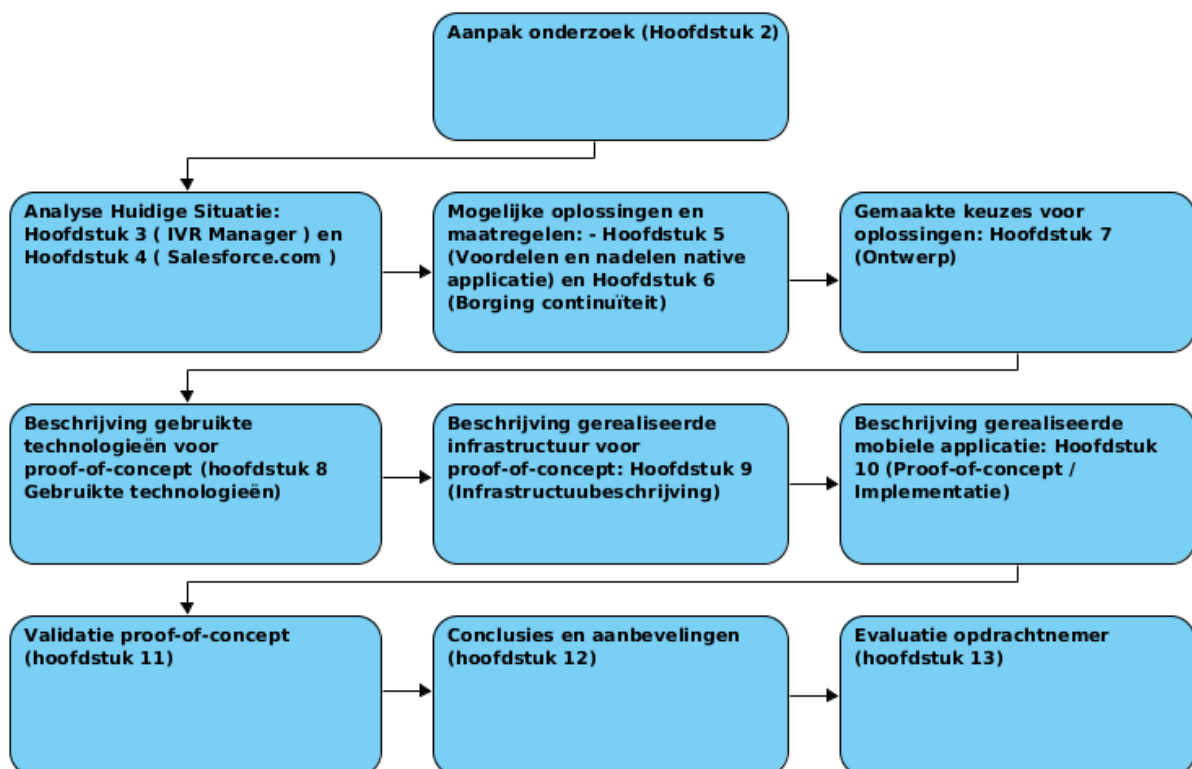
Hoofdstuk 9 (Architectuurbeschrijving) beschrijft de architectuur zoals deze dient te worden ingericht ten behoeve van het correct opleveren van een werkende applicatie is ingericht.

In hoofdstuk 10 (Proof of Concept) wordt het 'proof of concept', een eerste versie van de mobiele applicatie, uiteengezet.

In het laatste kernhoofdstuk, hoofdstuk 11 (Validatie proof of Concept) wordt het proof of concept getoetst aan de wensen en de eisen van de opdrachtgever.

In de conclusie (hoofdstuk 12 van dit rapport) wordt een antwoord gegeven op de probleemstelling, aan de hand van de verzamelde resultaten in de kernhoofdstukken (hoofdstuk 3 tot en met hoofdstuk 11).

In figuur 1.1 is de opbouw van het rapport schematisch weergegeven.



Figuur 1.1 Schematische weergave opbouw rapport

2. Aanpak

In dit hoofdstuk wordt nader uiteengezet hoe tot de gevonden onderzoeksresultaten is gekomen op basis van de eerder bepaalde onderzoeksvragen. Om te komen tot de onderzoeksvragen is gebruik gemaakt van literatuuronderzoek en overig onderzoek (daaronder inbegrepen, maar niet gelimiteerd tot peer-to-peer-research, code review en interviews). In de volgende paragrafen worden respectievelijk het literatuuronderzoek en overig onderzoek toegelicht.

2.1. Literatuuronderzoek

Voor het literatuuronderzoek (ten behoeve van de eerste drie deelvragen) is er voornamelijk gebruik gemaakt van de heersende theorie op het gebied van HCI en design voor mobiel, hierbij is gebruik gemaakt van boeken op het gebied van ontwerp en realisatie op het gebied van mobiele applicaties².

Er is ook veelvuldig gebruik gemaakt van de gangbare bronnen voor ontwikkelaars, veelal developer-websites zoals developer.android.com en stackoverflow.com.

Voor het ontwerpen van de diverse applicatie-flows, klasse-diagrammen, ERD's is gebruik gemaakt van de modelleertaal UML³.

Daarnaast is er gebruik gemaakt van de beschikbare literatuur op het gebied van het CRM-softwarepakket Salesforce.com, daarbij voornamelijk uitgaande van de officiële documentatie uitgebracht door Salesforce.com. Veel van deze documentatie is via internet beschikbaar via developer.force.com.

Daarnaast heeft Salesforce.com literatuur uitgebracht, hierbij valt te denken aan onder andere het boek 'Force.com Platform: Fundamentals'.

2.2. Overige onderzoeksmethoden

Verder zijn de volgende overige onderzoeksmethoden toegepast, met betrekking tot de vierde en vijfde deelvraag:

Bijeenkomst met de opdrachtgever

In deze bijeenkomst (bron: samenvatting bijeenkomst opdrachtgever, zie bijlage C) zijn de eisen en wensen van de opdrachtgever in kaart gebracht. In overleg met de directie zijn de eisen en wensen opgesteld en geprioriteerd.

Semi-gestructureerd interview met Massxess Engineer

Semi-gestructureerd interview met Massxess Operations Engineer Edwin omtrent werking van IVR Manager en XML en omtrent werking van achterliggende infrastructuur achter IVR Manager (het telefonie-netwerk van Massxess). De details van dit semi-gestructureerde interview zijn terug te vinden in bijlage A.

Samenwerkingsbijeenkomst bij telecomleverancier Massxess

In de vorm van een (ongestructureerde) groepsdiscussie en aanvullende presentaties over de mogelijkheden van het telefonie-netwerk van Massxess, is er door zowel MTTM.nl als door Massxess inzicht verkregen in aanwezige/bestaande software-oplossingen waarmee onder andere de voicemaildienst Greeting gerealiseerd kan worden. Een samenvatting van deze discussie is te vinden in bijlage B.

Code review bij collega's

De code reviews bij collega-programmeurs zijn enerzijds aan te merken als inspiratie en anderzijds aan te merken als leermoment voor de opdrachtnemer om inzicht te krijgen in de (bedrijfs)processen binnen MTTM.nl.

2.3 Gebruikte ontwikkelmethode

In overleg met de opdrachtgever heeft opdrachtnemer gekozen om de mobiele applicatie (het proof-of-concept) te ontwikkelen op basis van een 'agile' ontwikkelmethode. Vanuit de opleiding Informatica wordt zowel SCRUM als RUP als 'agile' ontwikkelmethode onderwezen.

Opdrachtnemer heeft aangegeven dit project uit te willen voeren op basis van 'agile development', door in korte 'sprints' of 'iteraties' te werken, waardoor op korte termijn al basisfunctionaliteit kan worden opgeleverd.

Vanuit de opdrachtgever is enkele malen getracht om SCRUM als ontwikkelmethode te integreren binnen de IT-afdeling. In praktijk bleek de SCRUM-aanpak niet te worden gebruikt, maar het ontwikkelen op basis van iteraties (korte sprints) is wel binnen de organisatie als 'best-practice' gebleven.

De opdrachtnemer heeft besloten tijdens de realisatie zo dicht mogelijk bij de best practices te blijven, die reeds binnen het bedrijf worden toegepast. Zodoende zal het proof-of-concept op een iteratieve ontwikkelwijze gaandeweg meer functionaliteiten bevatten teneinde een volwaardig proof-of-concept te ontwikkelen.

3. Aanwezige webservices binnen IVR Manager

In dit hoofdstuk wordt het product IVR-manager beschreven en de aanwezige webservices van IVR Manager uiteengezet.

3.1. Beschrijving IVR Manager

IVR Manager is een IVR-softwarepakket voor Windows, ontwikkeld door telecom-leverancier Massxess. Met IVR Manager is het mogelijk om de flow van een telefoongesprek van begin tot eind te configureren.

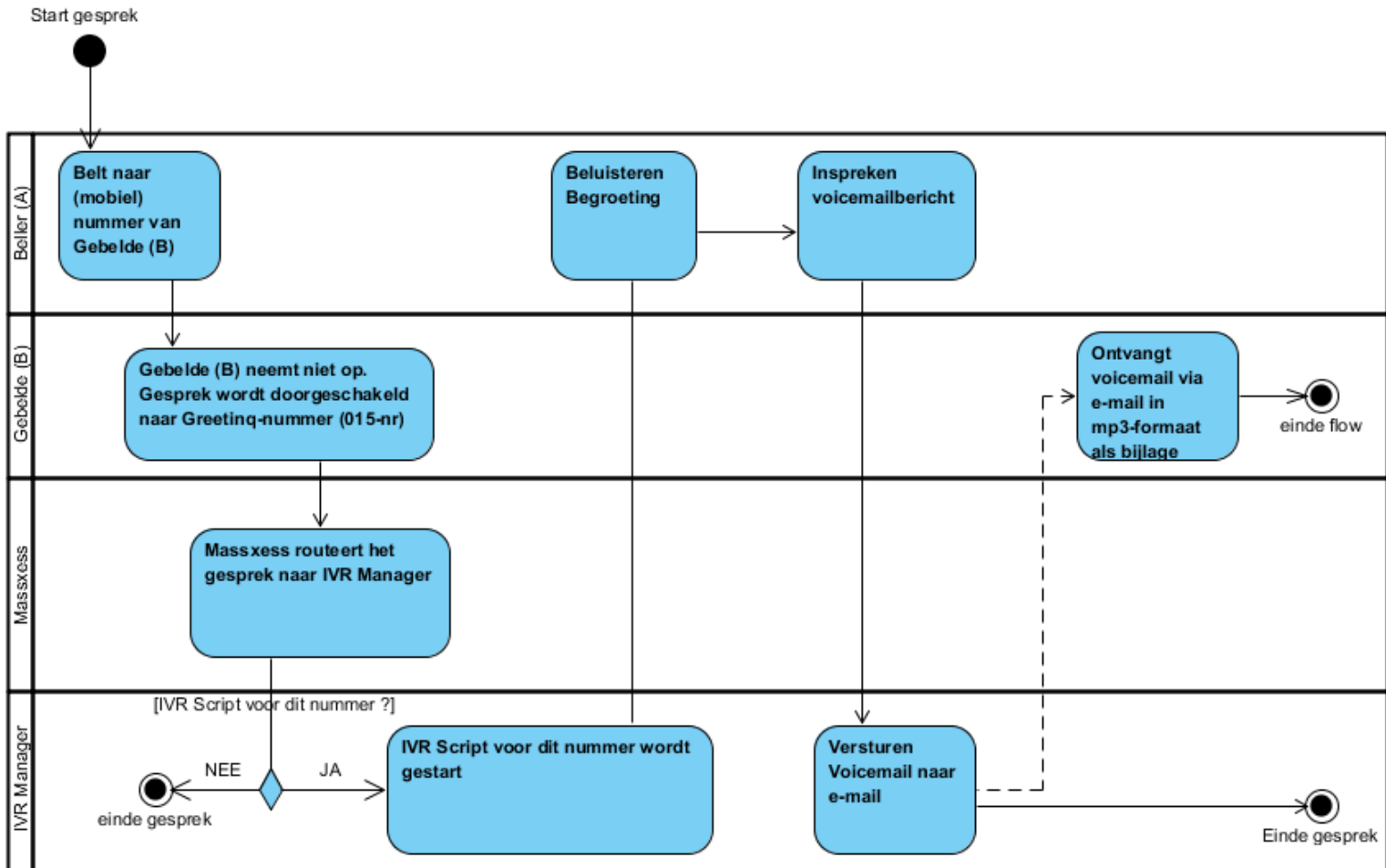
Hierbij valt te denken aan het opnemen van (een gedeelte) van het gesprek of de beller laten kiezen aan de hand van een nummergestuurd menu (Bijv. 'Kies 1 voor afdeling Sales, Kies 2 voor afdeling Support...'), of de beller door te sturen naar een bepaald nummer aan de hand van (een gedeelte) van zijn telefoonnummer. Een voorbeeld van een telefoonflow is te zien in Figuur 3.1.



Figuur 3.1 Voorbeeld IVR-script (flow van een telefoongesprek) in IVR Manager

IVR Manager draait op een stabiel netwerk en wordt door vele honderden Nederlandse ondernemingen gebruikt om het telefonie-verkeer mee te stroomlijnen.

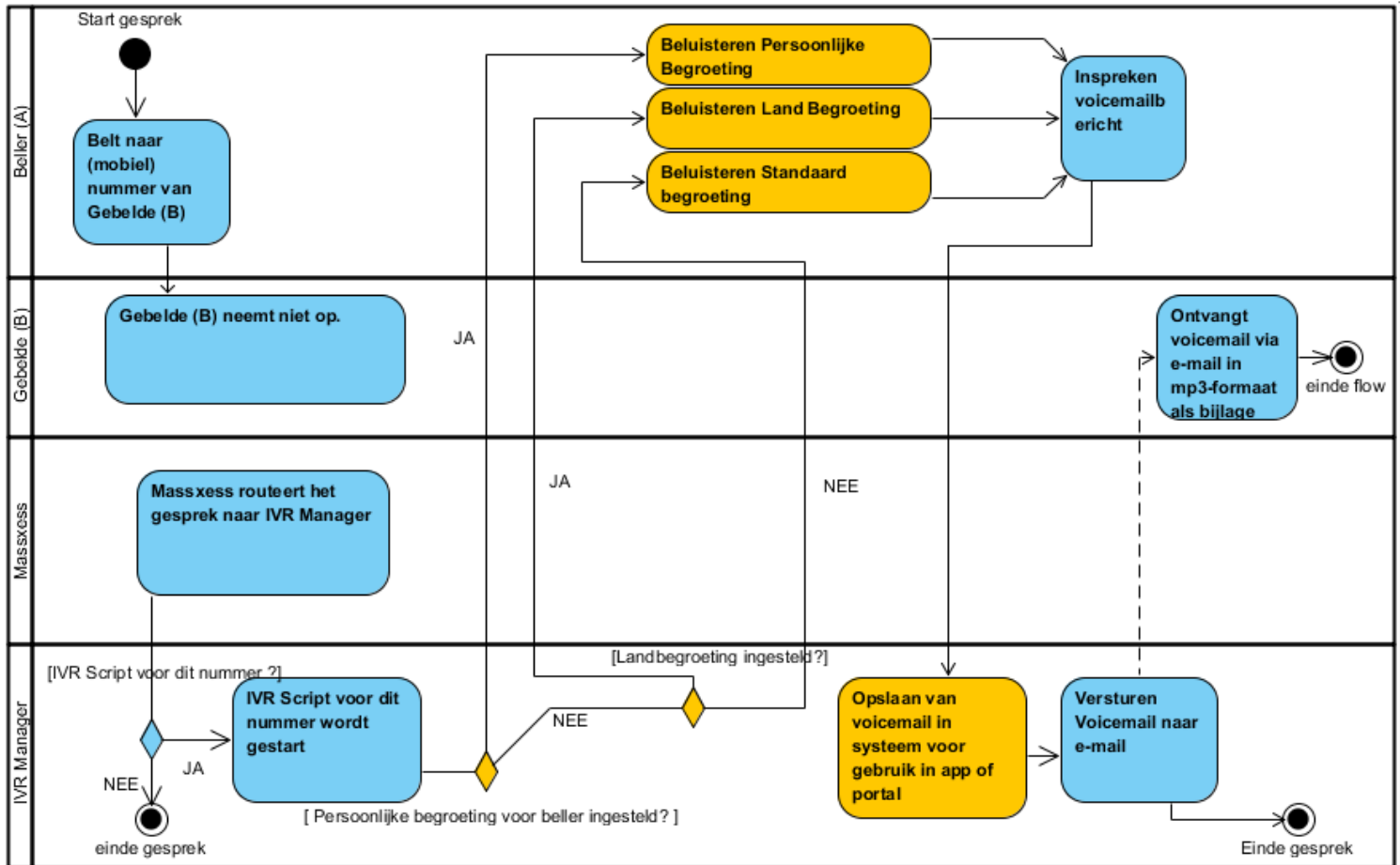
Voor de Greetingq-dienst wordt IVR Manager gebruikt om de inkomende telefonie voor een voicemailnummer af te handelen. In figuur 3.2 is weergegeven hoe het voicemailproces van Greeting in zijn werk gaat.



Figuur 3.2 Voicemailproces Greeting – huidige situatie (UML Activity Diagram)

Zoals uit gesprekken met leverancier Massxess naar voren is gekomen⁴, is IVR Manager in de huidige situatie niet ingesteld om persoonlijke voicemailprompts per beller af te spelen en is het nog niet mogelijk om het opgenomen voicemailbericht via een mobiele applicatie of een webapplicatie aan te bieden (met andere woorden: anders dan via e-mail) aan te bieden aan de Greetingq-gebruiker.

In de gewenste situatie is het mogelijk om via een mobiele applicatie of een webapplicatie (online klantenportal) de binnengekomen voicemailberichten te bekijken, te beluisteren en te verwijderen. In figuur 3.3 is te zien hoe het voicemailproces in de gewenste situatie functioneert.



Figuur 3.3 Voicemailproces Greeting – gewenste situatie (UML Activity Diagram)

3.2. Webservices van IVR Manager

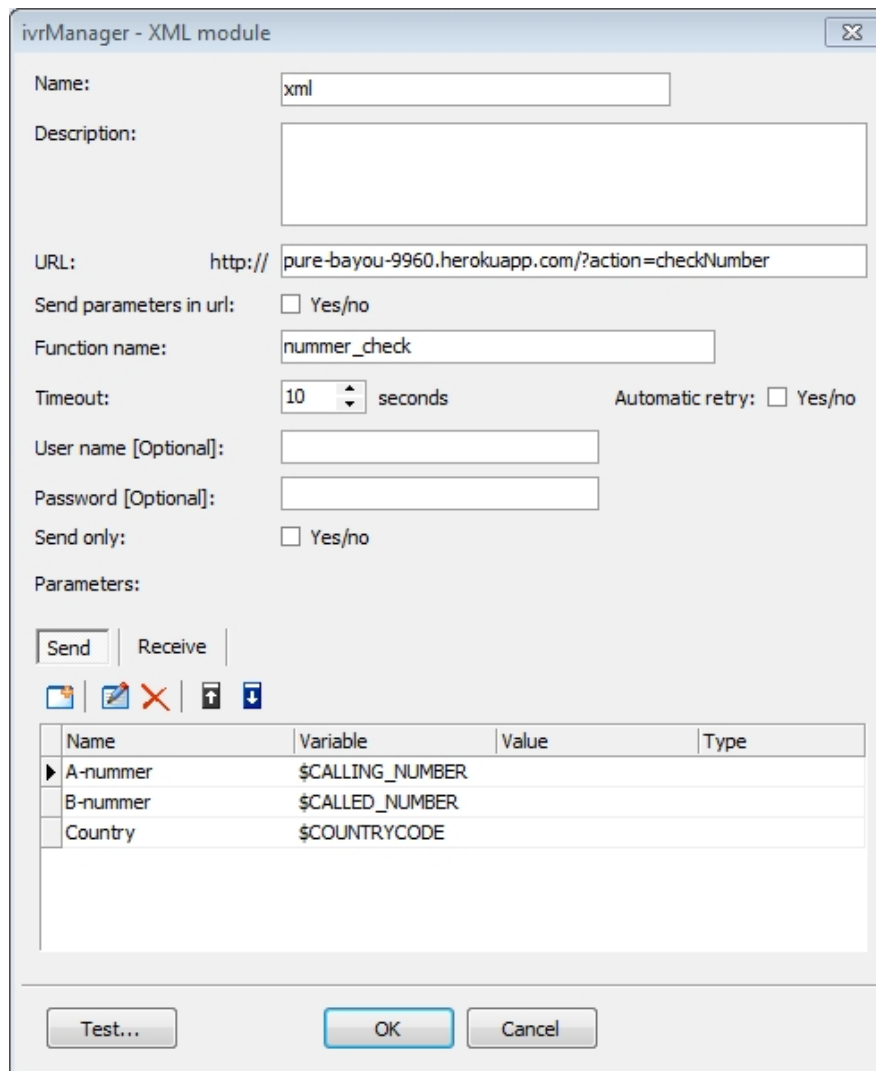
Om het extern aanspreken van de IVR Manager mogelijk te maken opdat het uitlezen/opslaan vanuit Salesforce en de mobiele applicatie mogelijk wordt, is er in overleg* met leverancier Massxess onderzocht in hoeverre de IVR-manager dit ondersteunt.

