

**Bachelorproject Media en Kennistechnologie (IN3405)**

Faculteit Elektrotechniek, Wiskunde en Informatica, TU Delft

# Een iPhone-webapplicatie voor Greeting Voicemail

*26 juni 2009*

**Auteurs**

Gijs van Tulder 1339389  
Roland Heinrichs 1215655

**Beoordelingscommissie**

R. Stronkman  
Prof. dr. L.J.M. Rothkrantz  
Ir. M. Sepers

Greeting, begeleider  
TU Delft, begeleider  
TU Delft, BSc.-coördinator





# Voorwoord

---

Als afsluiting van de bacheloropleiding Media en Kennistechnologie aan de TU Delft dient een bachelorproject gevolgd te worden. Het project vindt plaats als stage bij een extern bedrijf en vindt plaats in de laatste drie maanden van het derde jaar. De doelstelling van het project is niet alleen het toepassen van de in de bacheloropleiding geleerde theorie, maar ook de ervaring van het werken in een groep bij een externe opdrachtgever.

De bachelor Media en Kennistechnologie kent meerdere onderzoeksgebieden: Computer Graphics, Intelligent Information Processing en Man-Machine Interaction. Dit project behoort tot de groep voor Man-Machine Interaction. Tijdens het project is een applicatie ontwikkeld die de voicemail van Greeting beschikbaar maakt op de iPhone. Er is gekozen voor de iPhone omdat deze telefoon een aantal bijzondere mogelijkheden heeft, die tijdens dit project zijn gebruikt.

Het project is uitgevoerd bij het bedrijf Greeting in Delft. Dit bedrijf ontwikkelt persoonlijke voicemail. De opdrachtgever van het project vanuit Greeting was Richard Stronkman. Wij willen hem dan ook graag bedanken voor zijn hulp en zijn ideeën over het ontwikkelproces en het eindproduct. In de eerste weken van het project werden we vanuit de universiteit begeleid door dr. Stijn Oomes. Toen hij ons niet meer kon begeleiden vonden we in prof. dr. Leon Rothkrantz een goede vervanger. We willen hen beiden graag hartelijk bedanken voor hun adviezen tijdens dit project.



# Inhoudsopgave

---

<b>Voorwoord</b>	<b>1</b>
<b>Samenvatting</b>	<b>7</b>
<b>1 Inleiding</b>	<b>9</b>
<b>2 De opdracht</b>	<b>11</b>
2.1 Probleemdefinitie . . . . .	11
2.2 Maatschappelijke relevantie . . . . .	11
2.3 Methodologie . . . . .	12
2.4 Wetenschappelijke uitdagingen . . . . .	12
2.5 De keuze voor een iPhone-webapplicatie . . . . .	13
<b>3 Het ontwerpproces</b>	<b>15</b>
3.1 Eiseninventarisatie . . . . .	15
3.1.1 Doelgroep . . . . .	15
3.1.2 Doelen . . . . .	16
3.1.3 Omgeving . . . . .	16
3.1.4 Usability . . . . .	16
3.1.5 Gebruikerservaring . . . . .	16
3.1.6 Functionaliteiten . . . . .	17
3.2 Scenario's en use cases . . . . .	17
3.2.1 Scenario's . . . . .	17
3.2.2 Use cases . . . . .	19
3.3 Vormgeving . . . . .	20
3.3.1 Hiërarchie van de applicatie . . . . .	20
3.3.2 Schetsen . . . . .	22
<b>4 Interface</b>	<b>25</b>
4.1 Ontwerpkeuzes . . . . .	25
4.1.1 Berichtenlijst en berichten . . . . .	25
4.1.2 Contactenlijst en contacten . . . . .	26
4.1.3 Begroetingenlijst en begroetingen . . . . .	26
4.1.4 Menu, inloggen en instellingen . . . . .	26
4.2 Prototypes . . . . .	27
4.2.1 Clickable 1.0 . . . . .	27
4.2.2 Clickable 1.5 . . . . .	27
4.2.3 Clickable 1.6 . . . . .	28
4.2.4 Clickable 2.0 . . . . .	29

<b>5</b>	<b>Gebruikerstests</b>	<b>33</b>
5.1	Testopzet . . . . .	33
5.2	Testpersonen . . . . .	34
5.2.1	Eerste testpersoon . . . . .	34
5.2.2	Tweede testpersoon . . . . .	35
5.2.3	Derde testpersoon . . . . .	35
5.3	Uitvoering van de tests . . . . .	35
5.3.1	Over de test . . . . .	35
5.3.2	Over Greeting Voicemail . . . . .	35
5.3.3	Testcase 1: Verkenning . . . . .	36
5.3.4	Testcase 2: Nieuwe gebruiker . . . . .	36
5.3.5	Testcase 3: Gevorderde gebruiker . . . . .	37
5.3.6	Testcase 4: Ontwerp . . . . .	37
5.4	Evaluatie . . . . .	38
5.5	Conclusie . . . . .	40
<b>6</b>	<b>Implementatie en softwaretests</b>	<b>41</b>
6.1	Greeting API en Business Logic Layer . . . . .	41
6.2	Overzicht van de webapplicatie . . . . .	42
6.3	Details van de implementatie . . . . .	43
6.3.1	Models . . . . .	43
6.3.2	Controllers . . . . .	46
6.3.3	Views . . . . .	47
6.4	Nieuwe begroetingen opnemen . . . . .	49
6.5	Tests . . . . .	50
6.5.1	Voorbeeld . . . . .	51
6.6	Conclusie . . . . .	52
<b>7</b>	<b>Conclusie en aanbevelingen</b>	<b>53</b>
7.1	Aanbevelingen . . . . .	54
	<b>Appendices</b>	<b>55</b>
<b>A</b>	<b>Planning</b>	<b>55</b>
<b>B</b>	<b>Klassendiagrammen</b>	<b>57</b>
<b>C</b>	<b>Clickable 1.6</b>	<b>61</b>
<b>D</b>	<b>Oriëntatieverslag</b>	<b>63</b>
D.1	Inleiding . . . . .	63
D.2	Greeting . . . . .	63
D.2.1	Het bedrijf en het product . . . . .	63
D.2.2	Over de opdracht . . . . .	64
D.3	iPhone . . . . .	65
D.3.1	Ontwerpen voor de gebruiker . . . . .	65
D.3.2	Adviezen van Apple . . . . .	65
D.3.3	Beperkingen . . . . .	66

D.3.4	Ontwerpen voor de iPhone . . . . .	67
D.4	Safari Mobile . . . . .	68
D.4.1	Kenmerken en beperkingen . . . . .	68
D.5	Framework . . . . .	71
D.5.1	Criteria . . . . .	72
D.5.2	Struts 2 . . . . .	72
D.5.3	Spring MVC . . . . .	72
D.5.4	Vergelijking . . . . .	72
D.6	Conclusie . . . . .	73
<b>E</b>	<b>Plan van aanpak</b> . . . . .	<b>75</b>
E.1	Managementsamenvatting . . . . .	75
E.2	Inleiding . . . . .	75
E.2.1	Aanleiding . . . . .	75
E.2.2	Accordering en bijstelling . . . . .	75
E.2.3	Toelichting op de opbouw van het plan . . . . .	75
E.3	Projectopdracht . . . . .	75
E.3.1	Projectomgeving . . . . .	75
E.3.2	Doelstelling project . . . . .	76
E.3.3	Opdrachtformulering en op te leveren product . . . . .	76
E.3.4	Eisen en beperkingen . . . . .	76
E.3.5	Cruciale succesfactoren . . . . .	76
E.3.6	Aanpak . . . . .	76
E.4	Projectinrichting en voorwaarden . . . . .	77
E.4.1	Projectinrichting . . . . .	77
E.4.2	Voorwaarden aan opdrachtnemer . . . . .	77
E.4.3	Voorwaarden aan opdrachtgever . . . . .	78
E.5	Planning . . . . .	78
E.6	Kwaliteitsborging . . . . .	78
E.6.1	Productkwaliteit . . . . .	78
E.6.2	Proceskwaliteit . . . . .	78
E.6.3	Voorgestelde maatregelen . . . . .	79





# Samenvatting

---

Dit is het verslag van het bachelorproject bij Greeting. Tijdens dit project is een mobiele webapplicatie voor de iPhone ontwikkeld en getest. Met deze applicatie kunnen gebruikers van Greeting Voicemail via een visuele interface hun voicemail beluisteren en beheren.

Greeting maakt een product voor persoonlijke voicemail. Gebruikers kunnen voor iedere beller een persoonlijke begroeting instellen. Ook is het mogelijk om per land een begroeting te maken, bijvoorbeeld om internationale klanten een Engelstalige begroeting te laten horen. Greeting Voicemail was alleen via een telefoonmenu bereikbaar. Dat maakte het voor gebruikers lastig en omslachtig om de persoonlijke begroetingen te beheren. De visuele interface die tijdens dit project is gebouwd maakt dit een stuk makkelijker.

Om de interface zo herkenbaar mogelijk te maken voor iPhone-gebruikers, is het ontwerp gebaseerd op de lay-out van andere iPhone-applicaties. Ook zijn zo veel mogelijk de standaardelementen van de iPhone gebruikt.

Om het gebruikersgemak te controleren, is de interface tijdens het project getest met een aantal ervaren iPhone-gebruikers. De deelnemers konden over het algemeen goed met de applicatie overweg. Ze vonden de applicatie plezierig en gemakkelijk in het gebruik. De onderdelen die nog een probleem vormden zijn met de uitkomsten van de tests verbeterd.

Na de ontwerpfase is het ontwerp geïmplementeerd en gekoppeld aan de bestaande backend van Greeting. De webapplicatie die de backend met de interface verbindt is met behulp van speciaal gebouwde unit-tests automatisch getest. In het ontwerp van de implementatie is rekening gehouden met een toekomstige uitbreiding naar andere telefoons.

Op advies van de opdrachtgever bestond de eerste helft van het project uit het ontwerpen van de interface. Pas in de tweede helft van het project is gekeken naar de aansluiting op het bestaande systeem van Greeting. Doordat de mogelijkheden van het systeem in de ontwerpfase nog niet bekend waren, vormden de wensen van de gebruiker, en niet de beperkingen van het systeem, in deze fase het uitgangspunt. Uiteindelijk is in de tweede fase een deel van de ideeën geschrapt en is een deel van de ideeën toegevoegd aan Greeting Voicemail.

Het project is uitgevoerd volgens een iteratieve methode. Tijdens de ontwerpfase zijn verschillende ontwerpen gemaakt. Het commentaar van de opdrachtgevers en de resultaten uit de tests zijn gebruikt om het ontwerp in een volgende iteratie te verbeteren. Door deze manier van werken was tijdens het project voor alle betrokkenen duidelijk wat de applicatie zou gaan doen, zodat op een zinvolle manier met de opdrachtgevers kon worden overlegd.

Uiteindelijk heeft dit project een werkende applicatie opgeleverd, die door Greeting in de toekomst aan de gebruikers zou kunnen worden aangeboden.



# Inleiding 1

---

Dit is het verslag van het bachelorproject bij Greeting. De opdracht van het project was het ontwerpen van een iPhone-webapplicatie voor Greeting Voicemail. Met deze applicatie moeten iPhone-gebruikers via het scherm van hun mobiele telefoon hun voicemail kunnen afluisteren en hun begroetingen kunnen beheren.

Greeting maakt een product voor persoonlijke voicemail. In plaats van één begroeting voor alle bellers, kan de gebruiker van Greeting Voicemail meerdere persoonlijke begroetingen instellen. De begroetingen kunnen worden gekoppeld aan het nummer van de beller. Zo kan de gebruiker bijvoorbeeld voor zijn vrienden een andere begroeting inspreken dan voor zijn collega's. Het is ook mogelijk om de begroetingen per land in te stellen: een Nederlandse begroeting voor Nederlandse relaties, een Engelse voor alle internationale bellers.

Voor het afluisteren van berichten en het instellen van de begroetingen heeft Greeting op dit moment een telefoonmenu. Met een visuele interface zou het gebruiksgemak van Greeting Voicemail kunnen worden verbeterd. Greeting heeft van deze opdracht een bachelorproject gemaakt.

Vanwege de bijzondere mogelijkheden van de iPhone wil Greeting de applicatie eerst voor deze telefoon ontwikkelen. De iPhone biedt ruime mogelijkheden voor het ontwikkelen van gebruiksvriendelijke webapplicaties. Het is de eerste van een nieuwe generatie mobiele telefoons die ondanks de nadelen van het kleine formaat toch ruime mogelijkheden bieden. Met een groter scherm, betere invoermogelijkheden en grote aandacht voor de gebruikservaring wordt het plezier en gemak van deze telefoons steeds groter.

Dit rapport is als volgt opgebouwd. In hoofdstuk 2 worden de opdracht en de probleemstelling uitgewerkt. Het derde hoofdstuk beschrijft het ontwerpproces en de gemaakte keuzes. In het vierde hoofdstuk wordt beschreven hoe de interface is ontworpen en gebouwd. In het vijfde hoofdstuk zijn de usability-tests uitgewerkt. Hoofdstuk zes is een verslag van de aansluiting van de interface op de backend van Greeting. In hoofdstuk 7 staan de conclusie van het project en enkele aanbevelingen die wij naar aanleiding van het project aan Greeting doen.