Daarbij is gekomen tot de volgende resultaten:

- IVR-manager biedt geen ondersteunende bestandsopslag waarmee de voicemailberichten opgehaald kunnen worden
- IVR-manager biedt geen ondersteunende bestandsopslag aan waarmee de persoonlijke voicemailprompts opgeslagen kunnen worden
- IVR-manager biedt geen mogelijkheid aan om via een webservice de reeds opgeslagen voicemails terug te halen
- IVR-manager biedt geen mogelijkheid aan om via een webservice voicemailprompts aan te bieden om deze later door het voicemailsript te laten afspelen

* Gesprek met Massxess-engineer Edwin heeft plaatsgevonden op 07-03-2013. Gespreksverslag hiervan is te vinden in bijlage A.

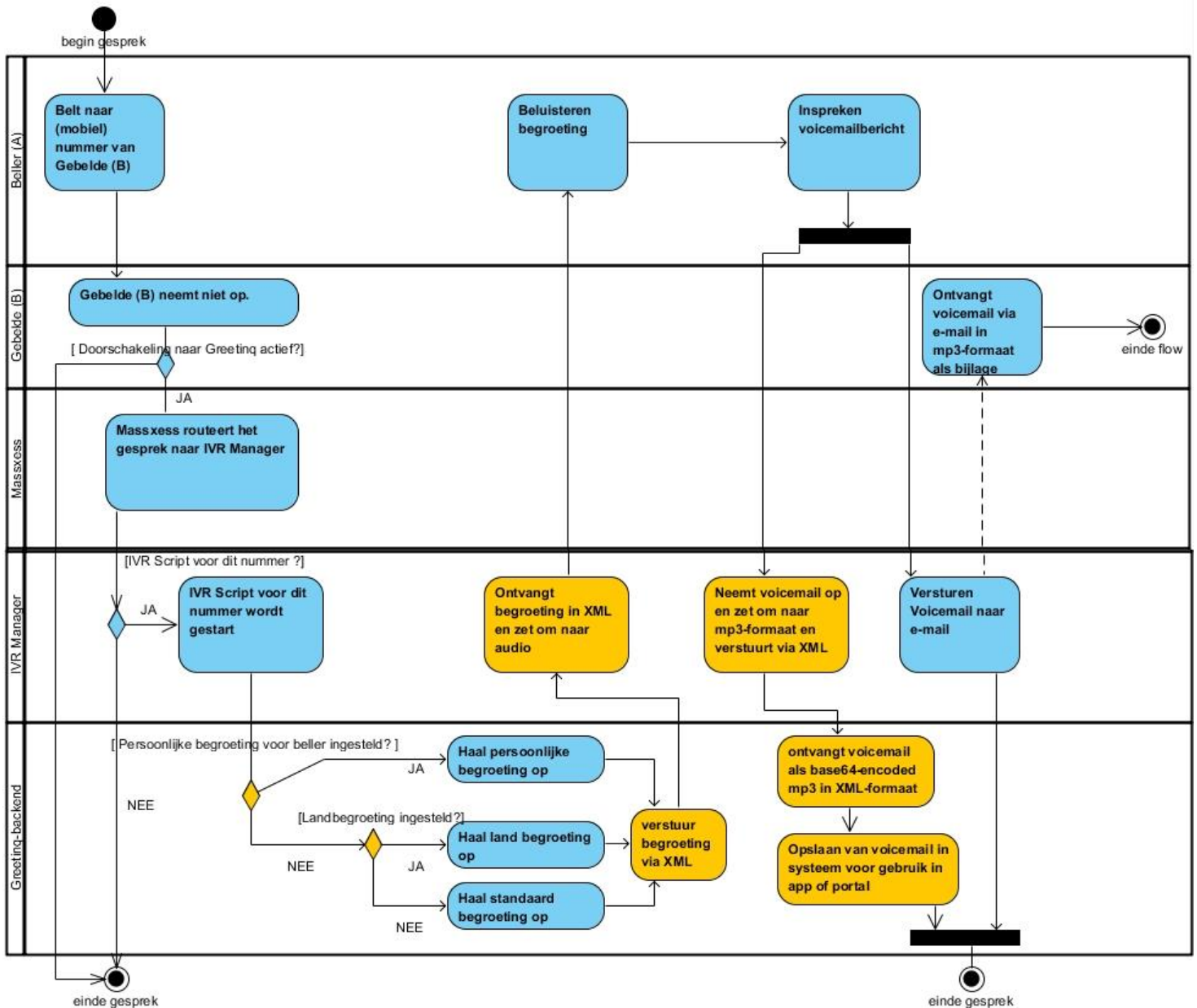
In overleg met de IT-afdeling van Massxess, is gekeken welke mogelijkheden er zijn om toch de gewenste functionaliteit te kunnen implementeren met de IVR-manager. De IT-afdeling van Massxess gaf aan dat IVR Manager wel de mogelijkheid heeft om XML te versturen en te ontvangen, en met de variabelen uit deze input kan werken.



Figuur 3.4 Instellen van IVR Manager om te werken met behulp van XML berichtgeving

In figuur 3.4 is te zien hoe IVR Manager kan worden ingesteld om op basis van deze variabelen te kunnen schakelen tussen verschillende voicemail-prompts.

Langs deze weg is het mogelijk om via XML een voicemailprompt toe te sturen, iedere keer dat er iemand belt. Indien dit verwerkt wordt in de flow van de gewenste situatie, dan komt men tot de volgende uitgewerkte workflow in figuur 3.5.



Figuur 3.5 Voicemailproces Greeting – gewenste situatie met behulp van XML berichtgeving (UML Activity Diagram)

3.3. IVR Manager en Greeting

Zoals in paragraaf 3.1 en 3.2 is beschreven, is de IVR Manager in de huidige situatie (configuratie) niet geschikt voor het afhandelen van de dynamisch ingestelde voicemailprompts van de Greeting-gebruiker.

Hiervoor zou de eerder besproken oplossing van XML-transacties tijdens het voicemailgesprek geïmplementeerd kunnen worden (zoals weergegeven in figuur 3.4).

In paragraaf 8.4 (IVR Manager en Massxess) van hoofdstuk 8 (Gebruikte technologieën) is nader beschreven welke configuratie kan worden toegepast om het gewenste resultaat te bereiken.

3.4. Alternatief voor IVR Manager (Asterisk)

Vanuit MTTM.nl is een must-have van de ontwikkeling dat de Greeting-dienst dient te werken met IVR Manager (van Massxess) als onderliggende infrastructuur. Desondanks heeft opdrachtnemer onderzocht of er vruchtbare alternatieven zijn voor IVR Manager, gezien de beperkte mogelijkheden van IVR Manager.

Na een korte voorselectie door opdrachtnemer is besloten om het pakket Asterisk te onderzoeken als alternatief voor IVR Manager. Asterisk is dé meest gebruikte open-source telefoniecentrale voor IVR doeleinden.

Asterisk heeft minimaal de functionaliteiten van IVR Manager en daarnaast zijn er nog veel meer uitbreidingen mogelijk om een compleet IVR-systeem in te richten.

Asterisk maakt het mogelijk om iedere computer, die geschikt is voor het besturingssysteem Linux, in te richten als telefoniecentrale. Na installatie kunnen extensions worden aangemaakt (bijv. 101 voor Sales, en 102 voor Support), die vervolgens op een VoIP-toestel ingesteld kunnen worden. Naast het instellen van extensions kunnen er ook IVR-menu's, wachtrijen, en opslag van ingesproken berichten geconfigureerd worden.

Door opdrachtnemer is gekozen voor het implementeren van de oplossing op basis van IVR Manager, om aan de eis van de opdrachtgever te voldoen, maar heeft opdrachtgever geadviseerd dat IVR Manager niet het enige product is, wat als telefoniecentrale zou kunnen dienen in deze situatie. Asterisk, draaiend op een Linux-server, zou hiervoor dus tevens geschikt zijn.

4. Aanwezige webservices binnen Salesforce.com

In dit hoofdstuk wordt het product Salesforce beschreven en de aanwezige webservices van dit product uiteengezet. Het doel van het in kaart brengen van de webservices van het Salesforce.com CRM is te bepalen in hoeverre er een directe koppeling mogelijk is met IVR Manager om Salesforce.com de binnenkomende voicemailgesprekken te laten afhandelen.

4.1. Beschrijving CRM-pakket Salesforce.com

Salesforce.com is een software-pakket voor Customer Relationship Management (CRM) waarmee klanten, leads/opportunities geadmineerd kunnen worden.

Salesforce.com is een cloud-CRM, waarbij het woord 'cloud' in dit geval aangeeft dat het gehele CRM-pakket online draait (in de browser), op (virtuele) servers die bij een eventuele storing automatisch het verkeer routeren naar een andere server, zodat er nagenoeg direct doorgewerkt kan worden.

Een unieke mogelijkheid aan Salesforce.com is het werken met Custom Objects waarmee de functionaliteit van het CRM-systeem door de CRM-gebruiker kan worden uitgebreid.

Een voorbeeld van een Custom Object is bijvoorbeeld het custom object 'Leenauto', die door MTTM.nl aan Salesforce is toegevoegd. In figuur 4.1 is het object 'Leenauto' te zien.

Custom Object
Leenauto

[Standard Fields \[4\]](#) | [Custom Fields & Relationships \[5\]](#) | [Validation Rules \[0\]](#) | [Page Layouts \[1\]](#) | [Field Sets \(BETA\) \[0\]](#) | :

Custom Object Definition Detail

[Edit](#) [Delete](#)

Singular Label	Leenauto
Plural Label	Leenauto's
Object Name	Leen_auto
API Name	Leen_auto__c
<hr/>	
Created By	Casper Gorter , 23-2-2012 14:43

Standard Fields

Action	Field Label	Field Name
	Created By	CreatedBy
	Last Modified By	LastModifiedBy
Edit	Naam van Leen auto	Name
Edit	Owner	Owner

Custom Fields & Relationships

[New](#) [Field Dependencies](#)

Action	Field Label	API Name	Data Type
Edit Del	Account	Account__c	Lookup(Account)
Edit Del	adres	adres__c	Formula (Text)
Edit Del	Datum-tijd	Datum_tijd__c	Date/Time
Edit Del Replace	Kenteken	Kenteken__c	Picklist
Edit Del Replace	Status	Status__c	Picklist

Figuur 4.1 Custom Object 'Leenauto'

Zoals in figuur 4.1 te zien is, zijn er een aantal Custom Fields aangemaakt (Account – bijbehorende klant, adres – waar staat de auto geparkeerd?, Kenteken – om welke auto gaat het hier?) op dit object, die per Leenauto mee te geven zijn.

Vanuit de API/webservice kan dit object worden benaderd door het object 'Leen_auto__c' aan te spreken. Hierdoor hoeft het object zelf niet geprogrammeerd te worden binnen de programmeeromgeving. Het object is na aanmaken direct beschikbaar en opvraagbaar.

4.2. Gebruik van Salesforce.com binnen MTTM.nl

MTTM streeft ernaar om zoveel mogelijk informatie in één centraal systeem vast te leggen. Enkele jaren geleden is daarom gekozen voor Salesforce.com vanwege de vele mogelijkheden om alle bedrijfsinformatie in het systeem vast te leggen.

Vanuit het bedrijfsbeleid volgt een harde eis (zie ook eisen en wensen opdrachtgever in paragraaf 1.4) om alle informatie omtrent Greeting in het CRM op te slaan. Dit houdt in dat alle informatie omtrent Greeting-accounts, Greeting-voicemails, Greeting-begroetingen, Greeting-abonnementen dient te worden vastgelegd in Salesforce.com Custom Objecten.

Om ervoor te zorgen dat deze Greeting-objecten gevuld, bijgewerkt en uitgelezen kunnen worden vanuit een extern systeem, beschikt Salesforce.com over een aantal ingebouwde webservices voor het extern aanspreken van de inhoud van het CRM. In de volgende paragraaf wordt hier nader op ingegaan.

4.3. Webservices van Salesforce.com

Salesforce.com biedt standaard meerdere webservices⁵ aan om gegevens op te halen en bij te werken vanuit externe systemen:

- Salesforce.com REST API
- Salesforce.com SOAP API

In paragraaf 4.3.1 wordt toegelicht wat de kenmerken zijn van de REST API (webservice). Daarna volgt in paragraaf 4.3.2 een toelichting van de SOAP API.

Zowel met de REST API als met de SOAP API wordt de informatie uit het CRM opgehaald met behulp van SOQL (Salesforce.com Object Query Language), een querytaal die qua syntax sterk verwantschap houdt met SQL (Structured Query Language).

SOQL is ontwikkeld door Salesforce.com voor de ontwikkelaars die applicaties ontwikkelen voor het Salesforce.com-platform. SOQL werkt met de Objects zoals die ook in het CRM Salesforce.com aanwezig zijn. SOQL beschikt zelfs over de functionaliteit om queries (opdrachten) uit te voeren op Custom Objects. In 4.3.1 en 4.3.2 worden voorbeelden gegeven van SOQL-queries bij zowel REST-transacties als SOAP-transacties.

4.3.1. Salesforce.com REST API

De REST-webservice⁶ van Salesforce.com is een HTTP-webservice gebaseerd op REST-principes⁷ waarmee vanuit andere applicaties de Salesforce.com objecten (informatie) opgehaald en gewijzigd kunnen worden.

REST staat voor Representational State Transfer, waarbij REST gebruik maakt van de reeds bestaande infrastructuur van het Hyper Text Transfer Protocol, namelijk de bekende GET, POST, PUT, DELETE transacties.

Een voorbeeld van het ophalen van een object met behulp van REST is, door met behulp van een GET-request te sturen naar:

```
(...)* /services/rest/data/v24.0/query/?q=SELECT Id, Name FROM Account LIMIT 2
```

* (...) = de basis URL van de desbetreffende Salesforce omgeving, bijv. <https://c.eu1.salesforce.com/>

Dit GET-verzoek geeft van het object 'Account' (wat in veel gevallen klant of leverancier betekent), de Id en de Name terug in JSON-formaat:

```
{
  "Account" :
    [
      {
        "Id" : "12345",
        "Name" : "TestBedrijf B.V."
      },
      {
        "Id" : "56789",
        "Name" : "Testbedrijf 2 N.V."
      }
    ]
}
```

Op deze manier kan met behulp van de ingebouwde request-types van het HTTP-protocol (GET, POST, PUT, DELETE) informatie worden opgehaald en bijgewerkt.

4.3.2. *Salesforce.com SOAP API*

De SOAP-webservice⁸ van Salesforce.com maakt het mogelijk om de objecten uit Salesforce op afstand aan te spreken.

SOAP staat voor Simple Object Access Protocol, wat inhoudt dat voor ieder stuk software een individueel toegepaste API beschikbaar is, die specifiek voor dat softwaresysteem ontwikkeld is. SOAP is dus niet universeel in de zin dat met nagenoeg dezelfde programmeercode een ander systeem kan worden aangesproken, voor ieder software-systeem wordt gebruik gemaakt van voor dat systeem ontwikkelde functies en objecten.

Een voorbeeld van het ophalen van het ophalen van informatie met behulp van SOAP is weergegeven in figuur 4.4.

```
SELECT Id, Name, Greeting_ID__c, Voicefile__c FROM Greeting_voicefile__c WHERE Greeting_ID__c = '2' LIMIT 3;
```

Figuur 4.4 Voorbeeld van SOAP-query op het Custom Object 'Greeting_voicefile' met behulp van de Salesforce API.

De __c staat in de query om aan Salesforce aan te geven dat het een Custom Object of een Custom Field betreft.

4.4. Beperkingen van Salesforce.com/Database.com

De grootste uitdaging voor het ontwikkelen van applicaties voor Salesforce.com ligt niet zozeer in de moeilijkheid van de applicatie, maar in het omzeilen van de beperkingen die Salesforce.com hanteert voor haar software-producten.

Salesforce.com heeft deze beperkingen uiteengezet in de Salesforce.com Limits Cheat Sheet⁹. Enkele van toepassing zijnde Salesforce.com-limieten voor de Greetingq-applicatie zijn de volgende:

Salesforce.com Limieten:	Gevolgen voor Greetingq-applicatie:
<p><u>API Query Cursor Limits</u> "A (API-)user can have up to 10 query cursors open at a time. If 10 QueryLocator cursors are open when a client application, logged in as the same user, attempts to open a new one, then the oldest of the 10 cursors is released. If the client application attempts to open the released query cursor, an error results."</p>	<p>Dit houdt in dat wanneer het Greetingq-platform bezig is om de voicemail-berichten van 10 bellers op te slaan, en er komt op datzelfde moment nog één voicemail-bericht binnen om te verwerken, dan zal het uitvoeren van de query van de 1e persoon komen te vervallen, met als gevolg dat deze voicemail niet (of niet juist) wordt opgeslagen. Voor een systeem dat enkele duizenden voicemails per minuut dient te verwerken, is dit limiet onoverkomelijk.</p>
<p><u>Total API Request Limits</u> The following table lists the limits for the total API requests (calls) per 24-hour period for an organization. 1.000 calls per user for Enterprise-edition</p>	<p>Dit houdt in dat wanneer het Greetingq-platform bezig is om te synchroniseren en er worden bijvoorbeeld 1000 gebruikers gesynchroniseerd, met 10 nieuwe voicemailberichten, en 20 nieuwe contactpersonen, dan zal het aantal calls ongeveer $1000 * 10 + 1000 * 20$ zijn. Dit komt neer op 30.000 calls per synchronisatie (dagelijks). Het limiet voor de MTTM Salesforce ligt rond de 40.000, voor alle applicaties die werken met Salesforce, dus dit limiet zal waarschijnlijk bereikt worden. Het limiet kan verhoogd worden, maar alleen door extra gebruikerslicenties aan te schaffen. Dit is dus geen wenselijke situatie.</p>
<p><u>Field Limits</u> A field can contain a maximum of 32,000 characters.</p>	<p>Dit houdt in dat elke eigenschap van een object een maximale waarde van 32.000 karakters/tekens mag bevatten. Dit is zeer onwenselijk voor het opslaan van de voicemailberichten en de begroetingen, die (uitgeschreven met behulp van base64) doorgaans vele malen langer zijn dan 32.000 karakters. Voor het opslaan van voicemails en begroetingen is derhalve in ieder geval een aparte opslag vereist.</p>

Hoewel niet alle limieten die van toepassing zijn, hierboven staan weergegeven, geven deze wel een indicatie van de beperkingen van Salesforce.com. Deze limieten bieden voor de implementatie van de Greetingq-afhandeling teveel beperkingen met zich mee.

4.5. Overwegingen met betrekking tot implementatie

Uit de voorgaande paragrafen is gebleken dat Salesforce.com veel mogelijkheden heeft om informatie van maatwerk-objecten op te slaan. Hieraan zitten echter wel enkele limieten (zoals uitgelicht in de voorgaande paragraaf), die het juist functioneren van de voicemaildienst belemmeren.

De opdrachtnemer heeft hiertoe in overleg met de opdrachtgever besloten om een alternatief te zoeken voor het direct opvragen en opslaan van informatie door IVR Manager in Salesforce.com. Een aanvullende voorwaarde die vanuit de opdrachtgever aan dit alternatief werd gesteld, was dat het alternatief makkelijk schaalbaar dient te zijn, in het geval de voicemaildienst een explosieve groei doormaakt.

Hiertoe is besloten om een applicatie te ontwikkelen die enerzijds de verzoeken (XML-requests) vanuit IVR Manager kan verwerken en anderzijds de informatie tijdelijk kan opslaan om deze vervolgens, binnen de door Salesforce gestelde limieten, te synchroniseren naar Salesforce.com. De applicatie dient in deze als een 'doorgeefluik' voor de gegevens van de voicemaildienst.

De applicatie heeft een eigen database, waarin de gegevens tijdelijk worden opgeslagen alvorens deze worden gesynchroniseerd.

De voicemailbestanden en begroetingen kunnen vanwege de Field Limits niet in Salesforce.com worden opgeslagen. Om deze toch te kunnen bewaren voor later gebruik, wordt de database van deze 'doorgeefluik'-applicatie gebruikt. Zodoende kan vanuit IVR Manager, Salesforce.com, de mobiele applicatie of de portal een voicemail of begroeting opgehaald worden uit de database van deze applicatie, door een webservice van deze applicatie aan te spreken.