Naast de hoofdstukken zijn nog enkele bijlages opgenomen. In bijlage A staat de planning van het project. Bijlage D is het oriëntatieverslag waarin staat beschreven wat Greeting is, wat de iPhone kan en welke koppelingsmogelijkheden er zijn. Bijlage E is het plan van aanpak, dat uitgebreid ingaat op de planning, de benodigde hard- en software en de methodologie.



# De opdracht 2

---

In dit project moet een applicatie worden ontwikkeld waarmee Greeting Voicemail met een iPhone kan worden bediend. In dit hoofdstuk wordt deze opdracht uitgewerkt in de probleemdefinitie. Vervolgens wordt kort ingegaan op de maatschappelijke relevantie, de te gebruiken methodologie en de wetenschappelijke uitdagingen van het project. Tot slot wordt kort toegelicht waarom is gekozen voor implementatie als een iPhone-webapplicatie.

## 2.1 Probleemdefinitie

Greeting heeft de opdracht gegeven om een mobiele interface te ontwerpen voor hun product, Greeting Voicemail. De interface moet specifiek voor de iPhone worden ontworpen en moet voldoen aan de volgende eisen:

1. Een iPhone-gebruiker kan het systeem zonder toelichting gebruiken.
2. Met het product kunnen alle functies van Greeting Voicemail worden gebruikt.
3. Het eindproduct is volledig geïmplementeerd en aangesloten op de backend van Greeting.
4. De implementatie is goed gedocumenteerd en getest.

## 2.2 Maatschappelijke relevantie

De maatschappelijke relevantie van ons product, de webapplicatie, komt voort uit de relevantie van het product Greeting Voicemail. De functionaliteiten van Greeting Voicemail staan beschreven in hoofdstuk 2 van het oriëntatieverslag (bijlage D). Voicemail bestaat al langer en is zeker niet nieuw. Het is echter een product dat nog vele uitbreidingsmogelijkheden heeft. Greeting wil ervoor zorgen dat voicemail een interessant product wordt. Door de mogelijkheden flink te verruimen wordt voicemail een stuk nuttiger en leuker. Zo maakt Greeting het mogelijk om persoonlijke begroetingen in te stellen, of om bijvoorbeeld iedereen die vanuit een Engels land belt in het Engels te beantwoorden.

De huidige applicatie van Greeting gebruik je via een telefoonmenu. Je belt op naar de server en luistert je berichten daar één voor één af. Daarbij zie je van tevoren niet hoe lang het bericht is, wie er gebeld heeft, of iemand vaker heeft gebeld etc. Het overzicht ontbreekt. Een ander probleem zijn de niet ingesproken berichten: mensen krijgen je voicemail aan de lijn, horen de piep, maar spreken vervolgens niets in. Er staat dan wel een bericht in je voicemail, maar daar is niets op te horen. Via het telefoonmenu moet je het dan toch afluisteren, omdat je niet kunt zien hoe lang het bericht is.

Het overzicht dat onze webapplicatie biedt maakt voicemail makkelijker te gebruiken. Als je inlogt zie je in een oogopslag je ingekomen berichten. Je ziet welke berichten nieuw zijn en welke onbeluisterd zijn. De lengte van het bericht staat achter ieder bericht weergegeven. Daarnaast is het niet alleen mogelijk om de beller terug te bellen, maar kan ook in één keer per sms worden gereageerd. Of per e-mail, als het e-mailadres bekend is. Verder zorgt

de webapplicatie ervoor dat alle functionaliteiten van Greeting Voicemail mobiel bereikbaar zijn. Een deel van de mogelijkheden is nu niet via het telefoonmenu beschikbaar, maar alleen via de webapplicatie die je alleen thuis kunt gebruiken. Met een visuele mobiele interface kun je vlak voor, of zelfs tijdens, een vergadering de voicemail nog snel even instellen op ‘vergadering’.

## 2.3 Methodologie

In het onderstaande schema staat kort welke stappen we in het project hebben ondernomen en wat het doel van die stappen was.

Actie	Doel
Brainstorm	Ideeën verzamelen.
Oriëntatie	Mogelijkheden iPhone en Greeting Voicemail onderzoeken.
Papieren ontwerp	Ideeën uitwerken en lay-out bedenken.
Clickable 1.	Functionaliteit testen.
Clickable 1.6	Ontwerp verbeteren.
Clickable 2.0	Bruikbaarheid testen.
Integratie	Koppeling met Greeting maken en testen.

Tabel 2.1: De stappen van het project met hun doel.

De opdracht van Greeting is om ons eerst volledig te richten op het ontwerp en ons daarna pas bezig te houden met de implementatie. De brainstorm is gehouden zonder verdere detailkennis van het product van Greeting. Ideeën die binnen het systeem van Greeting niet mogelijk waren zijn daarna geschrapt, maar er is ook een aantal functionaliteiten, zoals contactbeheer, toegevoegd aan Greeting Voicemail.

Het ontwerpproces is deels geïnspireerd door het boek *Getting real* [1], dat ons ook door Richard Stronkman is aangedragen. De schrijvers van het boek beginnen met het ontwerpen van de interface voordat ze beginnen met programmeren. Door het concrete, iteratieve ontwerp is al vroeg in het project duidelijk wat de functionaliteiten van het product zullen worden. Dat maakt het makkelijker om daar met de opdrachtgever over te overleggen. Een tweede strategie uit het boek is het zogeheten ‘epicenter design’. Bij het ontwerpen denk je eerst wat de belangrijkste informatie is en hoe je die het beste op het scherm krijgt. Pas daarna kijk je naar de andere onderdelen van het scherm, zoals de navigatieknoppen of eventuele andere informatie. Deze methode is bijzonder geschikt voor de iPhone, omdat de grootte van het scherm beperkt is en alleen plaats biedt aan de belangrijkste onderdelen.

## 2.4 Wetenschappelijke uitdagingen

De uitdagingen van dit project zitten vooral in de mogelijkheden en de beperkingen van de iPhone. In het oriëntatieverslag [2] zijn deze uitgewerkt. Voor een aantal problemen zullen oplossingen moeten worden gezocht. Maar bovenal zal er een webapplicatie ontwikkeld moeten worden voor een beperkt scherm met beperkte invoermogelijkheden.

In dit project moet het gebruiksgemak van Greeting Voicemail worden verbeterd. Het telefoonmenu biedt slechts beperkte mogelijkheden voor een goede gebruikerservaring. Daarom

wordt een visuele interface voor de iPhone ontworpen, waarmee gebruikers eenvoudig hun berichten en begroetingen kunnen beheren. Vooral ingewikkelde handelingen als het instellen van een begroeting, die kan worden gekoppeld aan telefoonnummers of landen, kunnen met een visuele interface eenvoudiger verlopen. Maar ook voor een snel overzicht van alle nieuwe berichten kan een visuele interface een uitkomst bieden.

De beperkte schermgrootte van een mobiele telefoon, de beperkte invoermogelijkheden en de haast waarmee het mobiele product zal worden gebruikt, stellen bijzondere eisen aan de interface. Deze beperkingen gelden voor bijna alle smartphones, palmtops, etc. Binnen dit segment scheidt de keuze voor de iPhone extra mogelijkheden, omdat deze telefoon ruime ondersteuning biedt voor het ontwikkelen van een mooie mobiele applicatie. De voor- en nadelen van de keuze voor de iPhone zijn in de volgende paragraaf verder uitgewerkt.

Als de interface is ontwikkeld, zal die goed moeten worden getest. De onbekendheid van de ontwikkelaars met het platform, en met de doelgroep, zorgt er voor dat tests onontbeerlijk zijn om te controleren of de applicatie voldoet aan de eisen van de gebruikers en of de gebruikers op een eenvoudige manier met de applicatie kunnen werken. Er zullen dan ook goede gebruikerstests moeten worden gedaan om te controleren of dit is gelukt en wat er moet worden verbeterd.

## 2.5 De keuze voor een iPhone-webapplicatie

Aan het begin van het project is gekozen om de applicatie als webapplicatie voor de iPhone te ontwikkelen. De applicatie had als iPhone-applicatie kunnen worden ontwikkeld. Daar is niet voor gekozen, omdat aan de bouw van een 'echte' iPhone-applicatie een aantal problemen kleven. Ten eerste zou de iPhone-applicatie moeten worden goedgekeurd door Apple voordat de applicatie voor gebruikers beschikbaar zou worden. Omdat Apple zelf al een product voor visual-voicemail heeft, is onzeker of de Greeting-applicatie zou worden goedgekeurd. Ten tweede bestaat er een patent voor 'visual-voicemail op een telefoon', waarvoor Greeting nog geen licentie heeft. Dit patent zou niet gelden voor een webapplicatie, die immers niet draait op een mobiele telefoon. Een derde reden voor de keuze voor een webapplicatie is de mogelijkheid om de applicatie ook op andere telefoons te gebruiken. Een webapplicatie is makkelijker geschikt te maken voor een andere telefoon dan een iPhone-applicatie.

De keuze voor een webapplicatie heeft een aantal nadelen. Zo is het minder eenvoudig om gebruik te maken van de ingebouwde mogelijkheden van de iPhone. De webapplicatie werkt daarom minder 'soepel' dan een iPhone-applicatie. Daarnaast is het werken met geluid bij een webapplicatie een probleem. Geluidsbestanden, zoals berichten, kunnen alleen full-screen worden afgespeeld. Opnemen van geluid, bijvoorbeeld een nieuwe begroeting, is via een webpagina zelfs helemaal niet mogelijk. Daarvoor is dus een andere oplossing nodig.

Ondanks die nadelen biedt de iPhone een bruikbaar platform voor het ontwikkelen van webapplicaties. Safari Mobile, de webbrowser van de iPhone, biedt verschillende functies waarmee webapplicaties meer op een iPhone-applicatie kunnen gaan lijken. In het oriëntatieverslag wordt daarvan een overzicht gegeven.





# Het ontwerpproces 3

---

Dit hoofdstuk beschrijft het eerste ontwerpproces. Dat eerste ontwerpproces bestond uit drie stappen. Eerst is een eiseninventarisatie uitgevoerd, vervolgens zijn scenario's en use cases opgesteld en uiteindelijk zijn op papier ontwerpen gemaakt.

## 3.1 Eiseninventarisatie

Tijdens de eiseninventarisatie worden de beoogde functionaliteiten van het product vastgesteld. De opzet van onze inventarisatie komt uit het boek [3, hoofdstuk 10] en bestaat uit zes onderdelen. Eerst wordt de doelgroep bepaald. Er wordt beschreven welke eisen de doelgroep stelt aan het product en hoe de doelgroep is opgebouwd. In de tweede stap wordt in de doelen aangegeven wat de gebruiker met het product wil. In de derde stap wordt de omgeving besproken waarin de applicatie gebruikt zal worden. In stap vier en vijf worden de bruikbaarheid en de verwachtingen van de gebruiker bepaald. Ten slotte worden de functionaliteiten van de applicatie opgesomd. Deze opsomming geeft een overzicht van de eisen waaraan het eindproduct moet voldoen.

### 3.1.1 Doelgroep

Greeting richt zich met het product eerst op de zakelijke markt. De opdracht is dan ook om de applicatie voor de zakelijke gebruiker van de iPhone te ontwerpen.

We onderscheiden drie kwaliteitseisen die de doelgroep aan de applicatie zal stellen.

*Bereikbaarheid* De gebruikers uit de doelgroep zijn veel onderweg en communiceren daarvoor vooral via de mobiele telefoon. Bovendien zijn ze vaak in bespreking of vergadering, waardoor ze hun telefoon niet altijd direct kunnen beantwoorden.

*Efficiëntie* De gebruikers uit de doelgroep zullen niet altijd ruim de tijd hebben om de gemiste oproepen te bekijken en te beluisteren. Het is dus belangrijk dat ze snel een duidelijk overzicht krijgen van de gemiste oproepen, zodat ze in één oogopslag kunnen beoordelen wat belangrijk is en wat even kan wachten. Voor dit overzicht en voor het correct te woord staan van contacten is het contactbeheer ook erg belangrijk.

*Bruikbaarheid* De applicatie moet makkelijk toegankelijk en betrouwbaar zijn. De structuur van het programma dient overzichtelijk te zijn en het programma moet altijd goed blijven werken. De hoge kwaliteit van de ingebouwde iPhone-applicaties zorgt voor hoge verwachtingen bij de gebruiker.

Om te kunnen bepalen hoe de gebruiker omgaat met het product hebben we de doelgroep in drie categorieën onderverdeeld:

*De verplichte gebruiker* moet met deze voicemailapplicatie werken omdat het product door zijn baas is aangeschaft, maar hij zit er zelf niet op te wachten. Hij wil graag daarom graag een eenvoudige applicatie waarmee hij snel en zonder gedoe zijn voicemail kan afluisteren.

*De zakelijke gebruiker* heeft zelf voor de voicemailapplicatie gekozen, omdat hij het in zijn werk goed kan gebruiken. Hij heeft dan ook vooral belangstelling voor de functionaliteit en de bruikbaarheid van de applicatie. Hij wil zich best even in de applicatie verdiepen als dat het werken sneller maakt.