Wegens de aanvullende voorwaarde dat deze applicatie schaalbaar dient te zijn, is er door opdrachtnemer voor gekozen om de applicatie als 'cloud'-applicatie te ontwikkelen, hetgeen de schaalbaarheid en de continuïteit van de voicemail-dienst ten goede komt.

In paragraaf 7.2 (Keuze Heroku-platform) wordt dieper ingegaan op de keuze voor het cloud-platform van deze te ontwikkelen applicatie.

5. Voordelen en nadelen van een native applicatie

In dit hoofdstuk wordt ingegaan in op de voordelen en nadelen van een native mobiele applicatie ten opzichte van een mobiele webapplicatie. Een native applicatie kan worden gedefinieerd als een mobiele applicatie die ontwikkeld is voor het specifieke platform waarop de mobiele telefoon draait.

5.1. Voordelen van native ten opzichte van web

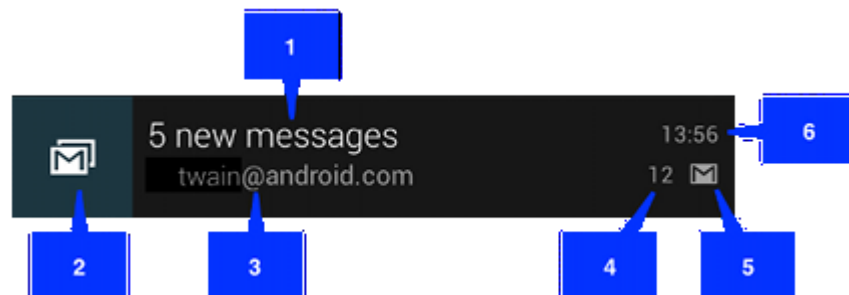
In deze paragraaf wordt nader ingegaan op de gevonden onderzoeksresultaten¹⁰ met betrekking tot de voordelen van een native mobiele applicatie ten opzichte van een mobiele web applicatie. Hierbij is voornamelijk uitgezocht welke voordelen specifiek gelden voor de Greetingq-applicatie.

5.1.1. Native functionaliteit: Notificaties

Eén van de belangrijkste functionaliteiten van een native mobiele applicatie is de beschikking over een notificatie-systeem.

Notificaties voorzien de gebruiker van de telefoon aanvullende informatie omtrent een bepaalde gebeurtenis, ook als de gebruiker de telefoon/desbetreffende applicatie niet binnen handbereik hebben of de telefoon op dat moment niet actief bediend wordt.

Daarnaast wordt er een geluidsfragmentje afgespeeld, vergelijkbaar met het geluid dat wordt afgespeeld wanneer een SMS wordt ontvangen op het toestel. In figuur 5.1 is een voorbeeld te zien van een notificatie op een Android-smartphone.



*Figuur 5.1 Voorbeeld van een Android-notificatie (bron: Android Developer Website¹¹)
1: Titel van notificatie, 2: Notificatie-icoon, 3: korte omschrijving
4: aanvullende omschrijving, 5: klein notificatie-icoon, 6: tijd van notificatie*

Hieruit volgt dat notificaties dus ideaal zijn om de gebruiker te voorzien van nieuws over gebeurtenissen waar de gebruiker aandacht aan dient te besteden. De notificatie verdwijnt nadat de gebruiker deze heeft bekeken of er voor kiest een handeling uit te voeren naar aanleiding van de notificatie.

Vanwege het aandachtstrekken karakter van de notificatie, is deze functionaliteit uitermate geschikt voor de mobiele applicatie Greetingq om de gebruiker op de hoogte te houden van gebeurtenissen (binnenkomende voicemailberichten).

Hoewel de notificatie-functionaliteit oorspronkelijk is ontwikkeld als een native functionaliteit, is het mogelijk met behulp van hybrid-frameworks (zoals Adobe Phonegap, Apache Cordova of XDK¹²) deze native functionaliteit toch te implementeren zonder een native applicatie te bouwen. Met deze mogelijkheid is de Notificaties-functionaliteit dus niet langer een functionaliteit die enkel voorbehouden is aan native applicaties.

5.1.2. Native functionaliteit: Contacts

Een andere belangrijke functionaliteit van een native mobiele applicatie is de beschikking over een koppeling met het interne telefoonboek van de mobiele telefoon vanuit de applicatie. Hiermee kan vanuit de mobiele applicatie een contactpersoon worden benaderd. Bij een webapplicatie is dit niet mogelijk. De browser heeft immers geen toegang tot het interne telefoonboek van de telefoon.

5.1.3. Snelheid

Vanuit verschillende bronnen (bron: "HTML5 vs. Native: The Mobile App Debate", "M. Mahemoff", 03-06-2011, <http://www.html5rocks.com/en/mobile/natedebate/>, bezocht op: 4 juni 2013) is het oordeel dat native mobiele apps over het algemeen sneller werken dan webapps of hybride apps. Vooral bij apps die ingewikkelde berekeningen uitvoeren of bij games, zijn de verschillen duidelijk merkbaar.

5.1.5. Vormgeving

Bij het ontwikkelen van een native app komt de vormgeving overeen met de ontwerprichtlijnen van dat specifieke platform, hierdoor wordt platformconsistentie geborgd en dat komt de gebruikerservaring ten goede. Hierin geldt echter wel dat het de ontwikkelaar weliswaar gemakkelijk gemaakt wordt om de ontwerprichtlijnen te volgen door voorgefabriceerde user interface-elementen, maar het is aan de ontwikkelaar om hier consistent gebruik van te maken.

5.1.6. Conclusie ten aanzien van de voordelen

Uit de voorgaande paragrafen volgen een aantal voordelen, die direct kunnen worden toegepast in de ontwikkeling van de mobiele applicatie, indien wordt gekozen voor een ontwikkeling in de vorm van een native applicatie.

- A. De mobiele applicatie kan gebruik maken van notificaties om de gebruiker op de hoogte te stellen van (nieuwe) ingesproken voicemailberichten.
- B. De mobiele applicatie kan gebruik maken van de koppeling met 'Contacts' om telefoonnummers behorend bij inkomende voicemailberichten te herkennen op basis van de ingebouwde contactpersonenlijst.
- C. De native mobiele applicatie reageert sneller en 'vloeiender' dan een mobiele webapplicatie.
- D. De vormgeving voldoet aan de ontwerprichtlijnen van dat specifieke platform, indien deze richtlijnen door de ontwikkelaar worden gevolgd.

5.2. Nadelen van native ten opzichte van web

In deze paragraaf wordt nader ingegaan op de nadelen van een native-applicatie ten opzichte van een mobiele web-applicatie. Hierbij is voornamelijk uitgezocht welke nadelen specifiek gelden voor de Greeting-applicatie.

5.2.1. Ontwikkeltijd

Aangezien voor ieder platform een aparte programmeertaal wordt gehanteerd en andere ontwikkeltools, randvoorwaarden en eisen zijn vastgesteld door de ontwikkelaars van de mobiele besturingssystemen, kost het vaak meer tijd om voor alle platformen een aparte native applicatie te ontwikkelen met eenzelfde functionaliteit.

5.2.2. Kosten

Voor het ontwikkelen van een native applicatie is meer kennis, vaardigheden en ervaring vereist van dat specifieke platform. Daarnaast kost het ontwikkelen van een native applicatie vaak meer, wegens de vele ontwikkel-uren van software engineers die benodigd zijn.

Voor het ontwikkelen van een native applicatie, zijn naast applicatie-ontwikkelaars ook nog hardware-matige eisen. Voor het ontwikkelen voor een iOS-applicatie dient de ontwikkelaar te beschikken over een Apple MacBook, en een iPhone/iPad of iPod Touch. Tenslotte dient een iOS-ontwikkelaar te beschikken over een Apple Developer Account. Hiervoor dient de opdrachtgever enkele duizenden euro's op jaarbasis te reserveren.

Zo gelden er voor ieder smartphone-platform soortgelijke eisen, en derhalve kosten.

5.2.3. Platform-specifiek

Een native-applicatie is per definitie platform-specifiek. Wordt er bijvoorbeeld een applicatie ontwikkeld voor het Android-besturingssysteem (in Java) dan zal deze niet werken op een iOS-telefoon (iPhone/iPad) omdat deze nou eenmaal niet werken met Java-applicaties, maar met Objective-C applicaties.

Zodoende zou er voor ieder van de huidige smartphone-besturingssystemen een individuele native app ontwikkeld dienen te worden.

Gezien de grote verscheidenheid aan besturingssystemen voor smartphones, te weten iOS (iPhone/iPad/iPod Touch), Android, Windows Phone, BlackBerry, HP webOS, Symbian, Bada en Palm, brengt dit aanzienlijk veel tijd en hoge kosten met zich mee.

5.3. Conclusie ten aanzien van voordelen en nadelen van native app

Wanneer men de voordelen (notificaties, contacts, snelheid en vormgeving) afweegt tegen de nadelen (ontwikkeltijd, kosten en platform-specifiek) lijkt er niet een definitieve oplossing. De opdrachtnemer heeft daartoe besloten in de vergelijking ook nog een hybride app op te nemen. De hybride app is een webapp met native mogelijkheden en dus een combinatie van een native app en

een webapp. In figuur 5.2 is een vergelijkingstabel te zien met de verschillende voor –en nadelen van native, web en hybrid apps. In de laatste rij is door opdrachtnemer een oordeel gegeven in hoeverre de desbetreffende vorm geschikt is voor de realisatie/implementatie van de Greeting-app en de bijbehorende functionaliteiten waarbij 1 staat voor het best implementeerbaar/buikbaar en 3 staat voor het minst buikbaar in deze specifieke situatie.

	Native app	Hybrid app	Web app
A. Mogelijkheid tot gebruik van notificaties	√	√	-
B. Mogelijkheid tot uitlezen van Contacten van het interne telefoonboek	√	√	-
C. Snelheid door hardware-acceleratie	√	-	-
D. Vormgeving naar specifiek platform	√	Tot op zekere hoogte	-
E. Ontwikkeltijd	Lang	Gemiddeld	Kort
F. Kosten	Hoog	Laag	Laag
G. Cross-platform	-	√	√
Ervaring opdrachtnemer met ontwikkeling	Gemiddeld	Veel	Veel
Oordeel ten aanzien van implementatie	③	①	②

In overleg met de opdrachtgever heeft opdrachtnemer gekozen voor de realisatie van de mobiele applicatie in de vorm van een hybrid app. In hoofdstuk 7.3 en hoofdstuk 10 wordt verder ingegaan op de ontwikkeling van deze hybride mobiele applicatie.

6. Borging van de continuïteit van de applicatie

In dit hoofdstuk wordt ingegaan op de vraag in hoeverre de continuïteit* van de applicatie geborgd kan worden, gelet op de verwachte veranderingen (updates/upgrades) binnen het Salesforce.com-platform.

6.1. Veranderingen met betrekking tot Salesforce

Salesforce brengt elk kwartaal een nieuwe update uit van het CRM Salesforce.com, hierbij worden vaak vele grote wijzigingen in één keer doorgevoerd.

Alvorens een nieuwe update in productie te zetten, wordt deze eerst, bij wijze van acceptatietest, uitgebracht op de Sandbox-versies van de Salesforce.com klant-omgevingen. Een Sandbox-omgeving is een kopie van de live-omgeving zonder 'productie'-data. Een Sandbox-omgeving heeft hierdoor vaak dezelfde functionaliteit, maar werkt met dummy-data.

De ontwikkelaars van MTTM.nl kunnen deze nieuwe update een aantal maanden lang proberen in de Sandbox-omgeving om te bepalen of er wijzigingen dienen te worden doorgevoerd om de applicaties die zijn gebouwd voor de live-omgeving te laten werken.

Naast deze Sandbox-omgeving is door opdrachtnemer aan de opdrachtgever voorgesteld om een aantal maatregelen op te stellen om de continuïteit van de applicatie te borgen. Deze vastgestelde maatregelen worden in de volgende paragraaf uiteengezet.

6.2. Vastgestelde maatregelen voor borging van de continuïteit van de applicatie

Uit de voorgaande paragraaf volgen enkele maatregelen die in de applicatie dienen te worden doorgevoerd om de continuïteit van de applicatie te borgen:

- De continuïteit van de applicatie kan worden geborgd door de applicatie uitvoerig te documenteren.
- De continuïteit van de applicatie kan worden geborgd door gebruik te maken van progressive enhancement, een techniek die vooral in de webwereld wordt gebruikt om aan te duiden dat er in principe basisfunctionaliteit is die voor alle doelsystemen geschikt is, en wanneer het desbetreffende doelsysteem dit ondersteunt, er aanvullende functionaliteiten mogelijk worden (ingeschakeld worden). Te denken valt hierbij aan: het uploaden van een begroeting in de app als basisfunctionaliteit, en het direct inspreken met behulp van de app als progressive enhancement hiervan.
- De continuïteit van de applicatie kan worden geborgd door gebruik te maken van object-georiënteerde programmeercode, die makkelijk uitbreidbaar, aanpasbaar en onderhoudbaar is.
- De continuïteit van de applicatie kan worden geborgd door gebruik te maken van zogenoemde Exception Handling (foutafhandeling) waarbij de beheerder van een platform direct op de hoogte wordt gesteld indien er een fout optreedt op het platform.

* Continuïteit van de applicatie: de mate waarin de applicatie kan blijven bestaan/draaien als er veranderingen optreden in de onderliggende architectuur of in de koppeling met data-leveranciers.

- De continuïteit van de applicatie kan worden geborgd door pro-actieve controles in te bouwen, die erop letten dat applicatielimiten niet worden overschreden.
- De continuïteit van de applicatie kan worden geborgd door een applicatiebeheerder aan te stellen die zich bezighoudt met veranderingen in de markt en veranderingen omtrent het Salesforce-platform.

7. Ontwerp

In dit hoofdstuk worden de belangrijkste ontwerpkeuzes tijdens het ontwikkelproces nader toegelicht. Tevens wordt beschreven waarom gekozen is voor de oplossingen waarmee het Proof-of-concept is ontwikkeld.

7.1. Keuze IVR Manager

Hoewel de conclusie van de opdrachtnemer uit paragraaf 3.4 (Alternatief voor IVR Manager) aangeeft dat Asterisk wellicht beter geschikt is als telefoniecentrale ten opzichte van IVR Manager om de Greetingq-dienst te realiseren, is vanuit de opdrachtgever meegegeven dat het werken met IVR Manager een vereiste is.

Daarom zal de Greetingq-dienst (en de uiteindelijke mobiele applicatie) dus werken via IVR Manager. Uiteraard zou een omschakeling van IVR Manager naar Asterisk nog steeds mogelijk zijn, met inachtneming van een aantal nader te bepalen arbeidsuren.

IVR Manager verzorgt, in de opzet van het Greetingq-platform, de schakeling tussen het telefonienetwerk en de Greetingq-dienst.

7.2. Keuze Heroku-platform

Vanuit de wensen en de eisen van de opdrachtgever volgt impliciet dat de applicatie stabiel dient te draaien en ook op piekmomenten zal blijven presteren. Derhalve is het wenselijk om hier bij de keuze voor de achterliggende infrastructuur rekening mee te houden.

Een dergelijke situatie, waarbij het piekmoment van een dienst lastig te bepalen is, maar waarbij uptime van groot belang is, vraagt om een platform dat (redelijk) eenvoudig schaalbaar is. De keuze zou, gezien de recente ontwikkelingen in de IT, vallen op een cloud-platform.

Vanuit de organisatie was geen ervaring met cloud-platformen anders dan Salesforce.com aanwezig. Opdrachtnemer heeft vervolgens onderzocht welk platform voor deze toegepaste situatie het best bruikbaar zou zijn en heeft opdrachtgever vervolgens aangeraden de Greetingq backend-applicatie (de schakel tussen telefonie en Salesforce.com) te gaan ontwikkelen op het Heroku cloud-platform.

In hoofdstuk 8.3 (Gebruikte technologieën tijdens ontwikkeling voor Heroku) worden de gebruikte technologieën omtrent Heroku uiteengezet. In hoofdstuk 9 is toegelicht hoe de Greetingq-backend op het Heroku-platform uiteindelijk is opgebouwd.

7.3. Keuze hybrid-applicatie

Vanuit paragraaf 5.3 (Conclusie ten aanzien van voordelen en nadelen van native applicaties) is gekomen tot het oordeel dat de mobiele Greeting-app niet noodzakelijkerwijs als native applicatie ontwikkeld dient te worden, maar dat dit wel vele voordelen biedt. De nadelen, met name de verhoogde ontwikkeltijd en benodigde kennis en middelen, wegen in deze situatie meer op tegen de voordelen.

Opdat de functionaliteiten die bij native applicaties als voordelen werden genoemd behouden blijven in de ontwikkeling, is gekeken naar de mogelijkheden van een mobiel hybrid-framework zoals Adobe Phonegap, Apache Cordova, AppMobi XDK.

Vanuit de opdrachtnemer is aan opdrachtgever geadviseerd om de mobiele applicatie te laten ontwikkelen als een hybride mobiele applicatie, om zowel gebruik te maken van enkele native functionaliteiten (notificaties, contacten, etc.), als de ontwikkeltijd te versnellen de programmeercode universeel bruikbaar te houden. De opdrachtgever heeft aangegeven een hybrid-applicatie een passende en bruikbare oplossing te vinden.

Vanwege persoonlijke ervaring met ontwikkeling voor Android en de hoge kosten voor andere platforms zal het proof-of-concept als Android-applicatie ontwikkeld worden.

In hoofdstuk 8.4 (Gebruikte technologieën) wordt uiteengezet met welke technologieën het proof-of-concept uiteindelijk ontwikkeld wordt. In hoofdstuk 10 (Proof-of-concept / Implementatie) wordt de opzet en werking van de mobiele applicatie toegelicht.

8. Gebruikte technologieën

In dit hoofdstuk worden de gebruikte technologieën beschreven die tijdens de ontwikkeling van de mobiele applicatie en de achterliggende infrastructuur zijn toegepast.

8.1. UML voor functioneel en technisch ontwerp

Voor het opstellen van een functioneel en technisch ontwerp is door opdrachtnemer gebruik gemaakt van UML, Unified Modeling Language, een modelleertaal waarmee onder andere applicaties, applicatiestructuren en bedrijfsprocessen gemodelleerd kunnen worden.



Figuur 8.1 UML: Unified Modelling Language (bron: UML Website¹³)

Voor het opstellen van de flows van de telefoongesprekken, zoals gebruikt in hoofdstuk 3 (IVR Manager) is gebruik gemaakt van het UML Activity Diagram.

Voor het opstellen van een diagram waarin het ontwerp van de object-georiënteerde programmatuur van het Greetingq-platform wordt uiteengezet, is gebruik gemaakt van het UML Class Diagram.

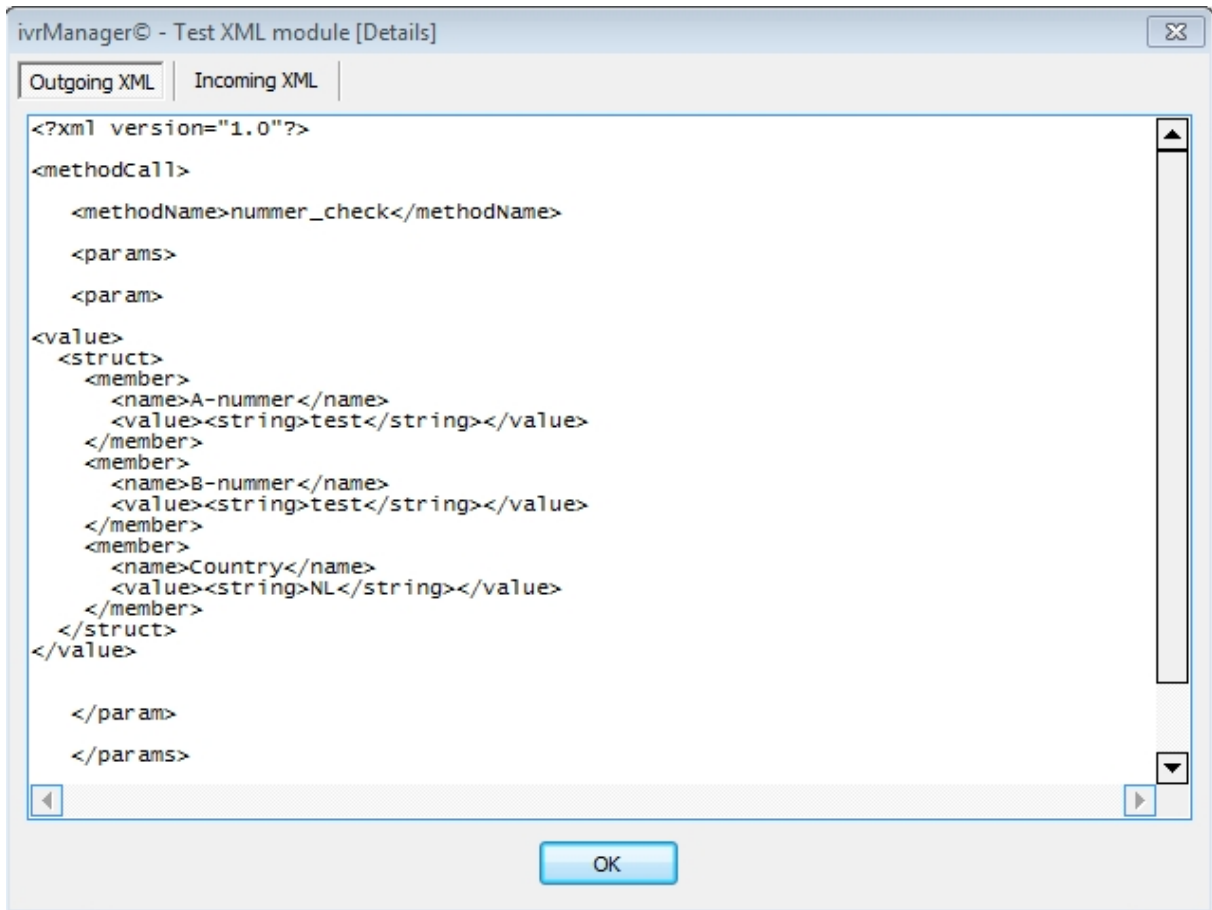
Om gebruikersscenario's te schetsen, is gebruik gemaakt van UML Use Case Diagrams en de daarbij behorende UML Use Cases. Vanuit de opleiding van opdrachtnemer wordt UML als standaard onderwezen. Daarnaast is UML in de software-ontwikkelingsbranche UML de de-facto standaard voor het modelleren van software-ontwerp.