*De gadget-freak* heeft het product misschien niet per se nodig, maar gebruikt het product vooral omdat hij het leuk vindt om dit soort nieuwe snuffes te gebruiken. Hij wil graag alle onderdelen van de applicatie uitproberen en vindt het ook leuk om zijn speeltje aan anderen te laten zien.

### 3.1.2 Doelen

De gebruiker gebruikt de applicatie met, in volgorde van relevantie, de volgende doelen:

1. Zijn voicemailberichten mobiel kunnen beluisteren en beheren;
2. Zijn persoonlijke begroetingen mobiel kunnen beheren/instellen;
3. Zijn overige Greeting-instellingen mobiel kunnen beheren.

### 3.1.3 Omgeving

De omgeving waarin de applicatie gebruikt zal worden heeft een duidelijke invloed op het ontwerp. We onderscheiden voor de iPhone-applicatie van Greeting Voicemail de volgende omgevingsfactoren:

*Fysiek* De applicatie wordt regelmatig onderweg gebruikt, bijvoorbeeld in de trein. De applicatie wordt daarom regelmatig gebruikt in situaties waarin het afluisteren van een voicemailbericht niet handig of niet gepast is.

*Sociaal* Het is voor de gebruikers belangrijk om de mensen die hen bellen op de juiste manier te woord te staan. Als de beller een bericht achterlaat, is het ook belangrijk dat de gebruiker terugbelt en het bericht niet vergeet.

*Technisch* De voicemailapplicatie wordt gebruikt op de iPhone. Er is dus een internetverbinding, de snelheid is sterk afhankelijk van het netwerk. De iPhone heeft een touch-screen. Het scherm is minder groot dan dat van een gewone computer.

### 3.1.4 Usability

De usability is de bruikbaarheid van het product. Deze hebben wij getoetst aan zes usability-aspecten uit hoofdstuk 1 van het boek [3]. De zes aspecten worden beschreven in figuur 3.1. We hebben er voor gekozen om hier de Engelse termen uit het boek aan te houden.

Eén van de belangrijkste eisen aan ons product is het gebruiksgemak. Dit betekent dat voor ons vooral de learnability en de memorability goed moeten zijn. De iPhone heeft echter een aantal beperkingen die de efficiency verlagen. Het kleine scherm geeft weinig mogelijkheden voor uitgebreide menu's of gedetailleerde overzichten. Voor sommige opties zal daarom een paar keer geklikt moeten worden. De navigatiestructuur en de keuze van de knoppen in het scherm, die later in het verslag behandeld zullen worden, zijn duidelijke beïnvloed door van de beperkingen die worden opgelegd aan de bruikbaarheid.

### 3.1.5 Gebruikerservaring

In [3, hoofdstuk 1] wordt ook een aantal punten gegeven waaraan de gebruikerservaring kan worden getoetst. Wij hebben ons gericht op het gebruiksgemak (de andere punten, zoals 'fun', zijn voor deze applicatie minder van belang). De gebruiker moet het makkelijk

*Effectiveness* verwijst naar hoe goed het product doet wat het moet doen.  
*Efficiency* verwijst naar hoe goed het product de gebruiker helpt in het uitvoeren van zijn doelen.  
*Safety* verwijst naar hoe goed het product voorkomt dat de gebruiker een fout maakt en hoe makkelijk de gebruiker een eventuele fout weer kan herstellen.  
*Utility* verwijst naar in hoeverre het product de gebruiker de juiste functionaliteiten geeft om zijn taak uit te voeren.  
*Learnability* verwijst naar hoe gemakkelijk een nieuwe gebruiker het product kan leren gebruiken.  
*Memorability* verwijst naar hoe gemakkelijk de gebruiker kan onthouden hoe een product werkt, als hij die werking eenmaal heeft geleerd.

Figuur 3.1: Zes usability-aspecten. Vrij vertaald uit [3].

en plezierig vinden om zijn voicemail via de applicatie te beluisteren. Daarbij gaat het ook om de tegenstelling tot bestaande voicemailsysteem. De gebruikers moeten nu bellen met het voicemailsysteem waarna ze in een zeer beperkt menu uitkomen. Vaak wordt dit soort menu's als onplezierig ervaren.

### 3.1.6 Functionaliteiten

Op basis van de eiseninventarisatie is een gedetailleerde lijst opgesteld van de gewenste functionaliteiten van het product. Een fragment van de lijst van functionaliteiten is weergegeven in figuur 3.2. Hierin zijn de functionaliteiten minder belangrijke functionaliteiten schuingedrukt weergegeven. De lijst functionaliteiten is doorgenomen met de opdrachtgevers. Op basis van de bestaande mogelijkheden van Greeting Voicemail en de onderdelen die op korte termijn zullen worden toegevoegd, zijn sommige functionaliteiten aangepast of helemaal weggelaten. De volledige lijst van functionaliteiten staat in het document eiseninventarisatie [4].

## 3.2 Scenario's en use cases

De scenario's en use cases zijn beschreven in het document scenario's [5]. De scenario's en use cases zijn geschreven om een goed beeld te kunnen krijgen van het gebruik van de applicatie. De eerste versie is aangepast na commentaar van de opdrachtgever.

In dit verslag staan slechts enkele voorbeelden om te laten zien hoe onze scenario's en use cases er uit zien. Voor de volledige scenario's en use cases verwijzen wij naar [5].

### 3.2.1 Scenario's

De scenario's beschrijven verschillende situaties waarin Greeting Voicemail op de iPhone gebruikt zou kunnen worden.

**Berichten**

- Nieuwe berichten zien
  - Beller
  - Datum
  - Nieuw/beluisterd
  - Wel/geen bericht ingesproken
  - *Lengte van het bericht weergeven*
  - *Ordering/indeling/relevantie*
- Berichten afluisteren
- Berichten verwijderen
- Beller terugbellen
- *Beller sms'en*
- *Berichten archiveren*
- *Gearchiveerde berichten zien*
- *Zoeken binnen gearchiveerde berichten*

Figuur 3.2: Een deel van de lijst van functionaliteiten.

Je komt uit de vergadering en je wilt kijken of je nog gebeld bent. Je start Greeting Voicemail en je ziet dat je drie nieuwe berichten hebt. Je klikt op het eerste bericht om het te beluisteren. Je wilt op dit bericht reageren en besluit om even terug te bellen.

Nadat je gebeld hebt kom je weer terug bij je voicemailberichten. Bij het bericht dat je net hebt beluisterd is nu te zien dat je inmiddels hebt teruggebeld. Je beluistert daarom het volgende bericht op de lijst. Dit bericht bestaat uit een vraag waarop een kort en simpel antwoord mogelijk is. Je besluit de beller een sms te sturen.

In het scherm wordt aangegeven dat je als reactie op dit bericht een sms verstuurd hebt. Je beluistert het laatste bericht en je besluit als reactie een e-mail te sturen.

Nadat je de e-mail hebt verstuurd staan er geen nieuwe berichten meer in het overzicht.

#### Scenario 1: Nieuwe berichten beluisteren

Je komt in het berichtenscherm binnen waar de berichten op datum gesorteerd staan. Je wilt snel zien welke nieuwe berichten zijn binnengekomen. Het systeem heeft onthouden dat je de vorige keer voor het filter 'onbeluisterde berichten' had gekozen, en laat dus ook nu alleen de onbeluisterde berichten zien. De berichten zijn gesorteerd op datum, dus de nieuwe berichten staan automatisch bovenaan. Er zijn geen nieuwe berichten, maar je wilt nog wel een ouder bericht terugluisteren. Je kiest daarom voor het filter 'alle berichten'. Alle berichten worden nu weergegeven. Omdat je veel berichten hebt, zoek je met de zoekfunctie naar een specifiek bericht.

#### Scenario 2: Berichten filteren

### 3.2.2 Use cases

De use cases zijn gedetailleerde beschrijvingen van acties die de gebruiker kan uitvoeren.

#### Nieuwe berichten bekijken

1. De gebruiker opent de Greeting Voicemail
2. De gebruiker moet eventueel inloggen
3. Het systeem toont de inhoud van de mailbox
  - met onbeluisterde en beluisterde berichten?
  - of met alleen de onbeluisterde berichten?

4. De gebruiker kiest een bericht (beluisterd of onbeluisterd)
5. Optie: De gebruiker filtert de berichtenlijst
  - 5.1. De gebruiker kiest voor het filter 'Onbeluisterd' (alleen onbeluisterde berichten) / 'Alle berichten' (alleen met bericht) / 'Alle oproepen' (alle gemiste oproepen)
  - 5.2. Het systeem filtert de berichtenlijst
  - 5.3. Het systeem onthoudt het gekozen filter, om dat bij het volgende bezoek weer te gebruiken
6. Optie: De gebruiker drukt op 'bericht beluisteren'
  - 6.1. De gebruiker beluistert het bericht
  - 6.2. Het systeem markeert het bericht als 'beluisterd'
  - 6.3. De gebruiker kan deze markering eventueel weer verwijderen
7. Optie: De gebruiker drukt op 'bellen' / 'sms'en' / 'e-mailen'
  - 7.1. Het systeem opent het juiste scherm, waar het telefoonnummer/het e-mailadres is ingevuld
  - 7.2. De gebruiker voert de actie uit
  - 7.3. *Het systeem markeert het bericht als 'gebeld'/'ge-sms't'/'ge-e-maild'*
  - 7.4. De gebruiker kan deze markering eventueel weer verwijderen
8. Optie: De gebruiker wil het bericht verwijderen/archiveren
9. Optie: De gebruiker drukt op 'berichtdetails bekijken'
  - 9.1. Het systeem toont de details van het bericht

Use case 1: Nieuwe berichten bekijken

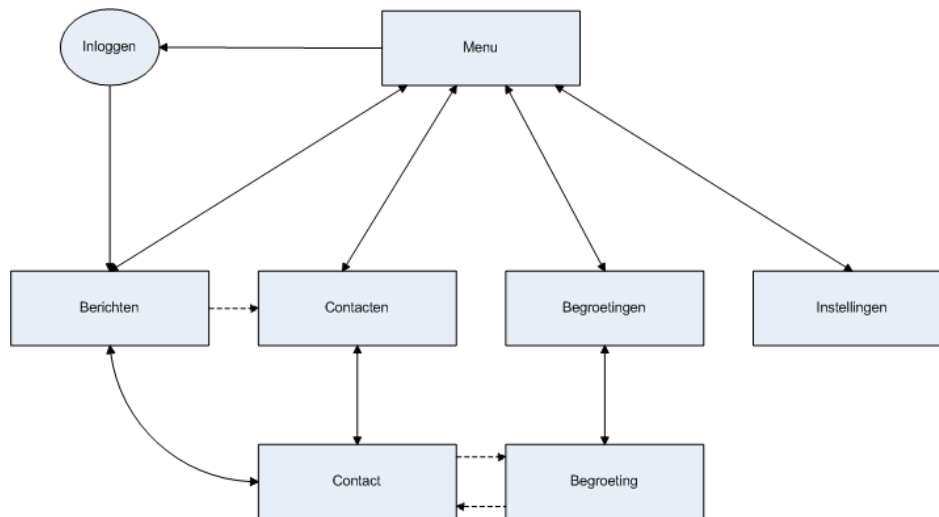
### 3.3 Vormgeving

Aan de hand van de eiseninventarisatie, de scenario's en de use cases zijn we begonnen met het maken van ontwerpschetsen op papier. De eerste schetsen zijn opgenomen in [6]. Na een bespreking met de opdrachtgever zijn er enkele aanpassingen gemaakt die beschreven staan in [7]. In deze paragraaf zullen we de combinatie van beide documenten beschrijven. We laten zien hoe de schetsen zijn gemaakt en verder zijn uitgewerkt. Ook hier zijn net als bij de scenario's en de use cases alleen enkele voorbeelden opgenomen in het verslag. Voor alle schetsen verwijzen wij naar [6] en [7].

#### 3.3.1 Hiërarchie van de applicatie

De applicatie is opgebouwd met een hiërarchische structuur, zoals dat door Apple wordt aanbevolen. De structuur is weergegeven in figuur 3.3. Hier zijn de schermen te zien waarbij de doorlopende pijlen aangeven welke sprongen er mogelijk zijn. De gestippelde pijlen geven aan dat een scherm wel informatie van het andere scherm laat zien, maar dat je vanuit dat scherm niet in het andere scherm kunt terechtkomen.

Door de hiërarchische structuur bevat de applicatie weinig lussen. De meeste paden lopen via het menu. Zo weet de gebruiker altijd waar hij is. Een nadeel van deze indeling is dat er wel veel geklikt moet worden: van een contact naar de instellingen kost bijvoorbeeld al drie stappen.