8.2. XML voor uitwisseling tussen IVR Manager en Heroku

Voor de uitwisseling tussen IVR Manager en Heroku wordt gebruik gemaakt van XML-berichtgeving. XML* is een standaard voor het uitwisselen van berichten via (internet)-applicaties. Voor het realiseren van dit proof-of-concept is gekozen voor XML omdat IVR Manager hiervan gebruik maakt voor het inladen en versturen van gegevens van en naar externe systemen.

In figuur 8.2 is een voorbeeld te zien van een XML-bericht dat na het afhandelen van een voicemailgesprek wordt verstuurd naar het Heroku/Greetingq-platform.

*XML: eXtensible Markup Language, een gestandaardiseerde opmaaktaal voor berichtenverkeer



Figuur 8.2 Voorbeeld van een XML-bericht dat door IVR Manager wordt verstuurd

8.3. Ontwikkeling voor Heroku

Voor het Heroku-cloudplatform waarop het Greetingq-platform is ontwikkeld, zijn diverse technologieën en methodieken gebruikt, die in onderstaande paragrafen individueel uiteengezet worden.

In paragraaf 8.3.1 (Heroku Java) wordt ingegaan op het gebruik van de programmeertaal Java op het Heroku-platform en daarnaast wordt de werking beschreven van Git, een version control system (VCS*), dat samenwerkt met Heroku.

In paragraaf 8.3.2 (Heroku PostgreSQL) wordt de achterliggende database van het Greetingq-platform toegelicht in hoe deze aangesproken kan worden. Tenslotte eindigt deze paragraaf met een beschrijving van de Eclipse IDE**.

*VCS: Version Control System, een systeem voor versiebeheer waarmee versies van programmeercode kunnen worden beheerd in een gecentraliseerd systeem. Git is hiervan een van de bekendste voorbeelden.

**IDE: Integrated Development Environment: een ontwikkelomgeving waarin programmeurs programmeercode kunnen schrijven die door de ontwikkelomgeving wordt gecontroleerd op juistheid. Bekende voorbeelden zijn Eclipse (<http://www.eclipse.org/>) en Netbeans (<http://www.netbeans.org/>).

8.3.1. Heroku Java (+Git Heroku)

Het Heroku cloud-platform ondersteunt de programmeertalen Ruby, Node.js, Clojure, Python, Java, Scala en PHP voor het ontwikkelen van Heroku cloud-applicaties.

Voor het Greetingq-platform zou iedere programmeertaal volstaan waarmee een applicatie kan worden ontwikkeld die via XML kan worden aangesproken, en die de te verwerken gegevens kan opslaan en ophalen met behulp van een database. Hiervoor volstaan alle bovengenoemde programmeertalen.

Vanwege de ervaring die zowel opdrachtnemer als opdrachtgever heeft opgedaan met het programmeren in Java, is gekozen om het Greetingq-platform te programmeren in de programmeertaal Java.

Voor het beheren van de versie van de programmeercode van de Greetingq-applicatie wordt gebruik gemaakt van het met Heroku gebundelde Git, een version control system.

Daarnaast kan een Heroku app gemaakt, gedeployed, getest en beheerd worden via de Heroku Command Line Interface (CLI) tool. In figuur 8.3 is een voorbeeld te zien van hoe met behulp van de Heroku CLI, een app gedeployed kan worden.

```

thijs@thijs:~$ cd git
thijs@thijs:~/git$ cd heroku
thijs@thijs:~/git/heroku$ ls
clienc-api  hibernate-test1  mttm-dev  mttm-dev2  mttm-wb-sync  mysterious-brook-5136
thijs@thijs:~/git/heroku$ cd mttm-dev
thijs@thijs:~/git/heroku/mttm-dev$ ls
application.conf  bin  index.html?action=joehoe  META-INF  pom.xml  pom.xml  Procfile  README.md  src  system.properties  target
thijs@thijs:~/git/heroku/mttm-dev$ git status
# On branch master
# Changes not staged for commit:
#   (use "git add <file>..." to update what will be committed)
#   (use "git checkout -- <file>..." to discard changes in working directory)
#
#       modified:   src/main/java/ResponseGetHandler.java
#       modified:   target/classes/ResponseGetHandler.class
#
# Untracked files:
#   (use "git add <file>..." to include in what will be committed)
#
#       src/main/java/getAppDeleteContact.java
#       target/classes/getAppDeleteContact.class
no changes added to commit (use "git add" and/or "git commit -a")
thijs@thijs:~/git/heroku/mttm-dev$ git push heroku master

```

Figuur 8.3 Voorbeeld van het uitvoeren van een deploy-actie met behulp van de Heroku CLI.

Heroku-apps draaien op "dyno's", waarbij één dyno feitelijk een virtuele server-instantie representeert. Voor iedere HTTP-request die gedaan wordt richting de Heroku-app zal er een dyno starten die het verzoek afhandelt.

Feitelijk betekent dit, dat hoe meer dyno's de applicatie tot zijn beschikking heeft, hoe meer verzoeken er parallel kunnen worden afgehandeld. Een verhoogd aantal dyno's komt de reactietijd van de applicatie dus ten goede.

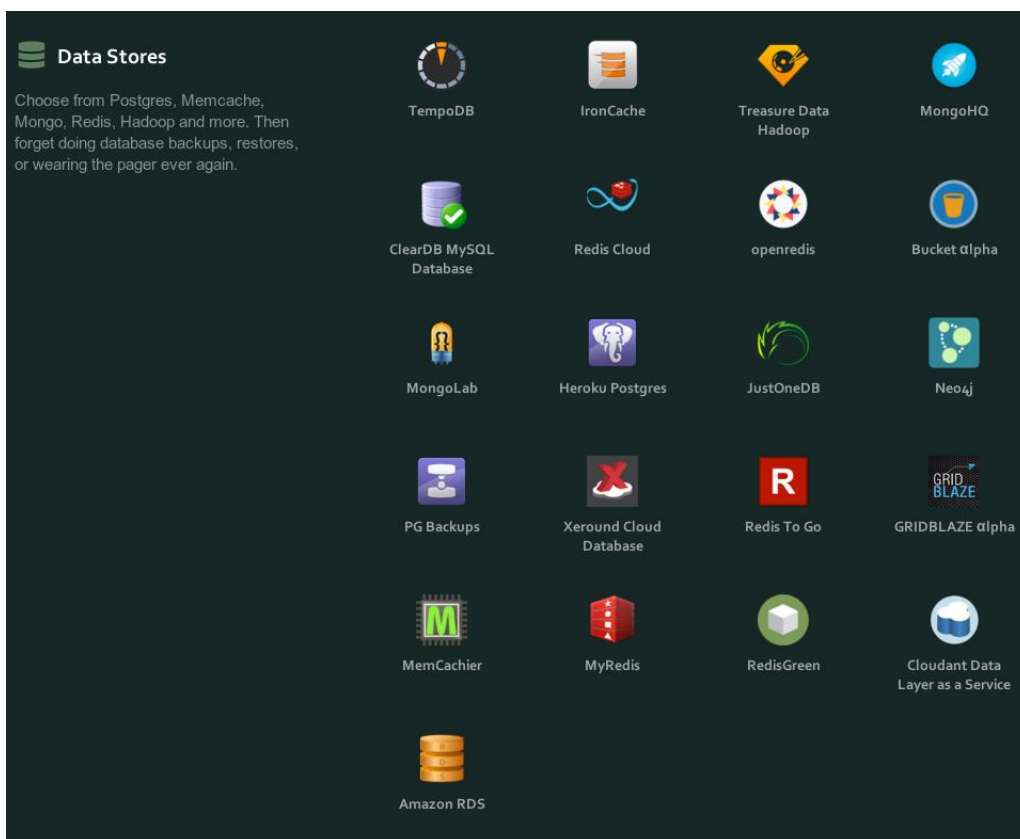
Standaard krijgt een Heroku-applicatie kosteloos één dyno toegewezen. Met behulp van de CLI kunnen Heroku apps geschaald worden naar meer of minder dyno's. Het draaien van een dyno kost \$0,05 per dyno per uur.

Na het intikken van het commando om dyno's op te schalen, neemt Heroku de technische verwerking voor zijn rekening en wordt de applicatie automatisch opgeschaald naar het gewenste aantal dyno's.

Op deze wijze is een applicatie eenvoudig te schalen, zonder verregaand met server-beheer en server-onderhoud bezig te zijn.

8.3.2. Heroku PostgreSQL database

Voor het opslaan van de Greeting-gegevens die op het Heroku-platform worden verwerkt, is er een database vereist. Het Heroku-platform biedt een verzameling add-ons, waaronder databases, die direct met Heroku geïntegreerd kunnen worden (figuur 8.4).



Figuur 8.4 Heroku add-ons voor data-opslag (bron: Heroku Add-ons website¹⁴)

Om de continuïteit van het platform te waarborgen, is gekozen voor een Heroku Postgres database (PostgreSQL) om de Greeting-gegevens in op te slaan. PostgreSQL is één van de meest populaire SQL-database-soorten.

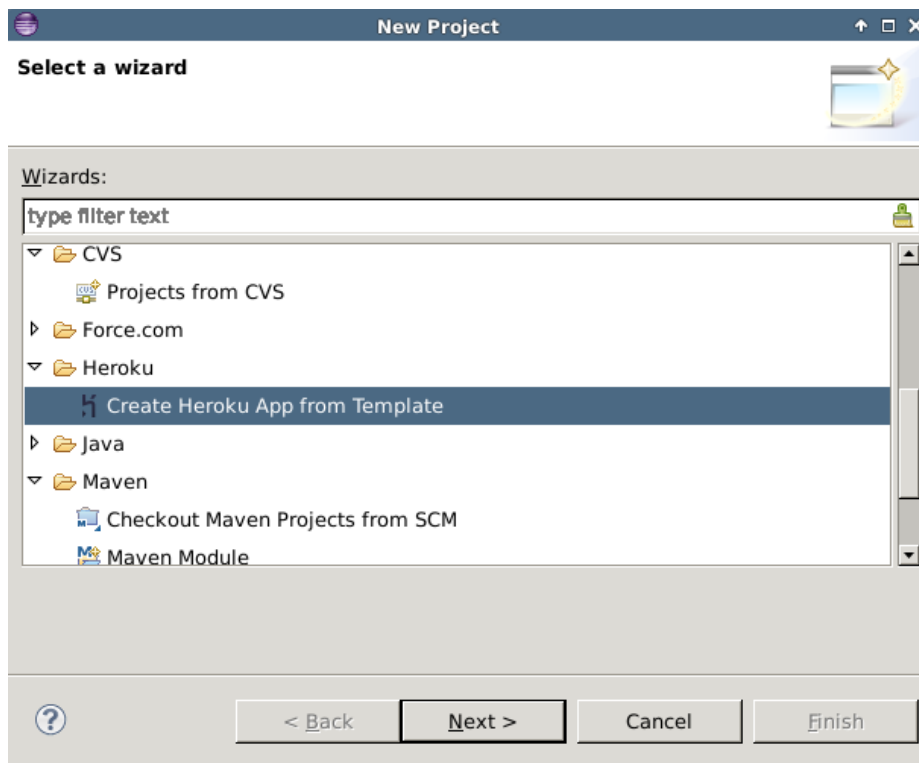
De website van PostgreSQL (bron: <http://www.postgresql.org/>) geeft een aantal kenmerken van Postgres databases, waaronder hoge schaalbaarheid, de mogelijkheid om met transacties te werken en een hoog aantal simultane gebruikers en bewerkingen. Postgres is een database-soort die door veel grote ondernemingen wordt gebruikt om data mee op te slaan.

Mede met deze mogelijkheden en door eigen ervaring van de opdrachtgever wordt de tussendatabase ontwikkeld in PostgreSQL.

8.3.3. Eclipse

Ten behoeve van het ontwikkelen voor het Heroku platform is gebruik gemaakt van de Eclipse IDE, waarmee met behulp van een plug-in* een koppeling kan worden toegevoegd met het Heroku-platform.

Met behulp van de Eclipse Heroku plug-in kunnen er direct Heroku-apps aangemaakt en gedeployed worden.



Figuur 8.5 Aanmaken van een Heroku App vanuit Eclipse met behulp van Heroku plug-in

In figuur 8.5 is een schermweergave te zien van het aanmaken van een heroku-app vanuit Eclipse. In figuur 8.6 is te zien hoe een Heroku-app gedeployed kan worden.

8.4. Ontwikkeling van mobiele applicatie voor Android

Voor het ontwikkelen van de mobiele applicatie is gebruik gemaakt van HTML5, een scripttaal waarmee webpagina's opgebouwd kunnen worden. Met behulp van CSS3 wordt vervolgens de vormgeving van deze webpagina's, die de schermen van de mobiele applicatie voorstellen, opgemaakt.

Tenslotte wordt Javascript gebruikt om de business-logica en de interactie van de gebruiker binnen de mobiele applicatie af te handelen.

*Plug-in: een klein software-onderdeel met eigen functionaliteit ter uitbreiding van bestaande software

Om ervoor te zorgen dat deze samenstelling van website-bestanden, stijlbestanden en Javascript geïnstalleerd kan worden als een opzichzelf staande app, en om gebruik te kunnen maken van de functionaliteiten die een hybride applicatie-framework vereisen, dient de app te worden voorzien van een hybride framework, als bijvoorbeeld Adobe Phonegap (of de open-source variant Apache Cordova).

Phonegap/Cordova is een hybride framework dat ervoor zorgt dat met dezelfde HTML5/CSS3/JS de native functionaliteiten vanuit Javascript aangesproken kunnen worden zonder daarvoor native programmeercode te schrijven voor het specifieke platform. Phonegap handelt deze vertaalslag af, en zorgt ervoor dat de Javascript-code die geschreven is, voor ieder platform de overeenkomstige functionaliteit uitvoert.

Een voorbeeld hiervan is te zien in figuur 8.7, waarin de native functionaliteit van het ophalen van de informatie van de contactpersoon kan worden aangesproken met Javascript.

```
<script type="text/javascript" charset="utf-8">

// Wait for Cordova to load
//
document.addEventListener("deviceready", onDeviceReady, false);

// Cordova is ready
//
function onDeviceReady() {
    // find all contacts with 'Bob' in any name field
    var options = new ContactFindOptions();
    options.filter="Bob";
    var fields = ["displayName", "name"];
    navigator.contacts.find(fields, onSuccess, onError, options);
}

// onSuccess: Get a snapshot of the current contacts
//
function onSuccess(contacts) {
    for (var i=0; i<contacts.length; i++) {
        console.log("Display Name = " + contacts[i].displayName);
    }
}

// onError: Failed to get the contacts
//
function onError(contactError) {
    alert('onError!');
}

</script>
```

Figuur 8.7 Aanspreken van native contacts met behulp van Javascript
(bron: Phonegap Documentation-website)

Android apps worden ontwikkeld in Java met behulp van de Android Software Development Kit (SDK). Hierin worden een aantal hulpmiddelen meegeleverd waarmee Android apps ontwikkeld kunnen worden, onder andere een SDK Manager, waarmee verschillende Android-versies binnengehaald kunnen worden, en een AVD-manager, waarmee virtuele Android apparaten geëmuleerd (nagebootst) kunnen worden.

Uiteindelijk kan de Android SDK in combinatie met Eclipse gebruikt worden om een Android app te 'build-en', hetgeen het samenstellen van een release aanduidt.

Deze release-versie kan vervolgens worden distribueerd via een app-store van het desbetreffende smartphone-platform, voor Android is dit de Google Play Store.

8.5. Ontwikkeling voor CRM Salesforce.com

Het CRM Salesforce.com staat bekend om haar uitbreidbaarheid en haar mogelijkheden met betrekking tot Custom Objects. Om het CRM Salesforce.com uit te breiden met business-logica, is er door Salesforce.com een programmeertaal en programmeeromgeving beschikbaar gesteld waarmee programmeurs delen voor het Salesforce.com-platform kunnen toevoegen. De programmeertaal APEX (een variant van Java) wordt hiervoor gebruikt. Er kan worden geprogrammeerd in de door Salesforce.com beschikbaar gestelde Force.com IDE. Met de Force.com IDE kunnen achterliggende triggers worden geschreven, die naar aanleiding van handelingen die gebruikers uitvoeren in Salesforce.com worden uitgevoerd.

Naast het schrijven van triggers, kunnen ook controllers worden geschreven voor pagina's binnen Salesforce.com. De vormgeving en gebruikersinterface van deze pagina's wordt dan bepaald door VisualForce, een opmaaktaal die qua syntaxis veel weg heeft van de opmaaktaal HTML.

In paragraaf 9.6 (Salesforce.com) worden de in te richten pagina's en controllers nader toegelicht.

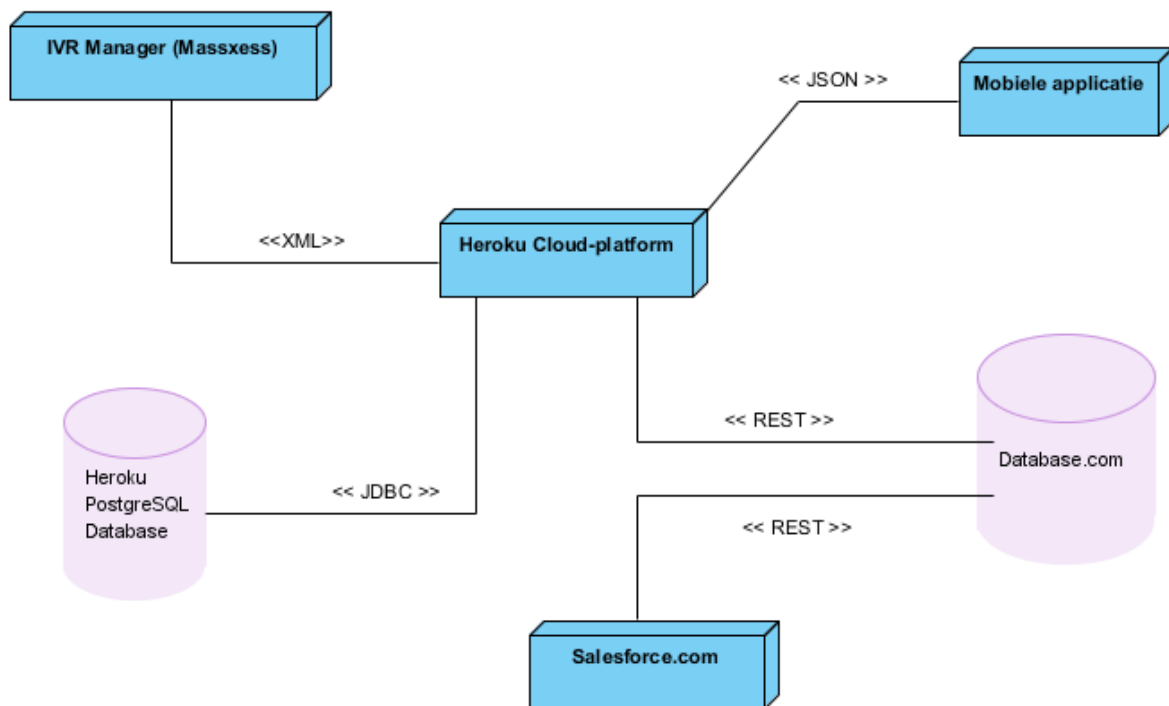
9. Architectuurbeschrijving

In dit hoofdstuk wordt de software-architectuur nader toegelicht waarmee de mobiele applicatie en het achterliggende platform zijn opgebouwd. De applicatie is gerealiseerd in de vorm van een 'proof-of-concept', om de gevonden onderzoeksresultaten te valideren.

9.1. Overzicht van de te realiseren infrastructuur

Om ervoor te zorgen dat de mobiele applicatie en achterliggende applicaties werken volgens gewenste eisen en wensen van de opdrachtgever, is gekomen tot hierna beschreven infrastructuur.

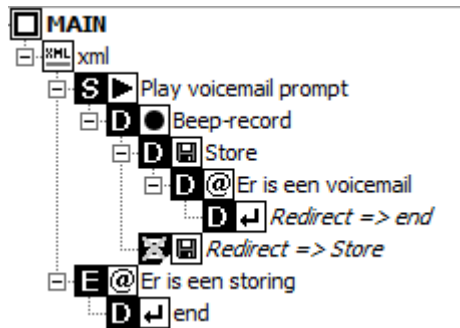
In figuur 9.1 is schematisch weergegeven hoe de infrastructuur dient te worden gerealiseerd. In hoofdstuk 10 wordt nader ingegaan op de implementatie (uitwerking) van het proof-of-concept: de mobiele applicatie Greeting.



Figuur 9.1 Overzicht gerealiseerde infrastructuur van Greeting-platform voor MTTM.nl (UML Deployment Diagram)

9.2. IVR Manager / Massxess

De afhandeling binnen het Greetingq-platform begint zodra het gesprek is doorgeschakeld naar het Greetingq-doorschakelnummer. Na binnenkomst op dit nummer, start het IVRscript van de desbetreffende Greetingq-gebruiker. Het IVR-script is opgebouwd volgens een vast stappenplan, zoals te zien in figuur 9.2.



Figuur 9.2 IVR-script zoals ingesteld in IVR Manager

Het IVR-script begint bij de start van het gesprek met het versturen van een XML-bericht (aangeduid door het -symbool. Hierin worden de gegevens van de beller meegegeven (zie figuur 9.3).

Name	Variable	Value	Type
A-nummer	\$CALLING_NUMBER		
B-nummer	\$CALLED_NUMBER		
Country	\$COUNTRYCODE		

Figuur 9.3 XML-parameters verzonden aan het begin van het gesprek

Deze XML-parameters (A-nummer, B-nummer, Country) worden verstuurd naar de Greetingq-applicatie die draait op Heroku. De applicatie handelt het inkomende verzoek af en stuurt een XML-bericht terug, waarin minstens de volgende parameters worden verwacht, zoals afgebeeld in figuur 9.4.

Name	Variable	Type
E-mail adres	E-mail adres	
GreetingQID	GreetingQID	
Klant	klant	
Voicemail prompt	filedata	MP3 to VOC

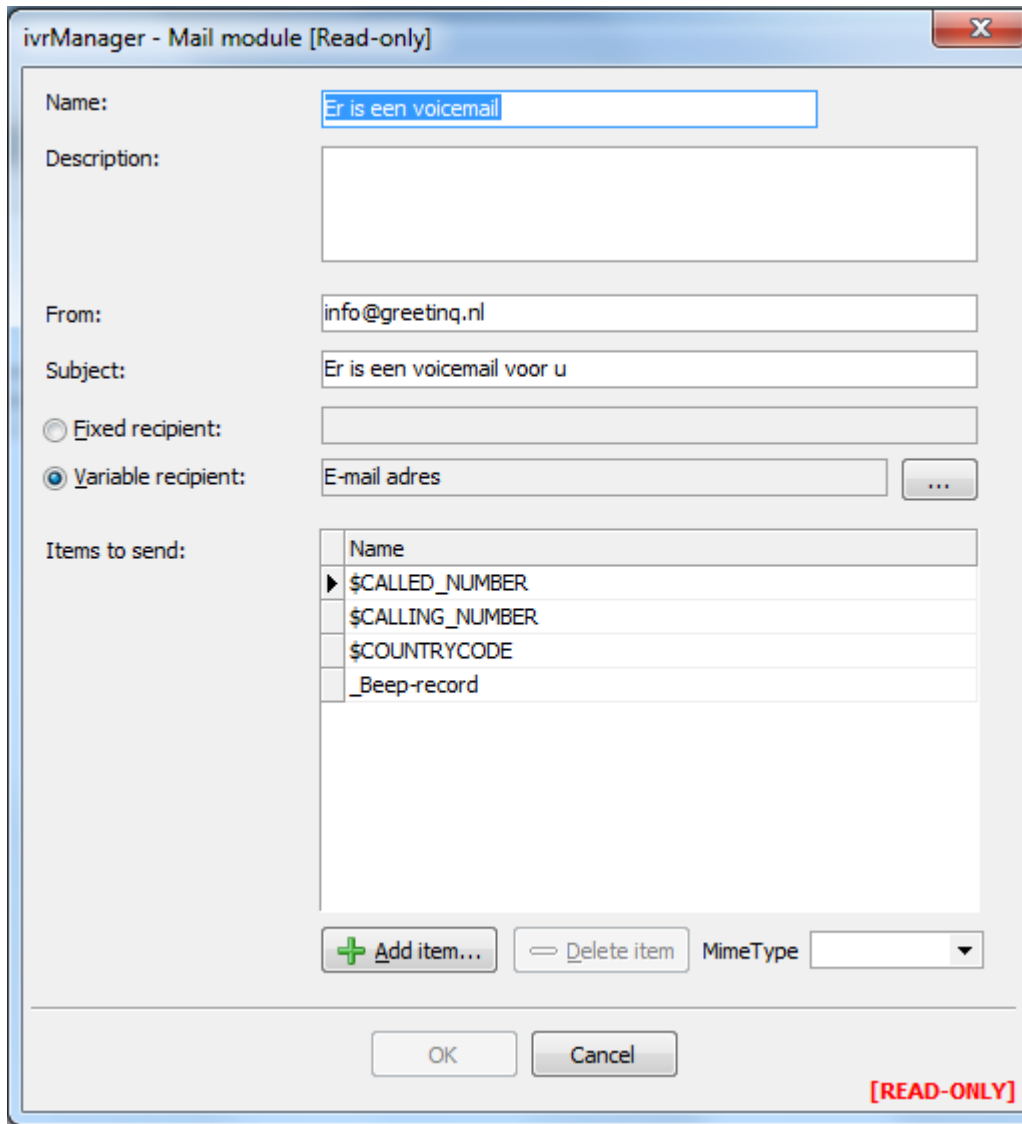
Figuur 9.4 XML-parameters verzonden vanuit Greetingq-applicatie op Heroku aan begin van gesprek

De Greetingq-applicatie vergelijkt de nummers van de beller en de gebelde (Greetingq-doorschakelnummer) met gegevens in de database en geeft op basis hiervan terug: het e-mailadres van de Greetingq-gebruiker, het GreetingqID van de Greetingq-gebruiker, of deze klant is (ja=1, 0=nee), en het af te spelen voicemailprompt.

De voicemailprompt wordt als Mp3 in Base64-formaat als XML-parameter meegezonden in het XML-bericht. IVR Manager zorgt er vervolgens voor dat de mp3 wordt omgezet in een voor de telefoniecentrale werkend geluidsformaat (VOC). Deze begroeting wordt aan de beller afgespeeld, waarna een pieptoon volgt.

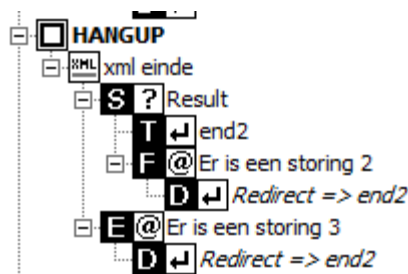
Indien de reactietijd van de Greetingq-applicatie te lang is, of er een fout optreedt, roept IVR Manager een alternatief scenario aan: de Error-module. Hierbij wordt er een e-mail gestuurd naar de Greetingq-applicatiebeheerder met daarin alle variabelen die op dat moment gebruikt worden en indien mogelijk een omschrijving van de opgetreden fout. Na het e-mailen van de foutmelding wordt het gesprek afgebroken.

Indien er geen fout optreedt en de pieptoon ten gehore wordt gebracht, spreekt de beller een boodschap in. Deze boodschap wordt opgenomen in het systeem van IVR Manager. Zodra de beller klaar is met inspreken, verbreekt hij de verbinding. IVR Manager stuurt direct daarna een e-mail met het ingesproken voicemailbericht als bijlage (in mp3-formaat) naar de Greetingq-klant, en gebruikt hiervoor het e-mailadres dat hij vanuit de Greetingq-applicatie heeft teruggekregen (zie figuur 9.4, eerste variabele).



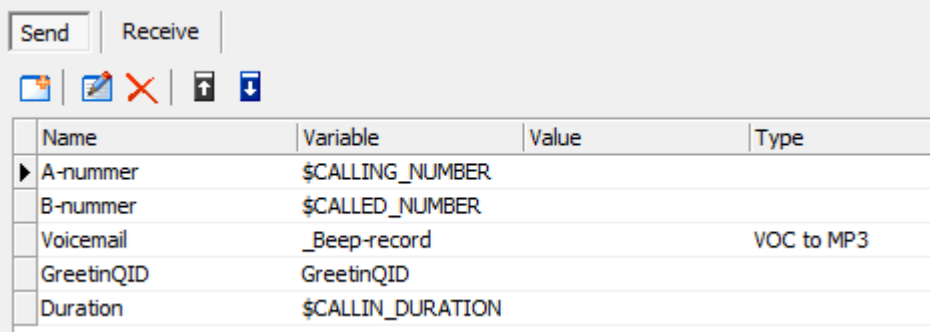
Figuur 9.5 E-mailmodule voor het versturen van het voicemailbericht

Nadat deze e-mail is verzonden, activeert IVR Manager het gedeelte van het IVR-script wat verantwoordelijk is voor de afhandeling ná afloop van het gesprek (zie figuur 9.6).



Figuur 9.6 IVR-script ná afloop van het gesprek

Zodra het gesprek is afgelopen, stuurt IVR Manager het tweede XML-bericht, bestaande uit de variabelen A-nummer, B-nummer, Voicemail, GreetingID, Duration (zoals te zien in figuur 9.7).

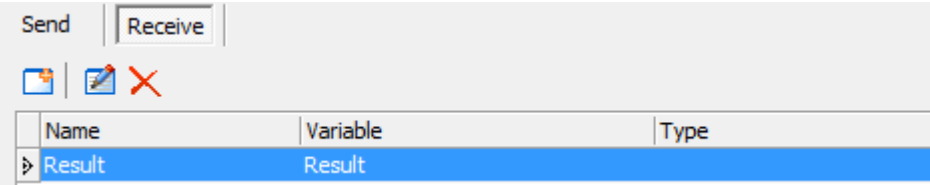


Name	Variable	Value	Type
A-nummer	\$CALLING_NUMBER		
B-nummer	\$CALLED_NUMBER		
Voicemail	_Beep-record		VOC to MP3
GreetingQID	GreetingQID		
Duration	\$CALLIN_DURATION		

Figuur 9.7 XML-bericht verzonden ná afloop van het gesprek

Het A-nummer is het nummer van de beller, en het B-nummer is het Greetingq-doorschakelnummer, beide gelijk aan het A-nummer en het B-nummer aan het begin van het gesprek. De variabele Voicemail is het opgenomen gesprek, dat naar de Greetingq-applicatie wordt verzonden, om te worden opgeslagen in de database. Daarnaast worden de duur van het gesprek en de Greeting ID meegegeven. De Greeting ID wordt meegegeven zodat de Greetingq-applicatie weet bij welke Greetingq-gebruiker deze voicemail dient te worden opgeslagen.

De Greetingq-applicatie verwerkt de binnengekomen voicemail en stuurt een XML-bericht terug met de variabele Result (zie figuur 9.8).



Name	Variable	Type
Result	Result	

Figuur 9.8 XML-bericht ontvangen ná afloop van het gesprek

Als de variabele Result niet wordt teruggegeven (althans niet binnen een termijn van 25 seconden), gaat IVR Manager er in de IVR-flow vanuit dat er iets is fout gegaan, en dan wordt een Error-module gestart die een mail stuurt naar de applicatiebeheerder, met alle systeemvariabelen en indien mogelijk een foutmelding. Deze actie volgt ook wanneer de variabele Result 0 (0 = niet succesvol) teruggeeft.

Als de variabele Result gelijk is aan 1 (1 = succesvol), eindigt het IVR-script. De voicemail is dan succesvol opgeslagen.

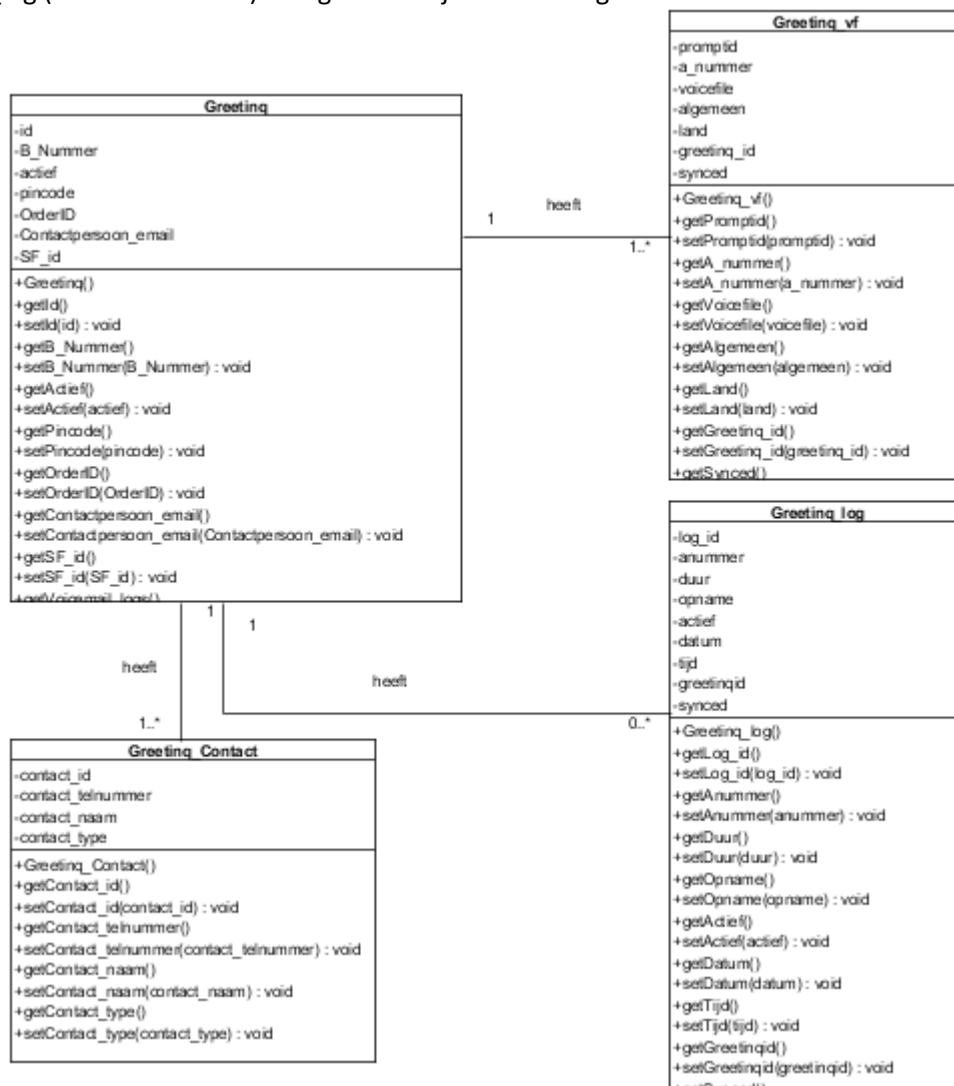
In de volgende paragraaf wordt beschreven hoe de XML-berichten aan de kant van de Greetingq-applicatie worden afgehandeld. Daarnaast wordt beschreven hoe de synchronisatie tussen de Greetingq-applicatie en Salesforce.com/Database.com is ingeregeld.

9.3. Heroku platform

Om de Greetingq-applicatie op het Heroku-platform de inkomende doorschakelingen te laten afhandelen is er in IVR Manager een XML module ingesteld (zie paragraaf 9.2). De XML module is zo ingesteld dat tijdens elk gesprek er twee keer XML-verkeer plaatsvindt: éénmaal bij de start van het gesprek, en éénmaal na afloop van het gesprek.

De XML-modules roepen beide een URL aan van de Greetingq-applicatie op het Heroku-platform. Aan het begin van het gesprek wordt de URL (<http://pure-bayou-9960.herokuapp.com/?action=checkNumber>) met behulp van een XML-bericht aangeroepen, en na afloop van het gesprek wordt de URL (<http://pure-bayou-9960.herokuapp.com/?action=saveVoicemails>) aangeroepen met behulp van een XML-bericht.

Het eerste gedeelte van het gesprek wordt opgevangen door de numberCheck controller van de Greetingq-applicatie op het Heroku-platform. Hiervoor zijn een aantal Java-classes geschreven die de entiteiten van de Greetingq-dienst vertegenwoordigen: Greetingq, Greetingq_vf (Voicefile/begroeting), Greetingq_log (voicemailbericht). In figuur 9.9 zijn de classes gemodelleerd.



Figuur 9.9 Klassediagram voor Greetingq-classes gebruikt in de applicaties van het Greetingq-platform

De class 'Greetingq' is een model van de Greetingq-gebruiker. De class 'Greetingq_Voicefile' en de class 'Greetingq_Log' vertegenwoordigen respectievelijk de begroetingen (voicefiles/voiceprompts) en de voicemailberichten. De laatste class 'Greetingq_contact' vertegenwoordigt één van de contactpersonen van de Greetingq-gebruiker.

9.4. Heroku Postgres Cloud Database

Om de gegevens (Greetingq-gebruikers, begroetingen en voicemails) op te kunnen slaan, is er door opdrachtnemer voor gekozen om een PostgreSQL-database te gebruiken (zoals toegelicht in paragraaf 8.3.2. Heroku Postgres).

In deze paragraaf wordt de structuur van de database-tabellen beschreven.

```

Table "public.greetingq"
  Column          |          Type          | Modifiers
-----+-----+-----
greetingq_id     | integer                | not null default nextval('greetingq_greetingq_id_seq'::regclass)
greetingq_b_nummer | character varying(15) |
greetingq_actief  | integer                |
greetingq_pincode | character varying(6)  |
greetingq_order_id | character varying(20) |
greetingq_sf_id   | character varying(20) |
greetingq_name    | character varying(100)|
greetingq_my_id   | character varying(15) |
greetingq_email_contactpersoon | character varying(500) |
greetingq_type    | character varying(50) |
greetingq_pass    | character varying(32) |
greetingq_telnum  | character varying(15) |
Indexes:
    "greetingq_greetingq_id_key" UNIQUE, btree (greetingq_id)

```

Figuur 9.10 Tabelstructuur van relatie 'greetingq' (Greetingq-gebruiker)

```

Table "public.greetingq_vf"
  Column          |          Type          | Modifiers
-----+-----+-----
vf_id            | integer                | not null default nextval('greetingq_vf_vf_id_seq'::regclass)
vf_a_nummer     | character varying(15) |
vf_algemeen     | integer                |
vf_land         | character varying(3)  |
vf_greetingq_id | integer                |
vf_synced       | integer                |
vf_voicefile    | bytea                  |
vf_old_filename | character varying(500) |
vf_naam         | character varying(500) |
vf omschrijving | character varying(500) |
vf_lengte       | integer                |
Indexes:
    "greetingq_vf_vf_id_key" UNIQUE, btree (vf_id)

```

Figuur 9.11 Tabelstructuur van relatie 'greetingq_vf' (Begroeting)

```

Table "public.greeting_log"
-----
Column | Type | Modifiers
-----
log_id | integer | not null default nextval('greeting_log_log_id_seq'::regclass)
log_greeting_id | integer |
log_datum | date |
log_tijd | time without time zone |
log_a_nummer | character varying(15) |
log_duur | integer |
log_actief | integer |
log_synced | integer |
log_opname | bytea |
log_old_filename | character varying(500) |
log_isnieuw | integer |
Indexes:
"greeting_log_log_id_key" UNIQUE, btree (log_id)

```

Figuur 9.12 Tabelstructuur van relatie 'greeting_log' (Ingesproken Voicemails)

```

Table "public.greeting_contact"
-----
Column | Type | Modifiers
-----
contact_id | integer | not null default nextval('greeting_contact_contact_id_seq'::regclass)
contact_nummer | character varying(15) |
contact_naam | character varying(200) |
contact_greeting_id | integer |
contact_type | character varying(100) |
Indexes:
"greeting_contact_contact_id_key" UNIQUE, btree (contact_id)

```

Figuur 9.13 Tabelstructuur van relatie 'greeting_contact' (Contactpersoon)

9.5. Database.com

Vanwege de vele limieten die Salesforce.com hanteert voor het uitlezen en opslaan van data is gekozen voor een tussenliggende database op het Heroku cloud-platform. Er is echter nog steeds een vereiste om alle informatie in Salesforce.com zichtbaar te maken.