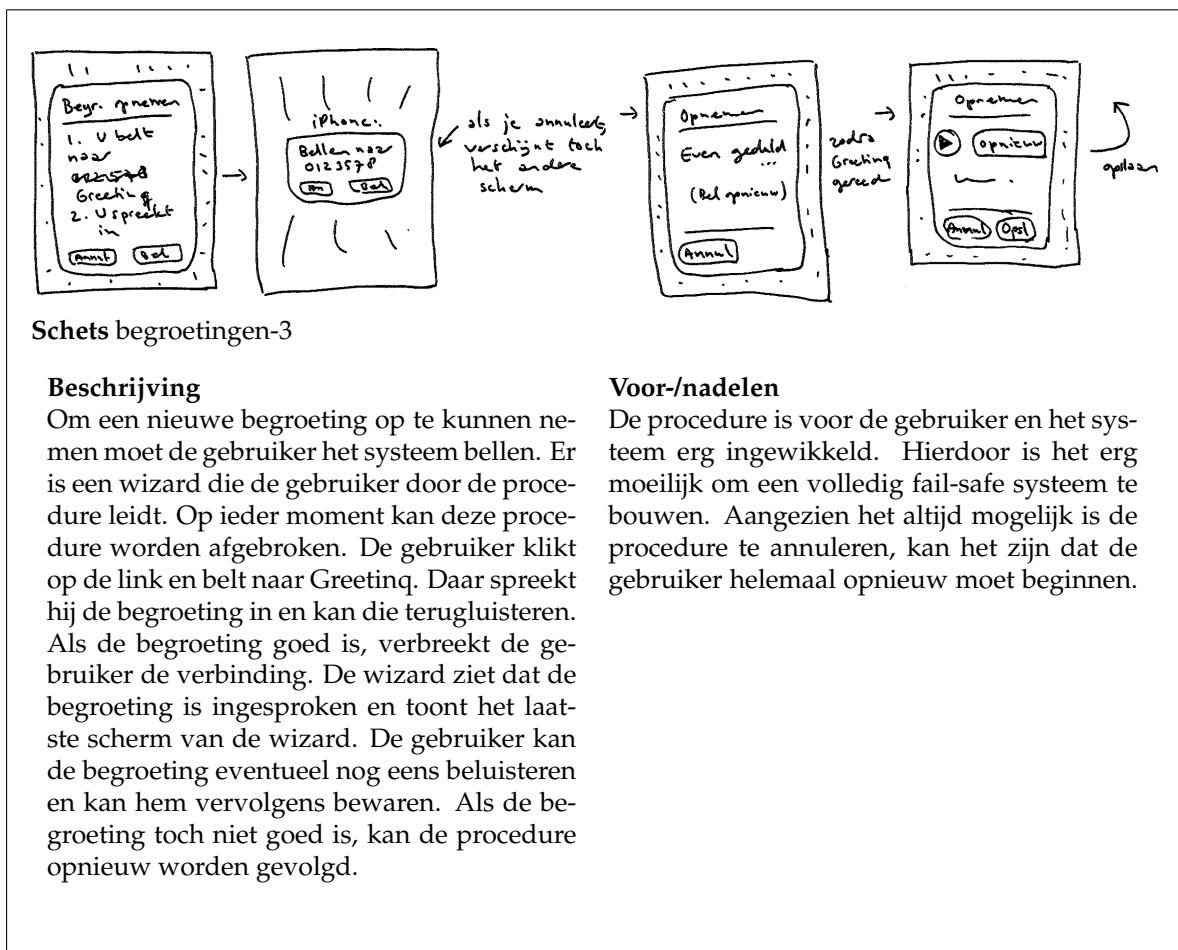
Figuur 3.3: De navigatiestructuur van de applicatie.

### 3.3.2 Schetsen

In het laatste deel van de eerste ontwerpfase hebben we het daadwerkelijke ontwerp van de interface gemaakt. Dit hebben we gedaan door van alle schermen schetsen te maken. De schetsen met hun voor- en nadelen zijn beschreven in de documenten ontwerpideeën fase 1 en ontwerpideeën fase 2 [6, 7].

Bij het ontwerpen zijn we iedere keer begonnen met het belangrijkste onderdeel op het scherm. Daaromheen kwamen vervolgens de overige functionaliteiten. Van belangrijke schermen hebben we in een aantal schetsen de verschillende opties uitgeprobeerd.

Het opnemen van een begroeting is een ingewikkeld onderdeel van de applicatie. Omdat het niet mogelijk is om direct via de webapplicatie geluid op te nemen, moet de gebruiker naar Greeting bellen om een nieuwe begroeting in te spreken. Om de kans op fouten te verkleinen is gekozen voor een pop-up-wizard, die het opneemproces begeleidt. In figuur 3.4 is te zien hoe de opneemprocedure met deze wizard verloopt.



Figuur 3.4: Het opnemen van berichten.



**Schets berichten-4**

**Beschrijving**  
 Bij ieder bericht worden knoppen weergegeven waarmee direct de functies voor dat bericht aangeroepen kunnen worden. De knoppen zouden ook kunnen weergegeven of een functie al gebruikt is.

**Voor-/nadelen**  
 Doordat de knoppen direct naast het bericht zijn geplaatst, is duidelijk bij welk bericht de knoppen horen. Als de knoppen onder in beeld zouden staan, is die relatie minder zichtbaar. Deze methode neemt, als de knoppen voldoende groot zijn, wel veel ruimte in beslag, waardoor mogelijk minder berichten in de lijst passen.

Figuur 3.5: Een schets van het berichtenscherf.

**Schets filter-5**

toggle-knopjes

**Beschrijving**  
 Voor de toggle-knopjes zijn verschillende varianten te verzinnen.

**Voor-/nadelen**  
 Deze uitgebreide toggle-knopjes geven duidelijker weer dat de filters de selectie alleen uitbreiden. De nieuwe berichten zijn altijd zichtbaar, de oude berichten en de gemiste oproepen kunnen er alleen extra bij worden weergegeven. Deze knopjes nemen wel relatief veel ruimte in beslag.

Figuur 3.6: De opties voor de knopjes.



In de tweede ontwerpfase zijn enkele clickable<sup>1</sup> prototypes van de interface gebouwd. In dit hoofdstuk zullen we eerst de ontwerpkeuzes per scherm behandelen, zodat duidelijk wordt wat er in ieder scherm te zien zal zijn. Vervolgens zullen we de prototypes beschrijven en uitleggen hoe ze zijn gebouwd.

## 4.1 Ontwerpkeuzes

We lopen de schermen langs in de volgorde waarin ze zijn ontworpen.

Gesture	Action
Tap	To press or select a control or item (analogous to a single mouse click).
Drag	To scroll or pan.
Flick	To scroll or pan quickly.
Swipe	In a table-view row, to reveal the Delete button.
Double tap	To zoom in and center a block of content or an image. To zoom out (if already zoomed in).
Pinch open	To zoom in.
Pinch close	To zoom out.
Touch and hold	In editable text, to display a magnified view for cursor positioning.

Figuur 4.1: Invoermogelijkheden van de iPhone (bron: [8])

### 4.1.1 Berichtenlijst en berichten

De gebruiker zal in de meeste gevallen de applicatie starten om even zijn voicemail te bekijken. Het is daarom handig als de applicatie na het inloggen direct met de berichtenlijst opent.

De belangrijkste vraag die de gebruiker stelt is dan: welke berichten zijn nieuw? We hebben daarom gekozen voor een lijst met berichten waarin je voor ieder bericht ziet of een bericht nieuw is en waarin de berichten op datum zijn gesorteerd. Bij elk bericht staan de datum van het bericht, de naam of het nummer van de beller en of er een boodschap is achtergelaten. Pas als je op het bericht klikt verschijnen de verdere mogelijkheden: het opvragen van het contact, het antwoorden op het bericht en het afluisteren van het bericht.

De lijst zelf kan naar voorkeur gefilterd worden op nieuwe berichten en/of alleen berichten met een boodschap. De applicatie onthoudt de voorkeur nadat die gewijzigd is.

---

<sup>1</sup>Een clickable is alleen de voorkant van de website. Alle links werken en de schermen verspringen op de juiste manier. Alleen de weergegeven informatie is nep omdat er geen achterkant is.

Daarnaast kan in de lijst worden gezocht op de naam van de contacten.

Een bericht verwijderen een door een 'swipe' uit te voeren op het bericht in de lijst. Er verschijnt dan een verwijderknop. (Een 'swipe' is een van de invoermogelijkheden van de iPhone, zie ook figuur 4.1 en voor meer informatie het oriëntatieverslag in bijlage D.) Deze manier van verwijderen wordt ook in andere iPhone-applicaties gebruikt. Omdat tijdens de tests bleek dat de gebruikers de 'swipe' niet gebruikten, is in een latere versie een vaste verwijderknop toegevoegd.

#### 4.1.2 Contactenlijst en contacten

De contactenlijst geeft de namen van de bekende contacten. Vanuit dit scherm is het mogelijk om nieuwe contacten toe te voegen of de details van een contact te bekijken.

In het detailscherm van het contact staan zijn naam, nummers en e-mailadressen. Vanuit dit scherm kan de gebruiker het contact bellen, e-mailen of sms'en. Ook kunnen alle berichten van dat specifieke contact worden opgezocht.

Via het contactscherm komt de gebruiker in het wijzigenscherm voor dat contact. Hier kan de naam van het contact gewijzigd worden en kan het contact worden verwijderd. Het wijzigen van nummers en e-mailadressen gaat weer in een eigen scherm. De nummers van een contact kunnen daar ook worden gekoppeld aan een begroeting.

#### 4.1.3 Begroetingenlijst en begroetingen

In de begroetingenlijst worden de begroetingen per soort weergegeven. Vanuit dit scherm kunnen nieuwe begroetingen gemaakt worden en kan er doorgeklikt worden naar de details van een begroeting.

Het detailscherm van een begroeting geeft de informatie over die begroeting en de eventueel gekoppelde landen of nummers. De begroeting kan hier ook worden beluisterd.

Een begroeting kan worden gewijzigd in het wijzigenscherm. Als een begroeting niet gekoppeld is, worden in dat scherm alle opties voor de begroeting weergegeven: het wijzigen van de naam, het opnieuw opnemen van de begroeting, het koppelen van nummers, het koppelen van landen en de knop om de begroeting de standaardbegroeting te maken. Als een begroeting van een specifiek soort is, worden de opties voor andere soorten niet weergegeven. In dat geval wordt wel uitgelegd hoe de begroeting van soort kan worden veranderd.

#### 4.1.4 Menu, inloggen en instellingen

Het menu bevat de links naar de drie lijsten, de instellingen en een knop om mee uit te loggen. Wanneer je uitlogt kom je vanzelf in het inlogscherm terecht. Bij de instellingen kan de taal van de applicatie worden ingevuld en kunnen de notificatie-instellingen worden veranderd. Ook kunnen in dit scherm het wachtwoord en de pincode worden aangepast.

## 4.2 Prototypes

In totaal zijn vier prototypes gemaakt, die steeds met Greeting zijn besproken. Het laatste prototype is gebruikt als basis voor het definitieve product. We zullen hier de vier prototypes beschrijven.



Figuur 4.2: Twee screenshots van clickable 1.0

### 4.2.1 Clickable 1.0

De eerste clickable bestaat uit een HTML-pagina voor ieder scherm in de applicatie. De pagina's zijn zo gebouwd dat met een stylesheet de lay-out van de hele applicatie kan worden beheerd. Op deze manier ziet elk scherm er hetzelfde uit. De clickable is gebruikt om onze eerste ideeën aan de opdrachtgevers te presenteren. De uitkomsten van de presentatie zijn verwerkt in clickable 1.5.

In figuur 4.2 is een impressie te zien van deze eerste clickable. We zien de berichtenlijst met twee nieuwe berichten. Het tweede bericht is uitgekapt. Het bericht kan beluisterd en beantwoord worden. Het pijltje aan de rechterkant van het bericht is de link naar de details van het contact. Aan de lijst is verder te zien welke berichten een boodschap bevatten en hoe lang die boodschappen zijn.

### 4.2.2 Clickable 1.5

Naar aanleiding van de presentatie van clickable 1.0 is het ontwerp op verschillende punten aangepast. Zo is het mogelijk geworden om een begroeting te hebben die nergens aan is gekoppeld. Ook hebben de begroetingen tussenkopjes gekregen. Tijdens de discussie werd ook besloten dat de filteroptie voor gemiste oproepen standaard aan moet staan, zodat in eerste instantie alleen de berichten met een boodschap worden weergegeven. Figuur 4.3 laat zien hoe deze opmerkingen in clickable 1.5 zijn verwerkt.



Figuur 4.3: Twee screenshots van clickable 1.5

### 4.2.3 Clickable 1.6

In clickable 1.6 hebben we het grafisch ontwerp van de applicatie verbeterd. We hebben hierbij rekening gehouden met het huisstijldocument van Greeting [9]. Alle plaatjes van deze versie zijn te zien in bijlage C. We zullen hier een paar van de plaatjes bespreken.

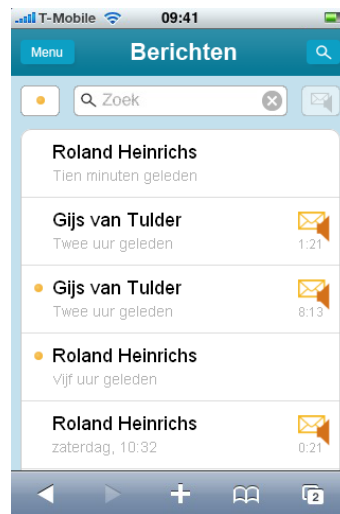
Iedere pagina heeft dezelfde kopbalk met daarin de navigatieknoppen. De balk beweegt uit het scherm als naar beneden wordt gescrold. De inhoud van iedere pagina is meestal weergegeven in witte kaders op een lichtblauwe achtergrond. Hierdoor is direct duidelijk wat de functionaliteiten zijn van iedere lijst. De knoppen die direct verbonden zijn met de inhoud staan binnen de kaders. Dit is duidelijk terug te zien in de berichtenlijst (zie figuur 4.4). De functies die met ieder bericht samenhangen staan in het uitklapmenu.

Gewone knoppen, bijvoorbeeld voor het opslaan van gegevens, hebben dezelfde blauwe kleur als de navigatieknoppen. De verwijderknoppen zijn altijd oranje. Deze kleur is gekozen om in de buurt te blijven van het rood van Apple, maar tegelijk ook de huisstijl te volgen. De verwijderknoppen zijn over het algemeen verder weggestopt op de verschillende pagina's, omdat verwijderen geen dagelijkse bezigheid is. Alleen bij de berichten zijn ze sneller toegankelijk.

### Berichten

Dit is de belangrijkste pagina. Met het ontwerpen zijn we dan ook hier begonnen. We hebben gekozen voor strakke, simpele pictogrammen die herkenbaar zijn vanuit Greeting of iPhone-applicaties. We hebben niet gekozen voor de gedetailleerde pictogrammen die Greeting in de pilotversie van de gewone webapplicatie gebruikt. Voor de herkenbaarheid zou dit wellicht beter zijn, maar de gedetailleerde pictogrammen passen niet goed bij de stijl van een iPhone-applicatie. Dit is terug te zien in figuur 4.5.

Nieuwe, dus nog onbekeken, berichten zijn gemarkeerd met een lichtoranje bolletje. Dit is afgeleid van de manier waarop andere iPhone-applicaties, zoals het e-mailprogramma, dit weergeven. Als een bericht daarnaast ook nog een boodschap bevat staat er een logo naast



Figuur 4.4: De berichtenlijst.

de naam zie figuur 4.6(a). De play-knop heeft een donkere kleur blauw gekregen en staat op een prominente plaats onder het logo dat aangeeft dat er een boodschap is. Wanneer het bericht uitklapt zal het logo direct doorverwijzen naar deze play-knop. De kleur van de play-knop komt terug in de knoppen waarmee gereageerd kan worden op het bericht. Hierdoor zijn de directe functionaliteiten rondom het bericht samen herkenbaar. Daarnaast is er de mogelijkheid om vanuit het bericht direct naar de gegevens van de contactpersoon te gaan. Dit is een ander soort functie dan het beantwoorden van een bericht. De contactpersoon heeft daarom een andere kleur gekregen en is gescheiden van de andere knoppen. Hij is gepositioneerd direct onder de naam en het nummer behorende bij het bericht.

### Begroetingen en contacten

De schermen waarin de verschillende gegevens gewijzigd kunnen worden zijn het ingewikkeldst. Omdat er geen ruimte is om met veel tekst de functionaliteiten uit te leggen, zal dit met pictogrammen moeten gebeuren. Het verwijderen van elementen uit een rij hebben we analoog gemaakt aan de methode van Apple. De icoontjes met de horizontale en verticale balken geven aan wanneer het element verwijderd kan worden. Toevoegen kan met het plusje, zie figuur 4.6(b). Het (opnieuw) opnemen van begroetingen gaat door middel van de record-knop. Door de record-knop en de play-knop naast elkaar in hetzelfde kader te zetten is het verband tussen de twee knoppen goed duidelijk.

#### 4.2.4 Clickable 2.0

De vierde en laatste clickable is gebruikt om de usability-tests uit te voeren die zijn beschreven in hoofdstuk 5. De clickable is een samenvoeging van de bevindingen in clickable 1.5 met het grafisch ontwerp van clickable 1.6. Deze clickable is bovendien gebruikt als basis voor het uiteindelijke product.

De aanpassingen die na de tests moesten worden gemaakt zijn direct verwerkt in de basis en niet meer in een nieuwe versie van de clickable. De grootste aanpassing was het toevoegen van een verwijderen-knop aan de berichten. Waar wij dachten dat een 'swipe' bekend



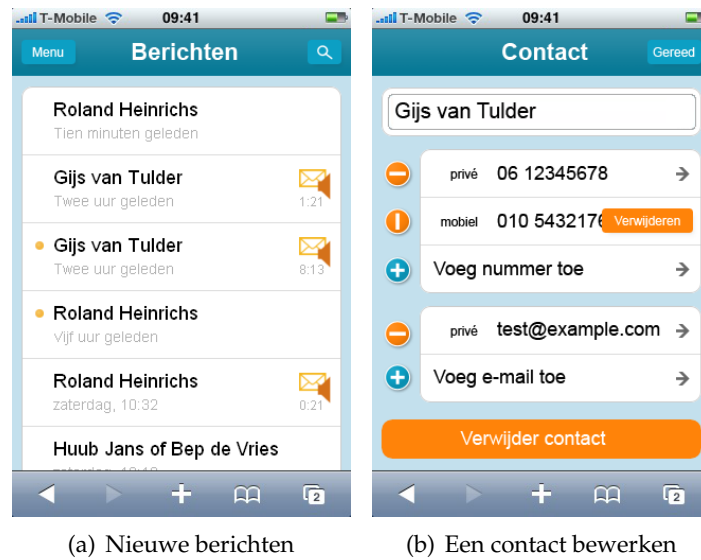
(a) Simpele pictogrammen

(b) Greeting-pictogrammen

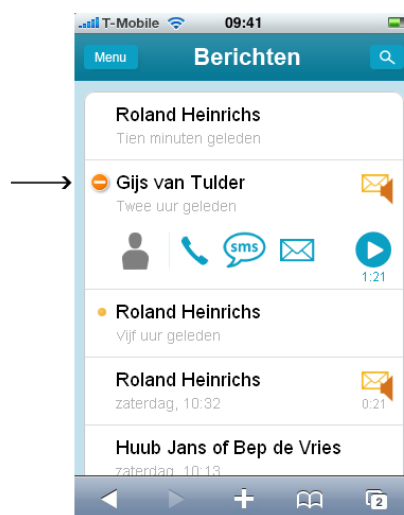
Figuur 4.5: De vergelijking tussen de verschillende pictogrammen.

of in ieder geval leerbaar zou zijn, bleek dit niet zo te zijn. Zelfs sommige gadgetfreaks waren niet met deze interactie bekend. In figuur 4.7 staat de verwijderen-knop in het bovenste bericht. De knop verschijnt in plaats van het bolletje voor nieuwe berichten, maar omdat dat bolletje verdwijnt als het bericht wordt uitgeklapt is dat geen probleem. Het knopje lijkt erg klein, maar omdat er geen andere knoppen naast zitten en ook het gebied naast het knopje aanklikbaar is, is verkeerd klikken bijna niet mogelijk.





Figuur 4.6: Twee screenshots van clickable 1.6



Figuur 4.7: De verwijderen-knop bij de berichten.



Na het ontwerp van de interface is een aantal gebruikerstests uitgevoerd. In deze tests konden ervaren iPhone-gebruikers de aanklikbare interface op een iPhone uitproberen. Met een aantal opdrachten is onderzocht hoe deze gebruikers de applicatie gebruikten en wat er aan zou kunnen worden verbeterd.

In het eerste deel van dit hoofdstuk wordt de opzet van de tests besproken. Het tweede deel beschrijft kort de profielen van de testpersonen. Het derde deel bevat de testplannen en opdrachten voor de gebruikers. Tot slot worden de resultaten van de tests geëvalueerd en samengevat.

## 5.1 Testopzet

Voorafgaand aan het maken van de tests is een testplan opgesteld volgens het DECIDE-schema van Sharp, Rogers en Preece [3, blz. 624–641]. In dit schema worden zes punten behandeld:

1. Doel;
2. Onderzoeksvragen;
3. Methode van aanpak;
4. Praktische uitvoering;
5. Ethische aspecten;
6. Evaluatie, analyse, interpretatie en presentatie van de data.

### Doel

Onderzoek naar de bruikbaarheid van de interface door bestaande iPhone-gebruikers.

### Onderzoeksvragen

- Kunnen nieuwe gebruikers snel met de interface leren werken?
- Kunnen bestaande gebruikers de applicatie efficiënt gebruiken?
- Hoe goed weten de gebruikers nog wat ze moeten doen als ze alles een keer gezien hebben?
- Vinden de gebruikers het plezierig om de interface te gebruiken?
- Vinden de gebruikers de interface mooi?
- Hoe kan de bruikbaarheid van de interface worden verbeterd?

### **Aanpak en praktische uitvoering**

Met een aantal usability-tests wordt een antwoord gezocht op de gestelde vragen. De testpersonen krijgen concrete opdrachten en worden geobserveerd terwijl ze die uitvoeren. Daarna worden enkele open en gesloten vragen gesteld.

De deelnemers aan de test worden eerst bekend gemaakt met de principes en mogelijkheden van Greeting Voicemail. Vervolgens krijgt de gebruiker de gelegenheid om zelfstandig rond te kijken in de applicatie. Op deze manier wordt getest hoe goed de verschillende onderdelen te vinden zijn (learnability). Na deze eerste verkenning wordt aan de hand van een aantal concrete opdrachten getest hoe eenvoudig de gebruiker deze met de applicatie kan uitvoeren. Hierbij kan de gebruiker de ervaring uit de verkenning gebruiken (memorability).

De iPhone wordt op de zakelijke markt nog niet heel veel gebruikt. Het is daarom lastig gebruikers uit deze doelgroep te vinden, wat het moeilijk maakt om te testen met de 'zakelijke gebruiker'. De gebruikers in de tests komen uit onze eigen kring van bekenden met een iPhone.

Tijdens de test worden de deelnemers geobserveerd. Hen wordt gevraagd om tijdens de test hardop te beschrijven hoe ze de applicatie gebruiken. De observaties worden genoteerd en na afloop uitgewerkt.

### **Ethische aspecten**

Voor de tests maken de deelnemers gebruik van de test-iPhone van Greeting. In de interface zitten geen echte telefoonnummers en de namen zijn niet de echte namen van de contacten van de deelnemers. De deelnemers worden zelf niet op beeld opgenomen of met naam genoemd in het rapport. Wel bevat het rapport een korte beschrijving van de gebruiker.

### **Evaluatie**

De evaluatie geeft, aan de hand van de aantekeningen die tijdens de tests zijn gemaakt, eerst antwoord op de vragen die hierboven gesteld zijn. Daaruit worden conclusies getrokken over het ontwerp en wordt aangegeven welke onderdelen in de applicatie naar aanleiding van de tests worden aangepast.

## **5.2 Testpersonen**

De tests zijn uitgevoerd met drie testpersonen. In een ideale situatie zou de applicatie zijn getest met zakelijke iPhone-gebruikers. Deze zijn echter lastig te vinden. De tests zijn daarom uitgevoerd met andere testpersonen, overigens wel ervaren iPhone-gebruikers.

### **5.2.1 Eerste testpersoon**

Student in de masterfase. Hij gebruikt de iPhone sinds januari, dus zo'n vier à vijf maanden. De belangrijkste functies die hij gebruikt zijn: bellen, agenda, e-mail en de treintijden. De persoon maakt nauwelijks gebruik van zijn gewone voicemail, omdat "berichten er toch altijd op neerkomen dat je terug moet bellen".

### 5.2.2 Tweede testpersoon

Student die bijna klaar is met afstuderen. Hij is al een jaar in het bezit van een iPhone. De testpersoon belt niet zo veel. Wel wordt er veel gebruik gemaakt van e-mail, spelletjes, Twitter en andere websites. De testpersoon noemt zichzelf “wel een gadgetfreak”. De testpersoon maakt geen gebruik van visual-voicemail van Apple.

### 5.2.3 Derde testpersoon

Eerstejaars student. Hij is sinds september in bezit van een iPhone, die hij voornamelijk gebruikt voor vermaak. Soms schrijft en leest hij e-mail via de iPhone. De testpersoon noemt zichzelf een gadgetfreak. Tijdens de test komt hij erg onrustig over, hij bekijkt de schermen tijdens de eerste test niet goed en mist daardoor veel details. De twee tests die daarop volgen worden iets minder goed uitgevoerd.

## 5.3 Uitvoering van de tests

De testpersoon krijgt globale uitleg over de test en het product Greeting. We gaan uit van een testpersoon die weet welk product hij heeft gekocht. Hij is dus bekend met de mogelijkheden van het product.

Om iedere deelnemer dezelfde informatie te geven, is vooraf vastgelegd wat zal worden verteld. Ook de opdrachten van de tests zijn van tevoren opgeschreven.

### 5.3.1 Over de test

- Bedankt voor uw komst.
- Doel van deze test: de interface testen, niet de testpersoon.
- De uitkomsten worden anoniem verwerkt.
- U mag stoppen als u er geen zin meer in heeft.
- Opzet van de tests.

### 5.3.2 Over Greeting Voicemail

- Systeem voor persoonlijke voicemail.
- U kunt per beller een aparte begroeting instellen, bijvoorbeeld voor vrienden of collega's.
- U kunt ook een ‘landenbegroeting’ maken, daarmee kunt u bijvoorbeeld een begroeting maken voor alle bellers uit Engelstalige landen.
- Met de iPhone-interface kunt u uw berichten beluisteren en uw begroetingen instellen.