In overleg met de opdrachtgever en mede vanwege de ervaring door de IT-afdeling met Database.com is ervoor gekozen om alle informatie over de Greeting-gebruikers, Greeting-begroetingen, Greeting-voicemails en Greeting-contactpersonen op te slaan in Database.com.

Database.com werkt met dezelfde achterliggende database en heeft dezelfde functionaliteiten als Salesforce.com. Database.com is door Salesforce.com als cloud-database ontwikkeld waarbij de focus ligt op custom object data-opslag. Database.com wijkt af van Salesforce.com in de grafische interface.

Er is namelijk geen grafische interface (geen VisualForce-pagina voor een eindgebruiker) waarin de informatie die is opgeslagen kan worden bekeken. Met behulp van queries (bijvoorbeeld vanuit een Salesforce.com-omgeving) of de ingebouwde webservices is het mogelijk om de informatie uit de database te halen.

9.6. Salesforce.com

Met behulp van VisualForce is er een Greeting-beheerpagina gemaakt door opdrachtnemer in het bestaande Salesforce.com-CRM. In figuur 8.8 is een voorbeeld te zien van de Greeting-beheerpagina, waarmee een overzicht van de Greeting-accounts te zien is binnen Salesforce.com. Deze informatie wordt direct opgehaald vanuit Database.com, een cloud-database die ook wordt geëxploiteerd door Salesforce.com.

The screenshot shows the Salesforce.com interface for Greeting management. The page title is 'Greeting-gebruikers'. The table below lists the data:

Naam	Greeting-nummer	E-mail contactpersoon	OrderID	Actief
0000000002	06test	thijs.otter@gmail.com	test1	true
0000000003	0612345678	thijsotter@gmail.com	12345	true
0000000001	31681262642	th.ijsotter@gmail.com	test3	true
0000000007	1234	thij.sotter@gmail.com	12345749	true
0000000011	0612345688	test@onlineq.nl	45454	true
0000000005	3123711	thijsotter@gmail.com	test5	true
0000000006	0612345679	t.hijsotter@gmail.com	12345667889	true
0000000008	12345678	thijsotte.r@gmail.com	0987654321	true
0000000009	023456789	thijsotter@gmail.com	789456123594	true
0000000013	0887000502	thijsotter@gmail.com	test	true

Figuur 9.10 Visualforce-pagina Greeting Beheer

10. Proof of Concept / Implementatie

Dit hoofdstuk beschrijft de uiteindelijke implementatie van de mobiele voicemail-applicatie Greeting. In paragraaf 10.1 wordt de algemene opbouw van de mobiele applicatie uiteengezet. Vervolgens wordt in de paragrafen daarna telkens een hoofdfunctionaliteit van de mobiele applicatie onder de loep genomen.

10.1. Opbouw Mobiele Applicatie Greeting

De applicatie is, zoals eerder toegelicht, opgebouwd als een webapp in een hybrid framework zodat, met behulp van de webscriptingtaal Javascript, de native functionaliteiten van de telefoon aangesproken kunnen worden.

De app bestaat uit de volgende hoofdschermen (met daarachter de paragraaf waarin de schermen worden toegelicht):

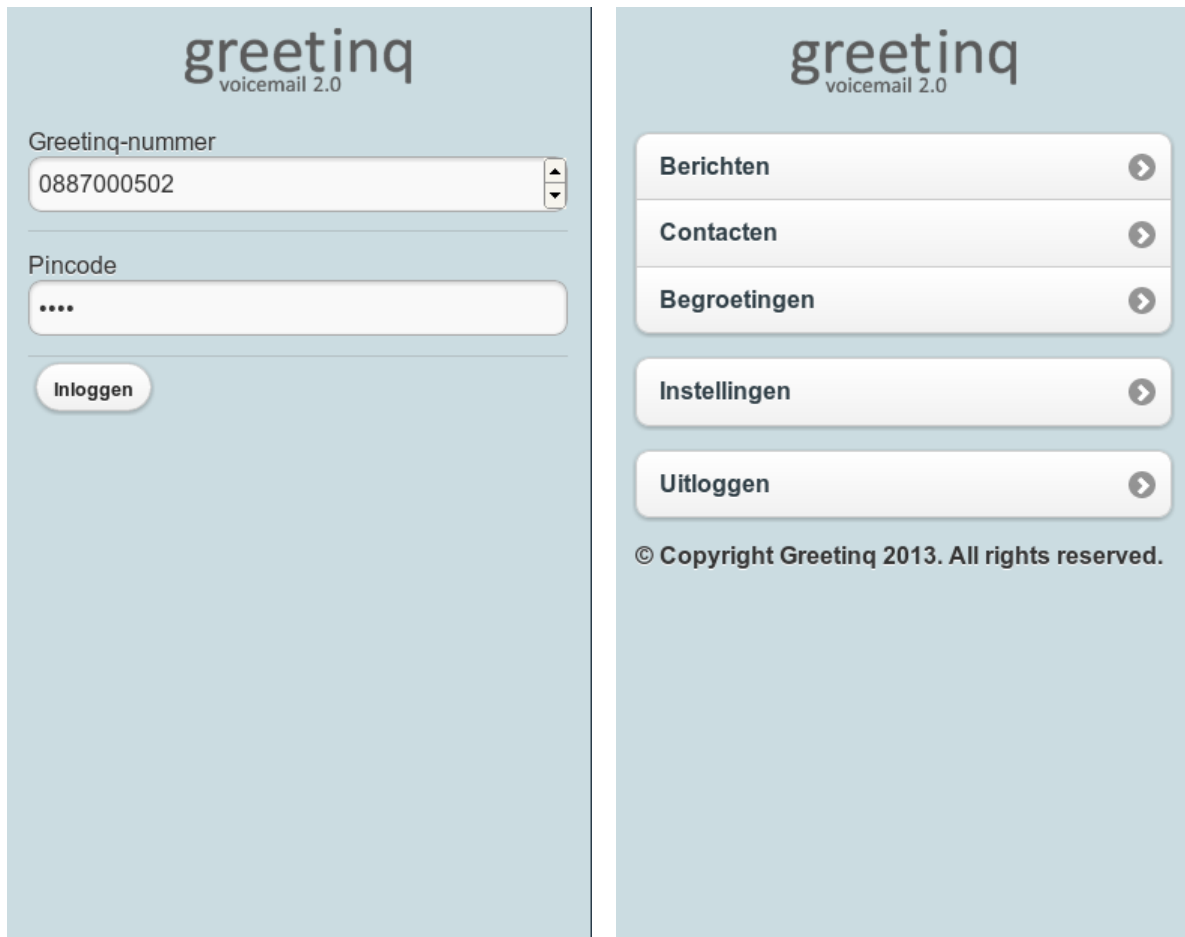
- Inloggen / Menu (paragraaf 10.2)
- Voicemailberichten (paragraaf 10.3)
- Contactpersonen (paragraaf 10.4)
- Begroetingen (paragraaf 10.5)

10.2 Home-scherm / Menu

Na het opstarten van de mobiele telefoon verschijnt een splashscreen (laadscherm) met het logo van MTTM.nl. Dit splashscreen staat ongeveer 3 seconden in beeld. Tijdens het tonen van het splashscreen worden op de achtergrond de, voor de app benodigde, resources ingeladen.

Na het inladen van de resources verdwijnt het splash-screen en wordt de webview opgestart en het hybrid framework geladen. Zodra beide zijn geladen verschijnt het inlogscherm van Greeting, indien de gebruiker nog niet eerder heeft ingelogd of indien de gebruiker zich bij het laatste gebruik heeft uitgelogd.

De gebruiker logt in met zijn toegewezen Greeting-nummer en een unieke pincode. Zodoende kan het systeem herkennen om welke gebruiker en dus welke voicemailbox het gaat.



Figuur 10.1 Home-schermen van de mobiele Greetingq-app (Android) – niet ingelogd (links), ingelogd (rechts)

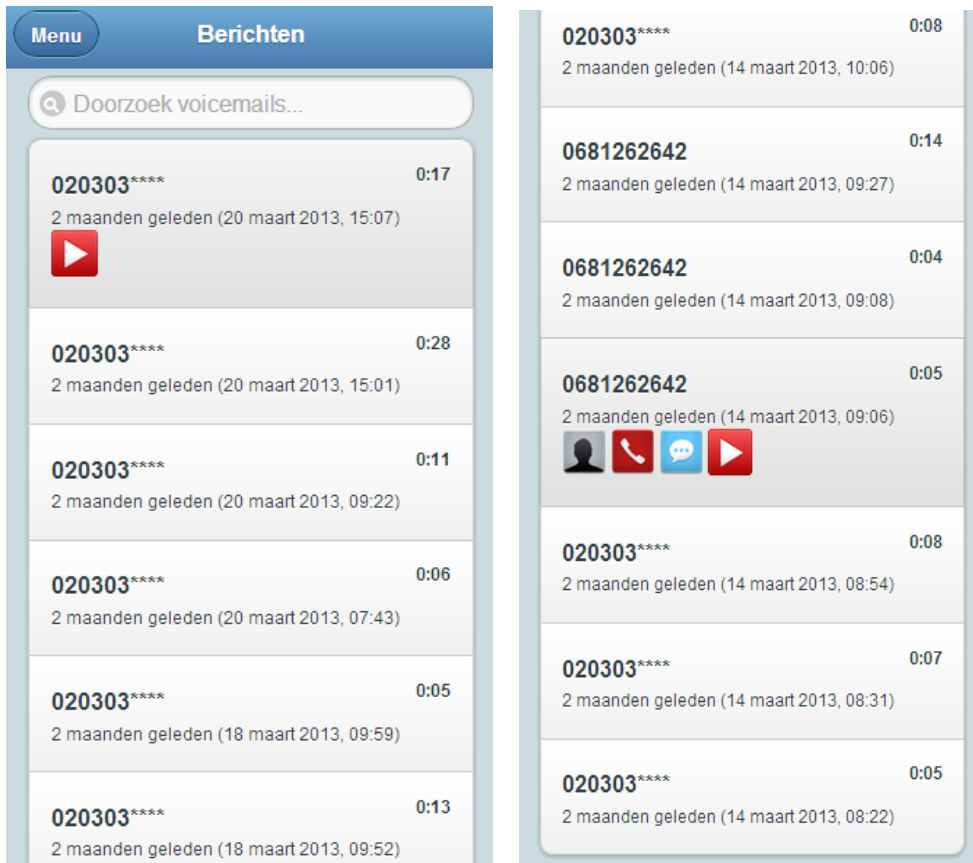
In figuur 10.1 is te zien hoe het eerste scherm aan de gebruiker getoond wordt na het starten van de app. Links is te zien hoe de applicatie het home-scherm toont wanneer de gebruiker nog niet eerder is ingelogd. Na het invoeren van de juiste logingegevens wordt het hoofdmenu van de Greetingq-app getoond.

10.3 Voicemailberichten

In onderstaande paragrafen worden de functionaliteiten van de mobiele app met betrekking tot de binnenkomende voicemailberichten toegelicht.

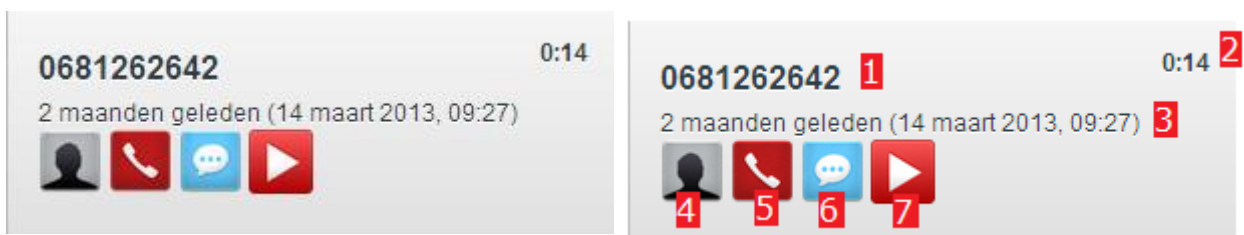
10.3.1. Berichten bekijken en beluisteren

In het scherm 'Berichten', zoals te zien in figuur 10.2, kan de gebruiker een overzicht zien van zijn ontvangen voicemailberichten.



Figuur 10.2 Overzicht van ontvangen voicemailberichten

In figuur 10.3 is te zien hoe een voicemailbericht is opgebouwd.



Figuur 10.3 Opbouw van een voicemailbericht
(1: Nummer/naam van de beller, 2: gespreksduur, 3: datum/tijd van voicemailbericht, 4: open contactpersoon in telefoon, 5: bel contactpersoon terug, 6: stuur sms naar contactpersoon, 7: beluister voicemailbericht)

10.3.2. Berichten downloaden

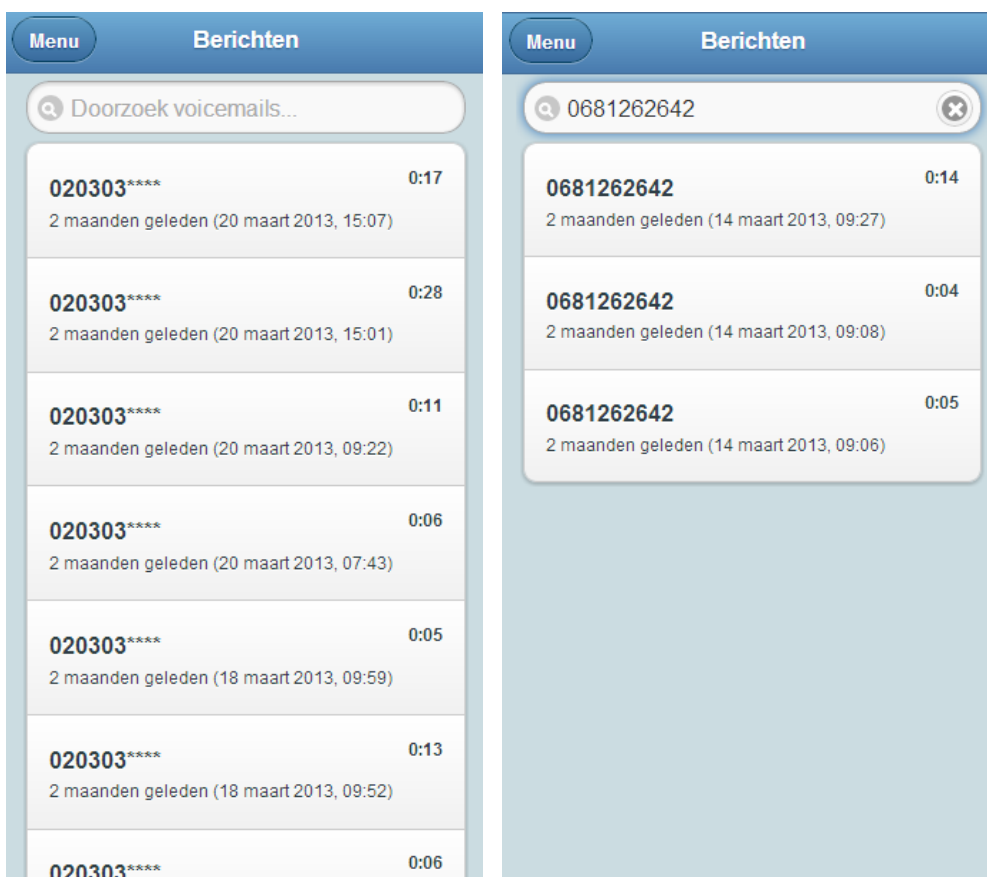
Indien de mobiele applicatie het afluisteren van de binnengekomen voicemailberichten niet ondersteunt, biedt de app de mogelijkheid de berichten te downloaden en op te slaan op de telefoon, zodat deze door de gebruiker op een ander apparaat beluisterd kunnen worden.

10.3.3. Berichten verwijderen

Naast de mogelijkheid om de berichten te beluisteren en te downloaden, is er uiteraard de mogelijkheid om de voicemailberichten te verwijderen. Na het drukken op het 'verwijder'-knopje in de applicatie worden de berichten definitief verwijderd.

10.3.4. Berichten filteren

Met behulp van de filterbalk, zoals in figuur 10.4 te zien is, kunnen berichten gefilterd worden op (een deel van de) datum of speelduur.



Figuur 10.4 Filteren van de berichten-lijst (links) op bijvoorbeeld telefoonnummer (rechts)

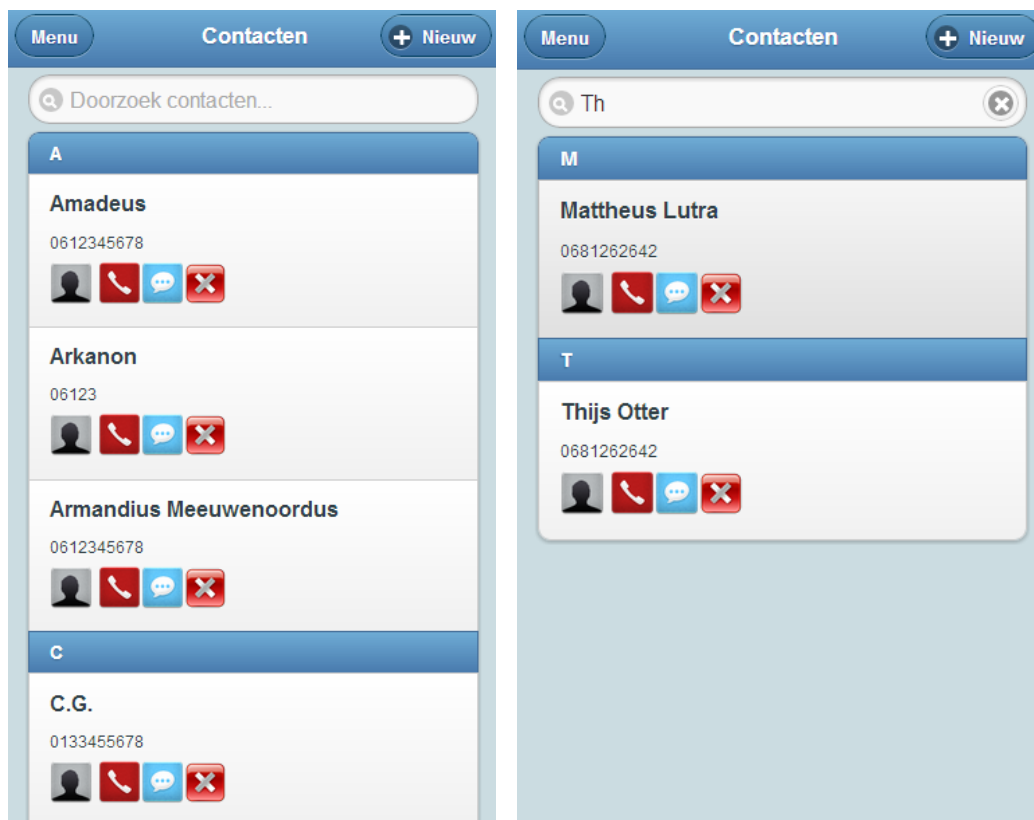
Indien op het kruisje wordt gedrukt, aan de rechterkant van de filterbalk, verdwijnt het filtercriterium en wordt de volledige lijst met voicemailberichten getoond. Zodoende is er voor de gebruiker snel te schakelen tussen een gefilterde lijst en de volledige lijst. Het filteren gebeurt bovendien direct na het intoetsen van één of meer letters en/of cijfers. Bij iedere wijziging in het filtercriterium filtert het overzicht automatisch. Zodoende hoeft er niet na iedere aanpassing op een knop gedrukt te worden.

10.4 Contactpersonen

In onderstaande paragrafen worden de verschillende functionaliteiten van de contactpersonen toegelicht.

10.4.1. Contactpersonen bekijken

Na het drukken op de knop 'Contacten' komt de gebruiker in het overzichtsscherm met alle contactpersonen die onder de Greetingq-account van de gebruiker beschikbaar zijn. Dit is een combinatie van contactpersonen uit de online portal van Greetingq, aangevuld met geïmporteerde contacten uit de smartphone, of toegevoegd aan de app, of geïmporteed vanuit Outlook of LinkedIn.



Figuur 10.5 Overzichtsscherm Contactpersonen ongefilterd (links) en gefilterd op criterium 'Th' (rechts).

10.4.2. Contactpersonen toevoegen

Het toevoegen van een contactpersoon gaat door het indrukken van de knop 'Nieuw' in het Contactpersonen overzicht. Dan opent het 'contactpersoon toevoegen'-scherm waarmee een Greetingq-contactpersoon kan worden toegevoegd. Na het invullen van naam en telefoonnummer (zie figuur 10.6), kan de contactpersoon worden toegevoegd.



Terug Contact toevoegen

Naam contactpersoon

Telefoonnummer

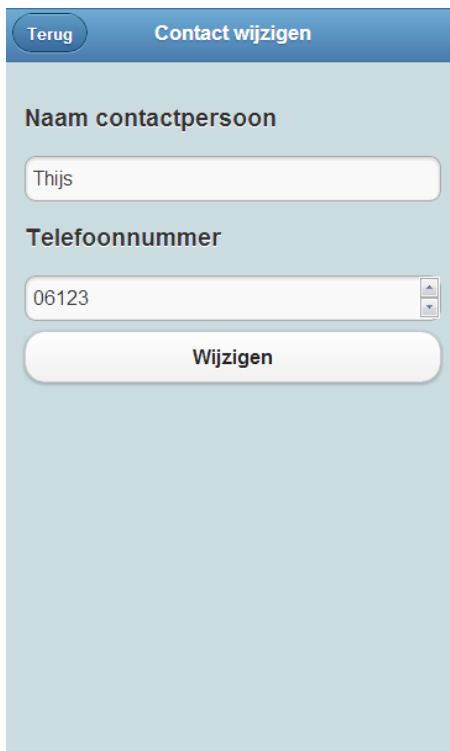
0612345678

Toevoegen

Figuur 10.6 Toevoegen van een contactpersoon

10.4.3. *Contactpersonen wijzigen*

Het wijzigen van een contactpersoon werkt op eenzelfde manier als het toevoegen van een contactpersoon. Na het wijzigen van naam en/of telefoonnummer kan de contactpersoon worden opgeslagen. Zie figuur 10.7 voor een schermvoorbeeld van het wijzigen van een contactpersoon.



Figuur 10.7 *Wijzigen van een contactpersoon*

10.4.4. Contactpersonen verwijderen

Verwijderen van een contactpersoon geschiedt door het drukken op het kruis-icoontje, zoals afgebeeld in figuur 10.8. Na het bevestigen van het definitief verwijderen van deze contactpersoon, wordt de contactpersoon uit de app en uit het account van de Greeting-gebruiker verwijderd.



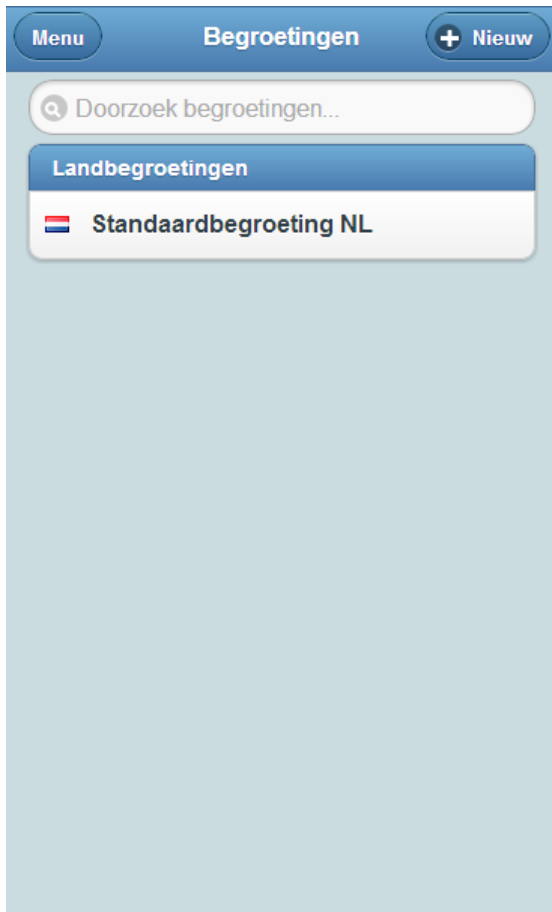
Figuur 10.8 *Verwijder-knopje voor het verwijderen van een contactpersoon*

10.5 Begroetingen

Naast de functionaliteiten Berichten en Contacten, is 'Begroetingen' een belangrijk onderdeel van de Greeting-dienst en mobiele app. In de volgende paragrafen worden de deelfunctionaliteiten van het onderdeel Begroetingen toegelicht.

10.5.1. Begroetingen bekijken en beluisteren

Een overzicht van de begroetingen raadplegen kan via het scherm 'Begroetingen'. In figuur 10.9 is een schermvoorbeeld te zien van het 'Begroetingen'-scherm.



Figuur 10.9 *Bekijken van begroetingen*

Na het drukken op een begroeting, kan deze afgehuisterd worden. Indien de begroeting niet kan worden afgespeeld, kan deze door middel van een download-knop worden gedownload.

10.5.2. *Begroetingen toevoegen*

Ook is het mogelijk om begroetingen toe te voegen in mp3-formaat via de app. Daartoe kan via de knop 'Nieuw' het 'Begroetingen toevoegen'-scherm worden geopend.

In figuur 10.10 is te zien hoe een begroeting kan worden toegevoegd via de app. De app-gebruiker kan er voor kiezen om een landbegroeting toe te voegen. Hierbij wordt de begroeting afgespeeld, zodra er een beller naar de voicemailbox belt die een nummer heeft in het door de Greeting-gebruiker ingestelde land.

Daarnaast kan de app-gebruiker een persoonlijke begroeting instellen. Een persoonlijke begroeting wordt ingesteld voor één bepaalde contactpersoon of één bepaald telefoonnummer.

Figuur 10.10 Toevoegen van een landbegroeting (links) of een persoonlijke begroeting (rechts)

10.5.3. *Begroetingen verwijderen*

Begroetingen verwijderen gaat, zoals ook bij Berichten en Contacten met behulp van het verwijderknopje (zie figuur 10.11) in het Berichtenoverzicht.



Figuur 10.11 Verwijderknopje voor het verwijderen van begroetingen

10.5.4. *Begroetingen downloaden*

Indien het apparaat of de smartphone het afspelen van geluidsbestanden niet ondersteunt, beschikt de app-gebruiker over de mogelijkheid om de begroeting te downloaden door middel van een download-knop (deze verschijnt in plaats van de afspeel-knop).

10.5.5. Begroetingen uploaden

Het uploaden van een begroeting gaat via het toevoegen van een begroeting zoals toegelicht in paragraaf 10.5.2 (Begroetingen toevoegen).

11. Validatie Proof of Concept naar eisen en wensen opdrachtgever

In dit hoofdstuk wordt bepaald in hoeverre het proof-of-concept (de gerealiseerde mobiele applicatie en het achterliggende Greeting-platform), zoals beschreven in hoofdstuk 10, voldoet aan de wensen en de eisen zoals gesteld door de opdrachtgever. Tevens wordt het proof-of-concept getoetst aan de maatregelen ten behoeve van de continuïteit van de applicatie(s), zoals door opdrachtnemer opgesteld in hoofdstuk 6.3 (Vastgestelde maatregelen ter waarborging van de continuïteit van de applicatie)

11.1. Validatie van must-haves

Vanuit de directie van MTTM.nl zijn een aantal eisen (must-haves) gedefiniëerd voor de (mobiele) applicatie. Van deze must-haves wordt in nevenstaande paragrafen beschreven in hoeverre de gerealiseerde applicatie voldoet aan deze must-haves.

11.1.1. Koppeling met telefoonboek

Vanuit de eis dat de mobiele applicatie gebruik dient te maken van een koppeling met het telefoonboek, kan worden gevalideerd of het proof-of-concept voldoet aan deze eis. In de ogen van de opdrachtnemer voldoet de applicatie aan de door opdrachtgever gestelde eis.

De mobiele app haalt bij ieder binnengekomen voicemailbericht de het bellende nummer op en controleert vervolgens of één er van de contactpersonen die door de Greeting-gebruiker onder zijn account zijn toegevoegd, een overeenkomstig nummer bekend is. Is er geen contactpersoon bekend met dit telefoonnummer, dan wordt vervolgens een koppeling gelegd met het interne telefoonboek van de smartphone, om op deze wijze een mogelijke contactpersoon bij dit nummer te koppelen.

11.1.2. Geluidsbestanden afspelen

De opdrachtgever heeft aangegeven dat de mobiele applicatie dient te beschikken over de functionaliteit om geluidsbestanden af te spelen, zodat de gebruiker van de mobiele applicatie de voicemailberichten kan afluisteren.

Door gebruik te maken van een WebView-element in de applicatie en door gebruik te maken van gestandaardiseerde technologieën als HTML5, beschikt het gerealiseerde proof-of-concept over een HTML5 audiospeler die op de gangbare smartphones werkt, en waarmee het mogelijk is om de geluidsbestanden in de applicatie af te spelen.

Ook aan deze eis is derhalve voldaan tijdens het ontwerp en de realisatie van het proof-of-concept.

11.1.3. Koppeling met CRM Salesforce.com van MTTM.nl

Door middel van een synchronisatie-applicatie op het Heroku-platform is de Greeting-dienst gekoppeld met de Salesforce.com/Database.com-omgeving vanuit MTTM.nl. Vanuit de beheeromgeving voor MTTM.nl, het CRM Salesforce.com, is de Greeting-dienst te beheren en te configureren. Hierin zijn de gegevens van de Greeting-gebruiker aan te passen. Daarom is aan de eis voldaan dat de Greeting-gebruiker vanuit Salesforce.com beheerd moet kunnen worden, en tevens

aan de eis voldaan de mobiele applicatie gekoppeld te zijn met het Salesforce.com CRM van MTTM.nl (weliswaar na synchronisatie via de Greetingq-applicatie op het Heroku-platform).

11.1.4. IVR Manager als infrastructuur

De laatste must-have aan de Greetingq-applicatie was de eis dat de Greetingq-dienst dient te functioneren met IVR Manager als onderliggende infrastructuur. Na het opleveren van proof-of-concept is uit de acceptatietest gebleken dat de Greetingq-dienst werkt met de IVR-Manager als onderliggende infrastructuur tussen de telefonie en de applicatie.

11.2. Validatie van should-haves

In deze paragraaf wordt nader ingegaan op validatie van de should-haves die door de opdrachtgever als dringende wensen aan de ontwikkeling van de applicatie zijn verbonden.

11.2.1. Doorschakeling tijdens installatie / eerste gebruik

Hoewel het bij aanvang van de ontwikkeling van de mobiele applicatie door opdrachtnemer mogelijk werd geacht om de mobiele applicatie zelfstandig een doorschakeling in te laten stellen, blijkt dit voor een Android-app niet mogelijk te zijn. Een Android-app heeft niet voldoende toegangsrechten om zelfstandig een telefoongesprek te starten met een vooraf ingevulde doorschakelcode. Hiertoe is door de opdrachtnemer een tweetal adviezen met betrekking tot alternatieven opgesteld:

1. Door het doorschakelnummer met de persoonlijke doorschakelcode in te stellen als contactpersoon in het telefoonboek van de telefoon, en door de gebruiker vervolgens te instrueren om deze te bellen, kan er toch een doorschakeling worden ingesteld.
2. Tijdens de eerste opstart van de mobiele applicatie kan een korte introductie worden getoond waarin aan de gebruiker wordt uitgelegd welke stappen dienen te worden ondernomen om de doorschakeling juist in te stellen. Dit is een eenmalig proces, dit hoeft enkel tijdens het eerste gebruik van de Greetingq-voicemaildienst of Greetingq-app te worden.

De opdrachtnemer heeft ervoor gekozen nog geen uitwerking van één van deze adviezen te verwerken in de proof-of-concept.

11.2.2. Importeren van contacten vanuit LinkedIn / Outlook

Het importeren van contactpersonen vanuit LinkedIn en Outlook in de mobiele applicatie is tevens een wens die door de opdrachtgever is uitgesproken.

Hoewel in eerste instantie door opdrachtnemer is ingepland om deze functionaliteit toe te voegen aan de mobiele applicatie, is de opdrachtnemer er door tijdgebrek (wegens vele ontwikkeluren besteed aan het ontwikkelen van de tussenliggende Heroku-applicatie) niet in geslaagd om deze functionaliteit in de mobiele applicatie toe te voegen.

In overleg met de opdrachtgever is besloten deze functionaliteit wel in de online portal toe te voegen zodat de Greetingq-gebruiker toch de mogelijkheid heeft om op een manier zijn of haar contacten te importeren.

11.3. Validatie van nice-to-haves

In deze paragraaf worden de nice-to-haves gevalideerd om te bepalen of aan de wensen van de opdrachtgever is voldaan. In paragraaf 11.3.1. wordt ingegaan op de wens vanuit opdrachtgever om in de mobiele applicatie begroetingen op te nemen. In paragraaf 11.3.2. wordt ingegaan op de wens vanuit opdrachtgever om in de mobiele applicatie gebruik te maken van notificaties om de gebruiker op de hoogte te stellen van nieuwe voicemailberichten.

11.3.1. Opnemen van begroetingen

Één van de wensen van de opdrachtgever aan de mobiele applicatie is de mogelijkheid om begroetingen niet alleen te uploaden, maar ook direct in te kunnen spreken.

Volgens de Phonegap/Cordova documentatie, is er een mogelijkheid om te werken met het opnemen en afspelen van audio. De 'Media API' van Cordova, geeft de mogelijkheid om audio op te nemen met behulp van Javascript.

Door opdrachtnemer is onderzocht of het opnemen van begroetingen mogelijk is, en hij heeft hier positieve resultaten in geboekt, opnemen van begroetingen is mogelijk. Wegens tijdgebrek en prioritering van overige functionaliteiten is opdrachtnemer er niet in geslaagd om het opnemen van begroetingen te implementeren binnen het door opdrachtnemer ontwikkelde proof-of-concept.

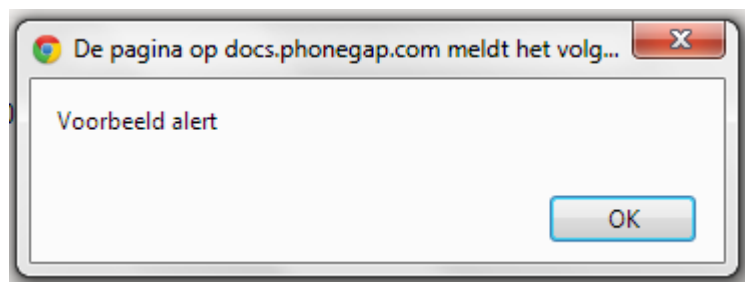
11.3.2. Notificaties bij nieuwe voicemailberichten

Naast het opnemen van begroetingen via de mobiele applicatie, is het ook een wens om de app-gebruiker een notificatie te laten ontvangen bij nieuwe voicemailberichten.

Hoewel de Phonegap/Cordova-documentatie aangeeft dat er gewerkt kan worden met notificaties, blijkt dat de, hier bedoelde, notificatie feitelijk een in-app notificatie is, vergelijkbaar met een alert in een webbrowser:

“ Most Cordova implementations use a native dialog box for this feature. However, some platforms simply use the browser's `alert` function, which is typically less customizable. ”

In figuur 11.1 is een voorbeeld te zien van een alert in een webbrowser, vergelijkbaar met een in-app notificatie.



Figuur 11.1 Voorbeeld van een alert in een webbrowser

Deze vorm van implementeren is echter niet toereikend om de gewenste functionaliteit te bereiken. Deze implementatievorm beperkt zich tot notificaties die enkel kunnen worden aangesproken

wanneer de app actief in gebruik is en op het scherm getoond wordt. Dat betekent dat wanneer de telefoon-gebruiker een andere app aan het gebruiken is, en er een nieuwe voicemail binnenkomt, deze notificatie dan niet wordt uitgevoerd, omdat de Greeting-app op dat moment niet actief is.

Hiervoor dient (in ieder geval voor Android) gebruik gemaakt te worden van een Service, een deel van een applicatie die op de achtergrond blijft draaien terwijl de app op de voorgrond reeds is afgesloten. Voor het gebruiken van een Service dient de app ontwikkeld te worden als een native app.

Het is dus niet mogelijk om aan de wens van de opdrachtgever te voldoen met betrekking tot de notificaties, omdat er gekozen is voor het ontwikkelen als hybride-app. Mocht de applicatie in een later stadium opnieuw ontwikkeld worden als een native app, dan kan er wel voor gekozen worden om de functionaliteit 'Notificaties' te implementeren.

11.4. Validatie van continuïteitsmaatregelen

In deze paragraaf worden de continuïteitsmaatregelen gevalideerd.

Uitvoerig gedocumenteerd

De Greeting-applicatie is gedocumenteerd, enerzijds door de programmeercode te voorzien van betekenisvol commentaar, anderzijds door de applicatie te beschrijven in een overdrachtsdocument die na afloop van het Greeting-project wordt overgedragen aan de opdrachtgever. In de ogen van de opdrachtnemer is dus voldaan aan de continuïteitsmaatregel om de applicatie uitvoerig te documenteren.

Progressive Enhancement

De continuïteit van de applicatie kan worden geborgd door het toepassen van *progressive enhancement*. Op de functionaliteit 'Afspelen van een voicemailbericht' is *progressive enhancement* toegepast. Indien het direct afspelen van een voicemailbericht vanuit de app niet wordt ondersteund, is het ook mogelijk om het voicemailbericht te downloaden, zodat deze met een externe muziekspeler kan worden afgespeeld en kan worden gedeeld.

Object-georiënteerd

De Greeting-applicatie op het Heroku Cloud-platform is geschreven in de programmeertaal Java, een object-georiënteerde programmeertaal. Door de opgedane ervaring en oefening tijdens de opleiding van opdrachtnemer, is het gebruik van Java en daarmee het object-georiënteerd programmeren in het Greeting-project consequent toegepast. In de mobiele applicatie, is deze techniek niet toegepast omdat de (programmeer en/of scripting)talen HTML5, CSS3 en Javascript zich hiervoor niet lenen. Door waar mogelijk, object georiënteerd programmeren toe te passen in de ontwikkeling van de applicatie, is aan de continuïteitsmaatregel voldaan.

Exception Handling

Ten behoeve van de continuïteitsmaatregel om foutafhandeling in te bouwen, is gebruik gemaakt van *Exceptions*, een standaardoplossing in Java om opgetreden fouten op een nette manier af te handelen zonder dat de rest van de applicatie hierop vastloopt. *Exceptions* worden afgevangen in het onderdeel 'catch' van een 'try/catch'-constructie.

Handelingen die data uitlezen of wegschrijven van/naar een externe bron (bijvoorbeeld een database-transactie) worden door middel van 'try/catch'-constructies in de programmeercode afgehandeld.

In de mobiele applicatie en IVR Manager kan vanwege de aard van de programmeertaal en de daarin ontbrekende voorzieningen geen gebruik gemaakt worden van *Exception Handling*. Desondanks is waar mogelijk gebruik gemaakt van foutafhandeling om aan de gestelde continuïteitsmaatregelen te voldoen.

Proactieve controles

Door op diverse delen in de applicaties een proactieve controle in te bouwen, die op bepaalde tijdsintervallen controleert of applicatielimiten zijn overschreden, kan continuïteit van de applicatie tevens worden geborgd. Hoewel aanvankelijk door de opdrachtnemer is besloten om in de diverse delen een proactieve controle in te bouwen, is hier door tijdgebrek in het ontwikkelproces, door opdrachtnemer niet in geslaagd. Er is, ten tijde van schrijven, dus aan deze continuïteitsmaatregel nog niet voldaan. Aan de opdrachtgever is echter wel aangeraden om deze continuïteitsmaatregel alsnog toe te voegen aan de applicatie(s).

Applicatiebeheerder

Om de continuïteit van de Greetingq-dienst te borgen, is het belangrijk een applicatiebeheerder aan te stellen. Advies aan de opdrachtgever hierin is iemand aan te stellen die verstand heeft van Salesforce.com, Database.com, telecom-oplossingen en maatwerk-software in de cloud.

11.5. Tussentijdse conclusie

Samenvattend kan de hierna volgende tussentijdse conclusie worden getrokken ten aanzien van het voldoen van de (mobiele) applicatie aan de must-haves, should-haves, nice-to-haves en de continuïteitsmaatregelen.

Met betrekking tot de must-haves (eisen van de opdrachtgever) kan worden geconcludeerd dat er aan de eisen is voldaan, omdat de mobiele applicatie beschikt over een koppeling met het interne telefoonboek en over de mogelijkheid om geluidsbestanden af te spelen. Daarnaast is de applicatie gekoppeld aan Salesforce en werkt deze op basis van IVR Manager. Derhalve is aan de eisen van de opdrachtgever voldaan.

Aan de should-haves is tijdens de ontwikkeling van het proof-of-concept niet voldaan. Zowel de mogelijkheid om zelfstandig een doorschakeling in te stellen tijdens het eerste gebruik als het importeren van de contacten vanuit LinkedIn en Outlook, zijn niet aanwezig in het ontwikkelde proof-of-concept.