### 5.3.3 Testcase 1: Verkenning

**Doel van de test**

Hoe reageert de testpersoon als hij het programma de eerste keer ziet?

---

**Uitvoering**

De testpersoon krijgt de iPhone, waarop het openingsscherm van Greeting Voicemail al geopend is. De testpersoon krijgt deze inloggegevens om in te loggen:

- telefoonnummer: 0612345678
- wachtwoord: xyz

Vervolgens wordt de testpersoon gevraagd om zelf wat rond te kijken en daarbij hardop te beschrijven wat hij ziet en denkt, voor- en nadat hij een actie uitvoert. Hierdoor kunnen we snel zien of de testpersoon anders tegen de interface aankijkt dan wij dachten dat hij zou doen.

### 5.3.4 Testcase 2: Nieuwe gebruiker

**Doel van de test**

De testpersoon moet zelf inzien waar hij heen moet navigeren. Zoiets als het maken van een nieuwe begroeting zou zonder verdere uitleg moeten kunnen. Hiermee worden het gebruikersgemak en de structuur van de interface getest.

---

**Uitvoering**

Na het rondkijken wordt de testpersoon gevraagd om de volgende opdrachten uit te voeren, en hierbij opnieuw hardop te beschrijven wat hij ziet, denkt en doet.

1. Beluister uw nieuwe berichten. Bel iemand terug en verwijder de beluisterde berichten.
2. Maak een begroeting voor uw collega's:
  - Jan, 06 12345678
  - Bep, 06 43214321
  - Bedrijfsnummer, 010 1234567
3. Voeg uw collega Piet toe aan uw contactenlijst en koppel hem meteen aan de collega-begroeting.
  - Piet de Vries

- mobiel: 06 12345678
  - werk: piet@werk.nl
4. Stel uw notificatie in, zodat u bij een nieuwe voicemail voortaan per sms een bericht krijgt.
  5. Maak een begroeting speciaal voor uw Engelstalige klanten uit Groot-Brittannië en de VS.
  6. Maak een andere standaardbegroeting.

### 5.3.5 Testcase 3: Gevorderde gebruiker

#### Doel van de test

Na de vorige test zijn eventueel de dingen uitgelegd die de testpersoon niet snapte. Bij deze test gaan we kijken of de testpersoon nog weet wat hij moet doen nadat hij de uitleg heeft gehad.

---

#### Uitvoering

De opdrachten zijn nu wat ingewikkelder dan bij de vorige test.

De testpersoon wordt gevraagd om een gemiste oproep te openen in het berichtenschermbild en er per sms op te reageren. De gemiste oproepen zijn nog niet zichtbaar, omdat het filter daarvoor niet staat ingesteld. De testpersoon zal dus moeten zien dat hij eerst het filter 'alleen berichten' uit moet zetten. De sms-functie zal het programma onderbreken. De testpersoon moet weer terug naar het scherm waar hij was.

- Bekijk de gemiste oproepen.
- Stuur een sms naar het nummer dat onlangs heeft gebeld.
- Ga weer terug naar het berichtenschermbild.

### 5.3.6 Testcase 4: Ontwerp

#### Doel van de test

Wat vindt de testpersoon van het ontwerp van de interface?

---

#### Uitvoering

In een gesprek met de testpersoon moet blijken of hij het ontwerp mooi en functioneel

vindt, en wat er nog kan worden verbeterd. Wellicht heeft de testpersoon ook nog andere opmerkingen of vragen.

- Wat vond u van het programma?
  - mooi / niet mooi
  - makkelijk / niet makkelijk
  - snel / niet snel
  - leuk / niet leuk
- Welk onderdeel sprak u het meest aan?
- Welk onderdeel beviel u het minst?
- Wat zou er kunnen worden verbeterd?

## 5.4 Evaluatie

Na afloop van de tests zijn de resultaten per testpersoon vastgelegd in een gedetailleerd testverslag. Hieronder zijn de belangrijkste resultaten samengevat.

De testpersonen konden in het algemeen goed overweg met de applicatie. Tijdens de eerste kennismaking met de applicatie waren ze in staat om de meeste elementen goed te benoemen. In de gesprekken na afloop van de tests lieten de testpersonen weten dat ze de applicatie over het algemeen makkelijk en efficiënt konden gebruiken. Ze vonden de interface bovendien plezierig werken. Het ontwerp van de interface en de overeenkomsten met andere iPhone-applicaties werden door de testpersonen gewaardeerd.

Op basis van de verslagen van de tests is geïventariseerd welke problemen de testpersonen tegenkwamen. De belangrijkste problemen en hun oplossingen worden hieronder toegelicht:

**Soorten begroetingen** De testpersonen hadden moeite met het begrijpen van de verschillende soorten begroetingen. Het feit dat niet zomaar alle begroetingen standaard kunnen zijn werd niet begrepen. Na de uitleg was het concept voor twee deelnemers duidelijk, de derde deelnemer bleef moeite houden.

Dit probleem kan worden opgelost door steeds alle opties in het wijzigenscherm van een begroeting weer te geven. Als een gebruiker dan een telefoonnummer probeert toe te voegen aan een landenbegroeting, dan krijgt hij een waarschuwing. De waarschuwing vertelt hem dat een begroeting maar van één soort kan zijn en dat alle landen ontkoppeld zullen worden als de gebruiker doorgaat met het koppelen van nummers. Dezelfde procedure geldt voor het omzetten van andere soorten begroetingen.

Een andere oplossing, de oplossing die uiteindelijk is gekozen, is het geven van een verklarende tekst in het wijzigenscherm van een begroeting. Op de plaats waar anders de knoppen zitten voor het toevoegen van nummers of het als standaard instellen van



de begroeting komt nu een verklaring. Ook wordt uitgelegd hoe een begroeting van soort veranderd kan worden, namelijk door de bestaande koppelingen te verwijderen.

**Swipe voor verwijderen** Niet alle testpersonen waren bekend met de 'swipe' van de iPhone. Ze konden daardoor de berichten niet verwijderen. Het waren wel allemaal testpersonen waarvan verwacht mag worden dat ze de mogelijkheden van de iPhone goed hebben uitgezocht. De uiteindelijke gebruikers zullen waarschijnlijk nog minder bekend zijn met de 'swipe'-mogelijkheid. In dat geval was het niet mogelijk om berichten te verwijderen.

De gekozen oplossing is om het verwijder-bordje voor het bericht te zetten als het bericht is uitgeklapt. Het komt dan op de plaats waar anders het bolletje voor een nieuw bericht staat. Het bericht is niet meer nieuw als je er op geklikt hebt, dus er is dan ruimte voor de verwijderfunctie. De 'swipe' blijft daarnaast gewoon bestaan en kan ook uitgevoerd worden op berichten die niet zijn uitgeklapt.

**Filterknoppen** De filterknoppen werden niet direct herkend. Zowel het feit dat het een knop betreft als de vraag waar precies op gefilterd wordt was onduidelijk. Deze functies zijn echter wel goed te onthouden als je ze eenmaal ontdekt hebt. Er is daarom niets aan gewijzigd.

**Structuur van de navigatie** De structuur van de applicatie was voor de meeste testpersonen niet meteen duidelijk. Na het inloggen kwamen ze meteen in het berichtenscherm. Als ze van daaruit naar het menu gaan, is dat voor de meeste gebruikers niet 'hoger' in de hiërarchie, maar juist dieper. De link naar 'Berichten' in het menu werd door de testpersonen in eerste instantie begrepen als 'iets instellen over berichten'. Doordat de applicatie niet op het hoogste niveau begint, duurt het iets langer voor de structuur duidelijk is. We hebben hier niets aan veranderd.

**Toetsenborden** Alle testpersonen merkten op dat de toetsenborden niet altijd geschikt zijn voor het invoeren van de gevraagde gegevens. Als een telefoonnummer moet worden ingevuld, zou het handig zijn als direct het nummertoesenbord verschijnt. Op dit moment is het niet mogelijk om dit met Safari Mobile in te stellen, dus hebben we hier niets aan veranderd.

**Categorielabels** Een van de testpersonen gaf aan dat het niet mogelijk is om zelf labels te maken bij de telefoonnummers en de e-mailadressen. Hij verwacht deze optie wel, omdat de iPhone de optie ook heeft. De optie is niet geïmplementeerd omdat de labels moeten aansluiten bij het achterliggende systeem van Greeting en moeten voldoen aan de vCard-standaard, waarin alleen vaste labels voorkomen.

**Gereed/Ga op toetsenbord** Het Safari-toetsenbord heeft knoppen 'Ga' en 'Gereed'. De 'Ga'-knop stuurt normaal gesproken het ingevulde formulier. In de applicatie heeft de 'Ga'-knop echter geen functie, maar moet de 'Gereed'-knop worden gebruikt. Tijdens de tests leverde dit regelmatig verwarring op. De 'Ga'-knop is bovendien groter en opvallender dan de 'Gereed'-knop. Aan de 'Ga'-knop is daarom een functie toegevoegd.

## **5.5 Conclusie**

In de gebruikerstest is de clickable interface met drie iPhone-gebruikers getest. Hoewel deze gebruikers niet direct tot de doelgroep behoorden, hebben de tests toch zinvolle resultaten opgeleverd. Een aantal van de problemen die de testpersonen tegenkwamen is in de uiteindelijke versie van de applicatie opgelost.

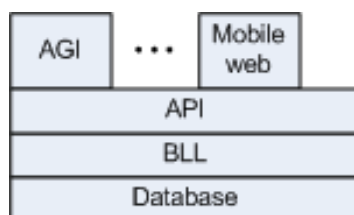
# Implementatie en softwaretests 6

Het resultaat van het ontwerp van de applicatie, de bouw van de interface en de gebruikerstests is een volledige set HTML-pagina's, afbeeldingen en JavaScript-bestanden. De volgende stap in de ontwikkeling van de applicatie is de koppeling van deze interface aan de applicatie op de webserver. In dit hoofdstuk wordt beschreven hoe dit deel van de applicatie is geïmplementeerd.

Het eerste deel van dit hoofdstuk geeft een kort overzicht van het bestaande systeem van Greeting, waarop de webapplicatie wordt aangesloten. Het tweede deel introduceert het framework Struts 2. In de laatste delen worden de implementatie en de tests van de webapplicatie verder toegelicht.

## 6.1 Greeting API en Business Logic Layer

De webapplicatie maakt gebruik van het bestaande systeem van Greeting. Dit systeem bevat de belangrijkste business logic en verzorgt het opslaan en opvragen van gegevens en audio-bestanden in de centrale database. De webapplicatie maakt voor de opslag van de gegevens volledig gebruik van deze bestaande backend.



Figuur 6.1: De applicatie 'mobileweb' maakt verbinding met de API van Greeting.

Figuur 6.1 geeft een overzicht van de belangrijkste lagen waaruit het Greeting-systeem is opgebouwd. Al het contact met de database, voor het opslaan en opvragen van gegevens, verloopt via de business logic layer (BLL). Deze laag bevat de bedrijfsregels van het systeem. De BLL controleert bijvoorbeeld of de invoer voldoet aan de regels.

De application programming interface (API) maakt gebruik van de methodes in de BLL. De API-laag is bedoeld als interface voor applicaties die het backendsysteem gebruiken. De API biedt daarvoor een aantal simpele methodes voor het opvragen en verwerken van gegevens. Er zijn bijvoorbeeld methodes waarmee in één keer een nieuwe begroeting kan worden ingesteld of waarmee op basis van een telefoonnummer en een pincode een account kan worden opgevraagd. Het is de bedoeling dat applicaties niet direct de BLL aanspreken, maar al hun opdrachten via de API versturen.

Boven de API-laag bevinden zich de applicaties die het systeem beschikbaar maken voor gebruikers. Een van de systemen is bijvoorbeeld het AGI-systeem<sup>1</sup>, dat verantwoordelijk is voor het interactieve telefoonsysteem waarmee gebruikers begroetingen kunnen instellen en berichten kunnen afluisteren.

<sup>1</sup>AGI staat voor Asterisk Gateway Interface, de interface waarmee de voicemailapplicatie met de telefooncentrale communiceert.

De webapplicatie voor mobiele telefoons is ook op dit niveau geïmplementeerd. De applicatie 'mobileweb' is de webapplicatie die de opdrachten van de telefoons opvangt, ze doorstuurt naar de API en vervolgens de resultaten terugstuurt naar de mobiele telefoon.

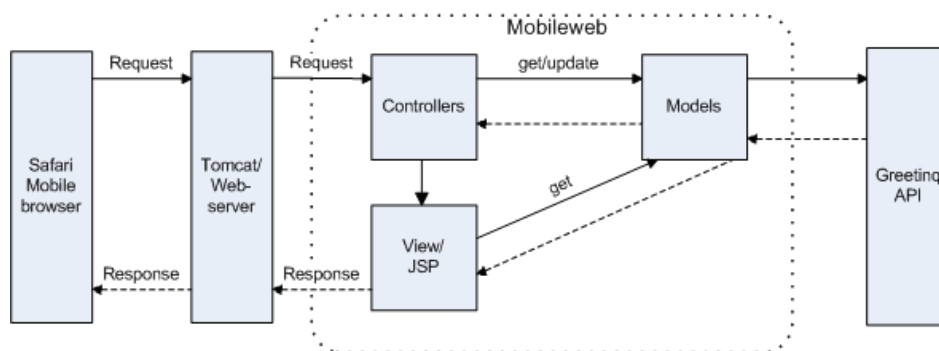
## 6.2 Overzicht van de webapplicatie

Omdat het bestaande systeem van Greeting gebruik maakt van Java Servlets, is vanzelfsprekend gekozen om de webapplicatie ook op die manier te implementeren.