Daarnaast zijn de nice-to-haves, het opnemen van de begroetingen, en het versturen van notificaties niet opgenomen in het proof-of-concept.

De mobiele applicatie voldoet wel aan de absolute eisen van de opdrachtgever en aan de gestelde continuïteitsmaatregelen, maar er is nog verbetering nodig om ook de should-haves en nice-to-haves op te nemen als functionaliteit. Advies vanuit opdrachtnemer aan opdrachtgever is dan ook, om de mobiele applicatie ontwikkeld door opdrachtnemer nog niet op de markt te brengen. De Greetingq-dienst kan wel (zonder mobiele app) worden aangeboden.

12. Conclusie en aanbevelingen

Om antwoord te geven op de vraag hoe een mobiele visual voicemail applicatie kan worden ontwikkeld naar de eisen en wensen van de opdrachtgever, is door opdrachtnemer onderzoek gedaan naar de mogelijkheden in de bestaande situatie, met IVR Manager als telefonie-centrale en Salesforce.com als CRM.

In het door opdrachtnemer gedane onderzoek, is naar voren gekomen dat IVR Manager in de huidige situatie (configuratie) niet geschikt is voor het afhandelen van de dynamisch ingestelde begroetingen van de Greeting-gebruiker. Hiervoor heeft opdrachtnemer onderzocht of de eerder besproken oplossing van XML-transacties tijdens het voicemailgesprek geïmplementeerd kunnen worden.

Naast het onderzoeken van de mogelijkheden van IVR Manager, is door opdrachtnemer onderzocht welke webservices er zijn om het CRM Salesforce.com te gebruiken als back-end voor de Greeting-dienst. Er zijn zowel SOAP als REST-services aanwezig, die het mogelijk maken om informatie uit Salesforce.com op te halen of informatie op te slaan.

Eenzijds lijkt het CRM Salesforce.com niet te beschikken over de benodigde opslagmogelijkheden met betrekking tot de geluidsbestanden van de begroetingen en voicemails van de Greeting-dienst, anderzijds lijken de door Salesforce.com gehanteerde limieten ook niet toereikend genoeg om de verzoeken van IVR Manager in grote getale te kunnen afhandelen, teneinde de Greeting-dienst te laten functioneren.

Hierbij is door opdrachtnemer geadviseerd een cloud-applicatie te ontwikkelen om de inkomende verzoeken van IVR Manager af te handelen. Deze applicatie kan de gegevens vervolgens synchroniseren met het CRM Salesforce.com zodat aan alle eisen van de opdrachtgever is voldaan.

Voor het borgen van de continuïteit van deze cloud-applicatie en voor het borgen van de continuïteit van de mobiele applicatie is door opdrachtnemer een aantal maatregelen opgesteld. Deze maatregelen (uitvoerige documentatie, progressive enhancement, object-georiënteerde code, exception handling, proactieve controle, applicatiebeheerder) zorgen ervoor dat zowel de cloud-applicatie als de mobiele applicatie kan blijven draaien wanneer er veranderingen optreden in de onderliggende architectuur of veranderingen

Naast de achterliggende infrastructuur is door opdrachtnemer onderzocht of de “visual voicemail”-applicatie in de vorm van een native app of een web-app zal worden ontwikkeld. Conclusie van de door opdrachtnemer gedane vergelijking is, om de app als een hybrid app te ontwikkelen die zowel functionaliteiten van een native app als functionaliteiten en kenmerken van een webapp behelst. Er is gekozen voor een hybrid app, vanwege de cross-platform inzetbaarheid, de versnelde ontwikkeltijd en de mogelijkheid tot het aanspreken van de native functionaliteiten van de smartphone.

Om bovenstaande onderzoeksresultaten te valideren, is door opdrachtnemer een mobiele applicatie en een cloud-platform als proof-of-concept gerealiseerd. Hiermee heeft opdrachtnemer aangetoond dat het mogelijk is om een mobiele ‘visual voicemail’-applicatie te ontwikkelen die werkt volgens de door de opdrachtgever gestelde eisen.

Vanuit het door de opdrachtnemer uitgevoerde onderzoek en realisatie van de mobiele applicatie volgt een aantal aanbevelingen voor de opdrachtgever met betrekking tot de Greeting-dienst, de cloud-applicatie en het proof-of-concept (de mobiele applicatie voor Android).

Ten eerste adviseert de opdrachtnemer aan de opdrachtgever het proof-of-concept te verbeteren alvorens de mobiele applicatie op de markt te brengen. Hierbij valt te denken aan het toevoegen van de functionaliteiten: *importeren van contacten vanuit LinkedIn en Outlook, het opnemen van de begroetingen, en het versturen van notificaties.*

Tevens adviseert de opdrachtnemer aan de opdrachtgever om vanuit dit proof-of-concept ook een iOS-applicatie (en eventueel Windows Phone en Blackberry) te laten ontwikkelen om zodoende het grootste deel van de smartphone-markt aan te spreken.

Daarnaast adviseert de opdrachtnemer om de huidige Greeting-klanten de Greeting-dienst te laten evalueren en de hieruit volgende verbeteruggesties te implementeren.

Tot slot volgt vanuit de opdrachtnemer de aanbeveling om nader te onderzoeken welke aspecten/functionaliteiten van de Greeting-dienst, mobiele applicatie en de achterliggende cloud-applicatie nog verbeterd kunnen worden.

13. Evaluatie

In dit hoofdstuk wordt de projectaanpak, het projectproces, het gedane onderzoek, het gerealiseerde product en het eigen functioneren van de opdrachtnemer geëvalueerd.

13.1. Projectaanpak

Zoals van de opdrachtnemer kan worden verwacht, is gestart met het opstellen van een plan van aanpak, en het inventariseren van de eisen en wensen van de opdrachtgever. Na enkele dagen te hebben georiënteerd, is door de opdrachtnemer gestructureerd verslag gemaakt van de aanwezige voorzieningen en mogelijkheden met betrekking tot de huidige situatie. Hiermee heeft de opdrachtnemer antwoord gegeven op de onderzoekende deelvragen (deelvraag één tot en met drie).

Vervolgens is door de opdrachtnemer, in overleg met opdrachtgever, bepaald dat er niet alleen een mobiele applicatie, maar ook een cloud-applicatie ontwikkeld dient te worden, die de verzoeken vanuit IVR Manager afhandelt en synchroniseert met Salesforce.com/Database.com. Op iteratieve wijze zijn zowel de mobiele applicatie als de achterliggende cloud-applicatie op het Heroku-platform ontwikkeld. Tijdens het ontwikkelen is antwoord gekomen op de vraag hoe de (infra)structuur van de applicatie eruit dient te zien (deelvraag vier).

Tot slot is na ontwikkeling van de mobiele applicatie door opdrachtnemer gevalideerd of het gerealiseerde bedrijfsproduct aan de eisen en de wensen van de opdrachtgever, hiermee is antwoord verkregen op deelvraag vijf van het onderzoek. Hoewel het onderzoek en de realisatie van het bedrijfsproduct op een traditionele wijze (lees: watervalmethode) zijn uitgevoerd, is door opdrachtnemer desondanks antwoord gekregen hoe op de hoofdvraag van het onderzoek. Er is duidelijk geworden hoe een mobiele visual voicemail applicatie ontwikkeld kan worden die voldoet aan de eisen en wensen van de opdrachtgever. De opdrachtnemer heeft het theoretische antwoord op deze hoofdvraag in praktijk gebracht door een proof-of-concept op te leveren aan de opdrachtgever.

13.2. Projectproces

Nadat door opdrachtnemer is gestart met het inventariseren van de eisen en wensen van de opdrachtgever en deze functionele eisen in een plan van aanpak heeft vastgelegd, is de projectopzet dusdanig gewijzigd, dat er aan opdrachtnemer ook werkzaamheden zijn toebedeeld die aanvankelijk niet waren ingepland. Deze werkzaamheden lagen (in de ogen van de opdrachtnemer) buiten de scope van het onderzoek en buiten de scope van de ontwikkeling van de mobiele applicatie. Dit heeft geleid tot vertraging in de oplevering van het proof-of-concept en tevens tot vertraging in de verwerking van de onderzoeksresultaten.

13.3. Onderzoek

Hoewel het onderzoek, verricht door opdrachtnemer, van voldoende kwaliteit is voor een HBO-waardig afstudeerproduct, is er in de ogen van zowel opdrachtnemer als opdrachtgever te weinig aandacht besteed aan de reproduceerbaarheid van het onderzoek en haar onderzoeksresultaten.

Daarnaast is niet onbelangrijk te vermelden dat er vanuit de opdrachtgever weinig waarde wordt gehecht aan een onderzoek van deze omvang. Hoewel het uitvoeren van dit onderzoek, en hiermee het voortschrijdend inzicht, nuttige inzichten heeft opgeleverd voor de opdrachtnemer, heeft de opdrachtgever meerdere malen aangegeven weinig belang te hebben bij de door opdrachtnemer gevonden resultaten. Het heeft echter wel geleid tot het uiteindelijke bedrijfsproduct.

13.4. Gerealiseerd product

Het uiteindelijk gerealiseerde bedrijfsproduct is tweeledig. Enerzijds is er door opdrachtnemer een cloud-applicatie opgeleverd, anderzijds is door opdrachtnemer een mobiele applicatie opgeleverd.

De cloud-applicatie is, zowel voor opdrachtgever als opdrachtnemer, naar tevredenheid opgeleverd, zodat de Greetingq-diensten volledig aangeboden kunnen worden.

De mobiele applicatie voldoet aan de eisen van de opdrachtgever, maar wordt zowel door opdrachtgever als door opdrachtnemer nog niet als compleet beoordeeld. Hiertoe dienen de wensen van de opdrachtgever nog te worden geïmplementeerd.

13.5. Reflectie op functioneren opdrachtnemer

De opdrachtnemer heeft zich gedurende de afstudeerperiode verantwoordelijk opgesteld, en heeft alle voorkomende werkzaamheden opgepakt. Hierbij zijn ook werkzaamheden uitgevoerd die ver buiten de scope van het bedrijfsproduct en ver buiten de scope van het onderzoek vallen.

Desondanks is door opdrachtnemer een bedrijfsproduct opgeleverd die aansluit op de eisen van de opdrachtgever. Verder heeft de opdrachtnemer zich betrokken, professioneel en open opgesteld.

Hoewel door de opdrachtgever weinig aandacht is geschonken aan het door opdrachtnemer uitgevoerde onderzoek, heeft de opdrachtnemer getracht hier een balans in te vinden, door zowel de eigen belangen (afstuderen) als de belangen van de opdrachtgever (afronden van het bedrijfsproduct) centraal te stellen tijdens het uitvoeren van de dagelijkse werkzaamheden.

Door de brede blik van opdrachtnemer en de kennis en ervaring die door deze in de organisatie en de ontwikkeling van het bedrijfsproduct zijn ingebracht, kan gesteld worden dat de opdrachtnemer voldoet aan de eisen die redelijkerwijs van een HBO-afstudeerder verwacht mogen worden.

De conclusie hierin is dan ook dat de opdrachtnemer past in een vergelijkbare beroepssituatie en dat de verwachte werkzaamheden geen problemen zullen opleveren.

Bijlagen

Bijlage A	Gesprek met Sales Engineer van Massxess	7 maart 2013
Bijlage B	Bijeenkomst met Massxess (samenwerkingsbijeenkomst)	18 februari 2013
Bijlage C	Overleg met opdrachtgever MTTM omtrent eisen en wensen	4 februari 2013

Bijlage A. Gesprek met Massxess-engineer Edwin op 7 maart 2013

Type: Semi-gestructureerd telefonisch interview

Thijs: “We zijn bezig om voor het Greeting-platform (voicemail platform met persoonlijke begroetingen) een koppeling te maken tussen IVR Manager en Salesforce. Ik ben in dit proces verantwoordelijk voor het ontwikkelen van een mobiele applicatie voor Greeting, waarmee de voicemails visueel bekeken en beluisterd kunnen worden. Met behulp van een prompt in IVR Manager wordt er een XML request gestuurd naar ons platform en die kijkt of het nummer van de beller in ons systeem staat, zo ja, dan krijgt deze een persoonlijke voicemailbegroeting terug. Deze begroeting wordt in de XML response teruggegeven aan IVR Manager in de vorm van een promptcode (bijvoorbeeld 0001 of 0004).”

Thijs: “Bestaat er een mogelijkheid om vanaf ons platform deze voicemailprompts te uploaden naar de IVR Manager en na een succesvolle upload een promptcode terug te ontvangen?”

Edwin: “In principe bestaat er vanuit IVR Manager geen mogelijkheid om voicemailprompts anders dan via onze software (handmatig) te uploaden. Er is hiervoor geen webservice beschikbaar.”

Thijs: “Eigenlijk heb ik dezelfde vraag over het proces van opgeslagen voicemails: In hoeverre is het mogelijk om vanuit ons platform de opgeslagen voicemailbestanden op te halen (eventueel via FTP), zodat wij deze in onze mobiele applicatie kunnen gaan aanbieden?”

Edwin: “Ook hiervoor zijn geen mogelijkheden in de huidige versie van IVR Manager. Op dit moment kunnen voicemailbestanden gemailed worden vanuit de IVR Mail Module.”

Thijs: “Zijn er nog andere mogelijkheden om toch de gewenste functionaliteit te bereiken met IVR Manager?”

Edwin: “IVR Manager beschikt over de mogelijkheid om met behulp van XML-RPC een URL aan te roepen en hier variabelen naar toe te sturen, zoals nummer van de beller, en nummer van de gebelde. Op deze manier kan je de variabelen opslaan in je eigen applicatie. Je kunt ook variabelen terugsturen naar IVR Manager die je vervolgens in je IVR-script weer verder kunt gebruiken. XML versturen doe je door de XML module aan je IVR script toe te voegen en daar de URL en de gewenste variabelen in te geven.”

Thijs: “Kan ik met behulp van XML ook audio versturen? Zodat ik de begroeting dan kan opvragen uit mijn applicatie en IVR Manager deze dan kan afspelen?”

Edwin: “Ja, dat kan ook met behulp van de XML module. Dan moet je ervoor zorgen dat je het mp3-tje of wav-bestandje aanbiedt met base64-encoding, zodat IVR Manager het kan omzetten en moet je kiezen voor VOC-to-MP3 of VOC-to-WAV als je de een opgenomen gesprek wilt versturen naar je applicatie. Wil je een begroeting afspelen vanuit je applicatie dan moet je deze variabelen instellen als MP3-TO-VOC of WAV-to-VOC zodat deze in het door IVR Manager VOC-formaat kan worden afgespeeld.”

Thijs: “Samenvattend zou je dus kunnen zeggen dat IVR Manager geen ondersteunende bestandsopslag ondersteunt voor begroetingen, geen ondersteunende bestandsopslag ondersteunt voor voicemailbestanden, en dat er geen webservices beschikbaar zijn om deze vooraf of op een later moment dynamisch aan te leveren of uit te lezen. Wel kan XML gebruikt om tijdens het gesprek een begroeting op te vragen en af te spelen.”

Bijlage B. Samenvatting samenwerkingsbijeenkomst Massxess

Locatie: Massxess Almere

Datum: 18 februari 2013

Aanwezig: Bob van Dam (Commercieel Directeur Massxess)
Roy Overbeek (Engineer Massxess)
Edwin Winters (Sales Engineer Massxess)
2 andere medewerkers van Massxess Operations

Casper Gorter (IT Manager MTTM.nl)
David de Boer (Developer MTTM.nl)
Frank Boon (Developer MTTM.nl)
Thijs Otter (Afstudeerstagiair MTTM.nl – project Greeting)

Type: enerzijds presentaties van zowel Massxess als MTTM.nl als een discussie waarin plannen zijn gemaakt voor langdurige samenwerking

Samenvatting:

Massxess heeft een demonstratie gegeven van het nieuwe IpCCC platform, de opvolger van het softwarepakket IVR Manager. Met ipCCC is meer mogelijk voor eindgebruiker, maar voor de ontwikkelaar vallen de mogelijkheden tegen. Er zijn nog steeds geen uitgebreide webservices beschikbaar om diensten zoals Greeting gemakkelijk te ontwikkelen. Vanuit MTTM.nl werden de verschillende telefonie-applicaties getoond die door MTTM geïntegreerd zijn in het CRM Salesforce.

Met de engineers van de Operations afdeling is gesproken over het project Greeting. Daarbij werd duidelijk dat het niet mogelijk is om te achterhalen vanuit welke telefoon de Greeting-gebruiker naar het Greeting-doorschakelnummer is doorgeschakeld omdat dit tussen operators niet door wordt gegeven.

Na het bepreken van de grenzen van de dienst, is besproken dat Massxess actief ondersteuning ging bieden voor o.a. het ontwikkelen van de Greeting-dienst. Het doel van de bijeenkomst was voornamelijk kennismaken en globaal de plannen met betrekking tot een samenwerking voor de komende tijd. Het doel van de bijeenkomst lijkt te zijn behaald.

Bijlage C. Overleg met MTTM.nl over eisen en wensen Greeting

Datum: 4 februari 2013

Aanwezig: Thijs Otter (opdrachtnemer)
Casper Gorter (opdrachtgever namens de directie)

Type: ongestructureerd interview

Samenvatting:

MTTM.nl werkt op alle afdelingen met het CRM Salesforce.com. Het bedrijfsbeleid dat MTTM.nl hanteert gaat uit van het CRM Salesforce.com als leidraad door de gehele organisatie heen. Alle informatie in Salesforce, dat is het doel. Daarom is het een vereiste dat de instellingen en gegevens van de Greeting-dienst via Salesforce kunnen worden beheerd. Hieruit volgt o.a. dat de app gebruik dient te maken van een koppeling met het CRM Salesforce.com. Daarnaast dient de Greeting-dienst te werken met het telecomproduct IVR Manager.

Een aantal functionaliteiten waarover de app minimaal dient te beschikken:

De app dient gebruik te maken van het interne telefoonboek van de smartphone om de telefoonnummers te herkennen aan de hand van een lijst met contactpersonen. Tevens moet het mogelijk zijn om de door een beller ingesproken voicemails te beluisteren via de mobiele app.

Ook zou het wenselijk zijn als de applicatie een doorschakeling instelt na de eerste start of tijdens installatie, dit neemt de gebruiker het instelwerk uit handen. Daarnaast is een wens van de directie om de contacten in Greeting te kunnen importeren vanuit LinkedIn en Outlook.

Tenslotte zijn nog enkele toekomstige wensen voor de Greeting-applicatie uitgesproken: enerzijds het gebruik maken van de functionaliteit om begroetingen op te nemen, en anderzijds het versturen van notificaties wanneer er een voicemailbericht binnenkomt.

Bronvermelding

-
- ¹ Ontleend aan: "Website MTTM Group, www.mttmgroup.nl, bezocht op 17 april 2013."
 - ² Fling, B. (2009). Mobile Design and Development (eerste druk). Sebastopol (Canada): O'Reilly
 - ³ UML: Unified Modeling Language. Opgevraagd op 5 februari 2013, van <http://www.uml.org/>
 - ⁴ Gesprek met Massxess heeft plaatsgevonden op 18 februari 2013. Gespreksverslag hiervan is te vinden in bijlage B.
 - ⁵ Salesforce.com. SOAP API Developer's Guide. Bezocht op 25 april 2013, van <http://www.salesforce.com/us/developer/docs/api/index.htm>
 - ⁶ Salesforce.com. Force.com REST API Developer's Guide. Bezocht op 25 april 2013, van http://www.salesforce.com/us/developer/docs/api_rest/
 - ⁷ Wikipedia.org. Representational State Transfer. Bezocht op 25 april 2013, van https://en.wikipedia.org/wiki/Representational_state_transfer
 - ⁸ Salesforce.com. SOAP API Developer's Guide. Bezocht op 25 april 2013, van <http://www.salesforce.com/us/developer/docs/api/index.htm>
 - ⁹ Salesforce.com. Salesforce App Limits Cheatsheet. Bezocht op 25 april 2013, van https://login.salesforce.com/help/doc/en/salesforce_app_limits_cheatsheet.pdf
 - ¹⁰ Sapra, J. HTML5 vs Native App: The Tough Battle. Bezocht op 10 mei 2013, van <http://www.toolsjournal.com/mobile-articles/item/432-html5-vs-native-app-the-tough-battle>
 - ¹¹ Android, Notifications | Android Developers, Bezocht op 10 mei 2013, van <https://developer.android.com/guide/topics/ui/notifiers/notifications.html>
 - ¹² Wikipedia.org. HTML5 in mobile devices. Bezocht op 25 april 2013, van http://en.wikipedia.org/wiki/HTML5_in_mobile_devices#Frameworks_for_app_development
 - ¹³ UML. UML Logo. Bezocht op 25 april 2013, van <http://www.uml.org/images/logos/UML-logo.jpg>
 - ¹⁴ Heroku. Heroku Addons. Bezocht op 26 april 2013, van <https://addons.heroku.com/>