Er is gekozen voor een implementatie op basis van het Struts 2-framework. Dit is een veelgebruikt framework voor het ontwikkelen van Java-webapplicaties. Het framework biedt uitgebreide ondersteuning voor zowel het ontwikkelen als het testen van dit soort applicaties.

Struts 2 is gebaseerd op het Model-View-Controller (MVC) pattern voor webapplicaties. Volgens dit patroon is de applicatie opgebouwd uit drie onderdelen, die elk verantwoordelijk zijn voor een deel van het werk. De models verzorgen de business logic van de applicatie, zoals het opslaan en opvragen van de gegevens. De views zijn verantwoordelijk voor de presentatie van de gegevens. De controllers coördineren het verloop van de applicatie: ze ontvangen de invoer van de gebruiker, geven de juiste opdrachten aan de models en laten vervolgens de views de gegevens presenteren.

Het belangrijkste voordeel van deze indeling, die in veel applicaties wordt gebruikt, is de scheiding tussen de gegevens, de presentatie en het gedrag. Doordat alle gegevens via de models worden opgevraagd en gewijzigd, hoeven alle regels alleen op die plaats te worden geïmplementeerd. Doordat de presentatie in de views gescheiden is van de gegevens, kunnen de views eenvoudig worden vervangen door een alternatieve presentatie, bijvoorbeeld voor een ander model telefoon. De rest van de applicatie hoeft in dat geval niet te worden aangepast.



Figuur 6.2: Het systeemdiagram

Het diagram in figuur 6.2 laat zien hoe de webapplicatie is opgebouwd. De gebruiker van de iPhone geeft een opdracht, bijvoorbeeld 'laat de berichten zien'. Safari Mobile, de webbrowser op de iPhone, vraagt hierop bij de webapplicatie om de pagina met berichten. Dit verzoek wordt ontvangen door de webserver (Tomcat) van Greeting. Deze bekijkt het verzoek en geeft het door aan de juiste controller in onze mobileweb-applicatie. Onze controller voert de opdracht uit: hij vraagt bij de models om de juiste gegevens en stuurt indien nodig de wijzigingen door. De models sturen deze opdrachten door aan de business logic

layer van Greeting en geven het resultaat terug aan de controller. De controller roept dan de juiste view aan, in het voorbeeld is dat de view voor het weergeven van de berichtenlijst. De view vraagt eventueel via de models de benodigde gegevens op en genereert vervolgens een HTML-pagina met het resultaat. Dit resultaat wordt via de webserver teruggestuurd naar de browser op de iPhone, die de lijst met berichten aan de gebruiker laat zien.

## 6.3 Details van de implementatie

De implementatie van de webapplicatie bestaat uit de models, de views en de controllers. (De onderdelen in het vak 'mobileweb' in figuur 6.2.) In dit deel van het hoofdstuk zullen deze onderdelen worden besproken.

### 6.3.1 Models

In de applicatie zorgen de models voor het contact met de API en BLL van Greeting. In de models-laag zijn alle entiteiten van het systeem geïmplementeerd. (Zie het klassendiagram in figuur 6.3.)

#### Extra laag

De BLL-objekten van Greeting worden niet direct in de controllers en views gebruikt. In plaats daarvan is gekozen om in de models-laag een extra laag te bouwen die de verbinding tussen de webapplicatie en het achterliggende systeem coördineert. Voor de gegevens uit de API-laag van Greeting zijn in de models-laag aparte klassen gebouwd. Deze klassen zorgen voor het contact met de API-laag: ze geven de gegevens door en verwerken de wijzigingen. De keuze voor deze extra laag heeft een aantal redenen.

Ten eerste is de extra laag noodzakelijk vanwege de specifieke eisen van de webapplicatie. Voor het maken van een nieuwe begroeting worden bijvoorbeeld meerdere HTTP-requests uitgevoerd: eerst wordt bijvoorbeeld de naam verstuurd, vervolgens te telefoonnummers. Pas in het derde request wordt de begroeting opgeslagen. Tussen het eerste en laatste request moeten de gegevens, zoals de naam, tijdelijk worden opgeslagen. De API-laag van Greeting biedt hiervoor geen mogelijkheden: de begroeting moet in één keer worden opgeslagen. Door de extra models-laag kunnen de gegevens daar tijdelijk worden opgeslagen, voordat ze in één keer naar de API-laag worden doorgestuurd.

Ten tweede geeft de extra laag het systeem extra flexibiliteit. Zo is het in de webapplicatie mogelijk om een begroeting van 'type' te laten veranderen: een telefoonnummerbegroeting kan bijvoorbeeld worden veranderd in een landenbegroeting. De API-laag van Greeting biedt hiervoor geen ondersteuning, zodat een dergelijke typewijziging betekent dat eerst het telefoonnummerprofiel moet worden verwijderd, waarna een nieuw landenprofiel kan worden aangemaakt. De logica voor dit soort vertalingen is in de models-laag geïmplementeerd, zodat de views en controllers er transparant gebruik van kunnen maken.

Ten derde kwam de extra laag goed van pas tijdens de ontwikkeling van de webapplicatie. Hoewel in het eerste deel van het project de API-laag nog niet beschikbaar was, konden de views en controllers op dat moment al wel worden ontwikkeld. Door een tijdelijke 'dummy'-implementatie van de models te maken konden de views en controllers worden ontwikkeld zonder dat een verbinding met de echte API-laag nodig was. Ook bij het testen van de controllers zijn deze 'dummy'-objekten gebruikt.

Een laatste voordeel van de extra laag is dat de controllers en views zo onafhankelijk blijven van de API-laag van Greeting. Dit heeft als voordeel dat bij een verandering in de API-interface alleen de models hoeven worden aangepast, terwijl de controllers en views ongewijzigd kunnen blijven.

## Structuur

Figuur 6.3 toont een versimpelde versie van het klassendiagram van het models-package. De structuur van een account bestaat uit drie belangrijke onderdelen: contacten, berichten en begroetingen.

Een contact is een contactpersoon van de gebruiker, met een naam, telefoonnummers en e-mailadressen. Een contact kan meerdere nummers kan hebben, bijvoorbeeld een privé-, een mobiel en een werknummer. Een telefoonnummer kan overigens bij meerdere contacten horen: denk bijvoorbeeld aan de vaste lijn van een bedrijf waar meerdere collega's werken. Omdat begroetingen worden gekoppeld aan een telefoonnummer en niet aan een contact, betekent dit dat al deze bellers dezelfde begroeting horen.

De berichten bevatten informatie over gemiste oproepen en ontvangen voicemailberichten. Bij een bericht zijn de tijd en, indien beschikbaar, het telefoonnummer van de beller opgeslagen. Als de beller een bericht heeft ingesproken is dat audiofragment aan het bericht gekoppeld.

Een begroeting is een voicemailbegroeting, die bestaat uit een naam en een geluidsfragment. Een begroeting kan worden gekoppeld aan telefoonnummers of landnummers. In dat geval krijgen dat bellers met dat telefoonnummer of uit dat land automatisch die voicemailbegroeting te horen.

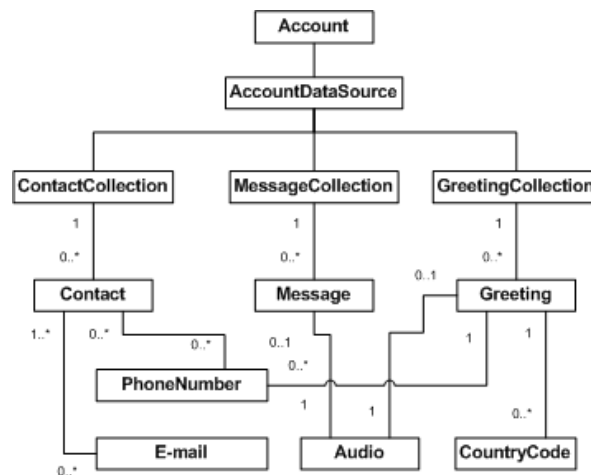
Of er telefoonnummers of landnummers kunnen worden gekoppeld hangt overigens af van het type van de begroeting. Wegens een beperking in de API van Greeting kan niet aan elke begroeting alles worden gekoppeld. Een begroeting is een telefoonnummerbegroeting, een landenbegroeting, de standaardbegroeting of een ongekoppelde begroeting. Het type van de begroeting bepaalt wat kan worden gekoppeld. Het type kan worden veranderd door eerst alle koppelingen te verwijderen: een ongekoppelde begroeting kan een telefoonnummerbegroeting of een landnummerbegroeting worden.

## Voorbeeld

De models zorgen voor de verbinding met de objecten uit de API-laag. De code in de models zorgt hierbij voor de vertaling van de API-gegevens naar de gegevens voor de webapplicatie en omgekeerd.

Een voorbeeld hiervan is te zien in codefragment 6.1. Dit fragment geeft de methode `loadListData()` uit de klasse `GreetingCollectionImpl`. Deze methode haalt de lijst met Profile-objecten uit de API-laag en zet deze om in de Greeting-objecten van de models-laag.

Deze vertaling is onder meer van belang vanwege het verschil in de opbouw van begroetingen in de API-laag en de representatie in de models-laag van de webapplicatie. De API-laag maakt onderscheid tussen het *profiel* (met daarin de verbonden telefoonnummers of landen) en een link naar de *begroeting* (het audiobestand met het welkomstbericht voor de beller). In de praktijk, en in de representatie van de webapplicatie, worden deze begroetingen en profielen samengevoegd tot een begroeting met een naam, gekoppelde telefoonnummers of landen en het audiobestand. De models-laag zorgt ervoor dat deze twee



Figuur 6.3: Versimpeld klassendiagram van het models-package. (Enkele uitgebreide diagrammen staan in bijlage B.)

representaties op de juiste manier worden vertaald.

Het codefragment 6.1 laat zien hoe de profielen worden vertaald naar begroetingen. In regel 10 wordt de lijst met profielen opgehaald. In regel 13 tot en met 17 worden de profielen omgezet en in de lijst met begroetingen geplaatst. In regel 9 is aan *com.greetingq.bll* te zien dat het Profile onderdeel is van de BLL van Greetingq. De Greeting komt uit *com.greetingq.mobileweb.models*.

```

1  /**
2   * Loads the list of records from the API layer and constructs a
3   * Greeting for each profile.
4   * @see com.greetingq.mobileweb.models.record.RecordCollectionImpl#getList()
5   */
6  @Override
7  protected List<Greeting> loadListData() {
8      List<com.greetingq.bll.items.profile.Profile> bllProfiles =
9          getAccountDataCollection().getMailboxDataCollection().getProfiles();
10     List<Greeting> greetings = new ArrayList<Greeting>();
11
12     for (com.greetingq.bll.items.profile.Profile bllProfile : bllProfiles) {
13         greetings.add(new GreetingImpl(getDataSource(),
14                                     getAccountDataCollection(),
15                                     bllProfile));
16     }
17
18     return greetings;
19 }

```

Codefragment 6.1: De methode loadListData() uit GreetingCollectionImpl

### 6.3.2 Controllers

De controllers zijn volgens het Model-View-Controller pattern verantwoordelijk voor de ontvangst en verwerking van de opdrachten van de browser. Via de models zorgen ze dat de opdrachten worden uitgevoerd. Als dat is gebeurd verzamelen ze de benodigde informatie en geven die voor de presentatie door aan de juiste view.

De Tomcat-webserver en het Struts 2-framework nemen een deel van het werk van de controllers op zich. Ze zorgen voor de ontvangst van het verzoek van de browser, halen daar de opdracht uit en zorgen dat de juiste controller wordt gestart om het antwoord te geven. Als de controller een verwijzing naar de view teruggeeft, zorgt het Struts 2-framework dat deze view wordt uitgevoerd en dat het resultaat wordt teruggestuurd naar de browser. Tomcat en Struts 2 zorgen bovendien voor het beheer van de inlogsessie, zodat de gegevens tussen de requests bewaard blijven.

De webapplicatie bestaat uit een aantal controllers. Elk onderdeel heeft zijn eigen controller. Via de REST-plugin van Struts 2 wordt bepaald welke controller een verzoek moet behandelen. De plugin kijkt hiervoor naar de URL van het verzoek: als de URL `/contacts/32` wordt opgevraagd zal bijvoorbeeld de `show()`-methode van de `ContactsController` worden aangeroepen om het contact met id 32 weer te geven, terwijl de URL `/contacts` leidt naar de `index()`-methode van dezelfde controller, die in dat geval de lijst met contacten laat zien.

#### Voorbeeld

Codefragment 6.2 geeft de functie `index()` van de `ContactsController`, die ervoor zorgt dat de lijst met contacten wordt weergegeven.

Als de `index` wordt aangeroepen moeten in principe alle contacten worden weergegeven. Er bestaat een mogelijkheid om de lijst te filteren op een telefoonnummer. Dit zien we terug in regel 12 en 13. Het telefoonnummer wordt in dat geval doorgegeven door de browser en kan via het Struts 2-framework worden opgevraagd. Als er niet gefilterd hoeft te worden, zal de methode aan de models-laag de lijst met contacten vragen. Als er wel gefilterd wordt zal de controller vragen om de gefilterde lijst. (Het eigenlijke opvragen en filteren gebeurt dus in de models-laag.)

In regel 20 tot en met 25 zien we de mogelijkheden voor de contactenlijst. Als de gewone lijst met contacten wordt opgevraagd zullen alle contacten worden weergegeven, via de gewone `index-view`. Het is echter ook mogelijk om een popup op te vragen met alle contacten die bij een bepaald nummer horen. In dat geval wordt de speciale view `index.popup` aangeroepen.

```

1  /**
2   * Lists all contacts.
3   *
4   * Action for GET /contacts
5   * Parameters:
6   *   number = if set, only show contacts with this phone number
7   *
8   * @return the response
9   */
10 public HttpHeaders index() {
11     // if a number is set, filter contacts by phone number
12     String numberStr = getParameter("number");

```



```

13     if(numberStr==null) {
14         list = getData().getContacts();
15     } else {
16         number = new PhoneNumberImpl(numberStr);
17         list = getData().getContacts().filterByPhoneNumber(number);
18     }
19
20     // return a list formatted for popups or a normal page
21     if (isPopupRequest()) {
22         return new DefaultHttpHeaders("index.popup");
23     } else {
24         return new DefaultHttpHeaders("index");
25     }
26 }

```

Codefragment 6.2: De index uit de ContactsController.java

### 6.3.3 Views

De presentatie van de resultaten wordt uitgevoerd door de views van de webapplicatie. Een view wordt gestart door de controller en krijgt dan de gegevens mee die moeten worden weergegeven, zoals het contact of de lijst met berichten. Via de models-laag kan de view eventueel nog extra gegevens opvragen. De view verwerkt die gegevens in een HTML-pagina, die als het resultaat van het verzoek wordt teruggestuurd naar de webbrowser.

De views worden in het Struts-framework geïmplementeerd als JavaServer Pages (JSP). Dit zijn sjablonen, HTML-pagina's, waarin op de juiste plaats gegevens worden ingevuld. Voor de implementatie in deze webapplicatie wordt daarnaast de Tiles-plugin gebruikt. Met deze plugin kunnen de views modulair worden opgebouwd, zodat bepaalde componenten eenvoudig kunnen worden hergebruikt. De titelbalk is op de meeste pagina's bijvoorbeeld hetzelfde en kan via Tiles in alle JSP-sjablonen worden overgenomen.

De views zijn gebaseerd op de HTML-pagina's van het clickable prototype. Die HTML-pagina's zijn omgezet in JSP-sjablonen, waarbij op de juiste plaats de gegevens uit de models worden ingevuld.

#### Voorbeeld

Codefragment 6.3 toont een deel van het sjabloon voor het weergeven van een contact. Deze view wordt aangeroepen door de controller voor de contacten, die het contact dat moet worden weergegeven aan de view meegeeft. Met Struts-tags, zoals `<s:property value="name"/>`, worden op basis van de gegevens de gegevens van het contact ingevuld, zoals de naam of de telefoonnummers.

```

1 <h2><s:property value="name" /></h2>
2
3 <s:if test="!numbers.isEmpty()">
4     <div class="box list">
5         <s:iterator value="numbers">
6             <div class="list-item list-item-linked list-item-label-value">

```

```

7         <a href="tel:<s:property value="number" />">
8           <span class="label"><s:property value="label" /></span>
9           <span class="value"><s:property value="number" /></span>
10          </a>
11        </div>
12      </s:iterator>
13    </div>
14  </s:if>

```

Codefragment 6.3: Een deel van contacts/show.jsp

## Vertalingen

Om de applicatie geschikt te maken voor internationaal gebruik, is het nodig om de Nederlandse teksten makkelijk te kunnen vervangen door bijvoorbeeld Engelse teksten. Afhankelijk van de taalinstelling van de gebruiker wordt dan de juiste taal gebruikt.

Het Struts 2-framework biedt ondersteuning voor het vertalen van de teksten in de JSP-templates. Door de teksten te vervangen door een vertaal-tag, wordt op die plaats in het sjabloon de tekst in de juiste taal ingevoegd. De tekstfragmenten uit alle sjablonen worden in een centraal bestand opgesomd. Door van dit bestand een vertaling te maken, en die vertaling vervolgens door de controller te laten activeren, worden op de plaats van de vertaal-tags de vertaalde teksten ingevoegd.

Om het maken van de vertalingen te vergemakkelijken, zijn in de JSP-sjablonen de Nederlandse teksten voorzien van vertaal-tags. Een speciaal script doorzoekt de map met JSP-sjablonen en verzamelt alle vertaal-tags en maakt daarvan een taalbestand. De Nederlandse tekst uit de sjablonen is daarin al ingevuld. Door de teksten te vertalen en het taalbestand onder de juiste naam op te slaan, is bijvoorbeeld een Engelse vertaling te maken.

Een voorbeeld van de vertaalmogelijkheid staat in codefragment 6.4. In de <h1> op regel 2 staat de vertaal-tag <s:text name="messages.index.title">Berichten</s:text>. De naam messages.index.title geeft de naam van deze tekst. In het Nederlandse taalbestand wordt op die plaats 'Berichten' ingevuld, in de Engelse variant wordt dat 'Messages', omdat dat in het Engelse taalbestand is ingevuld.

Het talenpakket is eenvoudig uit te breiden door het Engelse of het Nederlandse bestand te vertalen. De tags worden vervolgens automatisch ingelezen.

```

1 <!-- screen header -->
2 <div class="header">
3   <h1><s:text name="messages.index.title">Berichten</s:text></h1>
4   <!-- navigation -->
5   <a href="<s:url value="/menu"/>" class="button button-left" slidedir="ltr"><
6     s:text name="common.menu">Menu</s:text></a>
7 </div>

```

Codefragment 6.4: De header uit messages/index.jsp

### Ondersteuning van andere telefoons

Een van de eisen aan de webapplicatie is de ondersteuning voor andere telefoons. De applicatie is in eerste instantie ontwikkeld voor de iPhone, maar omdat de applicatie via de webbrowser werkt is het relatief eenvoudig om de applicatie ook voor andere telefoons beschikbaar te maken. In de applicatie is hiervoor een speciale voorziening opgenomen.

De controllers en models hoeven voor een ander model telefoon niet te worden aangepast. De models worden alleen door de controllers en views gebruikt. De controllers zijn zo veel mogelijk gebouwd om standaard HTTP-formulieren te verwerken. Er zullen daarom geen of slechts heel weinig wijzigingen nodig zijn om de controllers met andere views te laten werken.

De views, de HTML-sjablonen, de stylesheets en de JavaScript-code, moeten wel worden aangepast. Hoewel zo veel mogelijk de normale webstandaarden zijn gevolgd en het grootste deel van de applicatie ook in andere browsers werkt, bevatten de views enkele elementen die alleen met een iPhone goed werken. Bovendien is de interface afgestemd op de iPhone, zodat de gebruikerservaring met een andere telefoon niet ideaal zal zijn. Een interface die speciaal is ontworpen voor dat type telefoon heeft dan ook de voorkeur.

Als de nieuwe interface eenmaal is gebouwd, zijn de iPhone-views eenvoudig te vervangen door de views voor het andere model telefoon. Een speciale functie in de controllers geeft de mogelijkheid om per request, dus per gebruiker, de juiste view te kiezen. Ter illustratie van deze mogelijkheid is in de huidige applicatie een extra set views ingebouwd. Op basis van de browser worden de juiste views geselecteerd: gebruikers met een iPhone zien de iPhone-versie, andere gebruikers zien een uitgekleden versie van de interface.

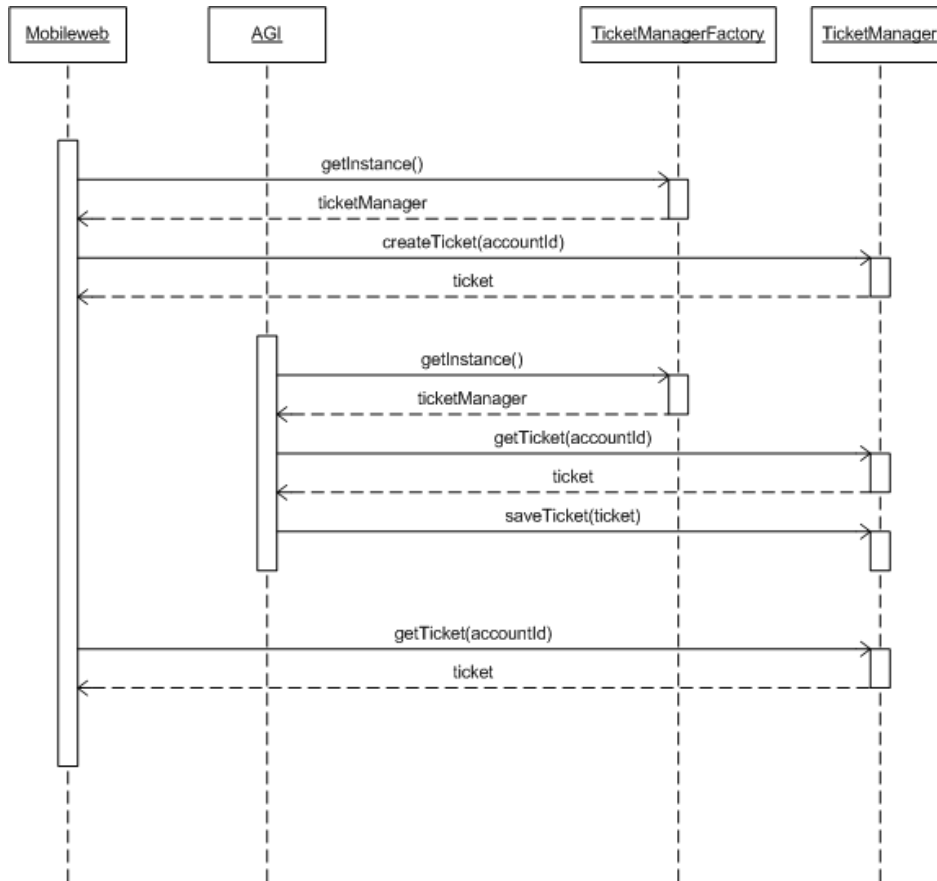
## 6.4 Nieuwe begroetingen opnemen

Het opnemen van begroetingen is met de iPhone een probleem. De iPhone biedt op dit moment geen mogelijkheid om via de browser geluid op te nemen. Voor het opnemen van de begroetingen is daarom een andere oplossing gekozen. Als de gebruiker een begroeting wil inspreken, moet hij opbellen naar het telefoonsysteem van Greeting om daar de begroeting op te nemen. De nieuwe begroeting kan vervolgens wel via de webapplicatie worden beheerd. Doordat voor deze opnamefunctie een apart telefoonnummer wordt gebruikt, kan de gebruiker de begroeting snel inspreken, zonder eerst het normale telefoonmenu te doorlopen.

Voor het opnemen van de begroetingen is een apart telefoonsysteem ontwikkeld, dat naast het normale telefoonmenu van Greeting functioneert. De webapplicatie communiceert met het telefoonsysteem via zogeheten tickets. Door het aanmaken van een ticket bereidt de webapplicatie het telefoonsysteem voor op het opnemen. Als de nieuwe begroeting is opgeslagen geeft het telefoonsysteem de begroeting via het ticket terug aan de webapplicatie.

Figuur 6.4 geeft een overzicht van de ticketprocedure. Eerst maakt de webinterface via de TicketManager een ticket aan. Hierin staan het accountId en de tijd waarop het ticket gemaakt was. Als de AGI een telefoongesprek binnenkrijgt herkent hij de gebruiker en vraagt hij het bijbehorende ticket op bij de TicketManager. Nadat de begroeting is opgenomen wordt de begroeting opgeslagen op de server. Het greetingId van de begroeting wordt in het ticket gezet. De gebruiker heeft nu de verbinding met het telefoonsysteem verbroken. De webinterface vraagt tijdens dit proces regelmatig het ticket op en kijkt of het greetingId al

is toegevoegd. Als dit is gebeurd, kan de webinterface de begroeting opvragen. Vervolgens kan de gebruiker de begroeting via de webinterface terugluisteren en bewaren.



Figuur 6.4: Sequentiediagram voor het opnemen van een begroeting.

## 6.5 Tests

De applicatie is op verschillende manieren getest. Er zijn automatische tests ontwikkeld voor de individuele onderdelen, zoals de models en de controllers. Met behulp van een coverage-programma zijn ontbrekende tests opgespoord en ingevuld. Ten slotte integratietests ontwikkeld die de hele applicatie, inclusief de integratie met de API van Greeting, kunnen testen.

Voor elke controller zijn tijdens de ontwikkeling unit-tests gebouwd. De controllers zijn in deze tests aangesloten op de zogeheten 'dummy'-models: models met dezelfde interface als de uiteindelijke implementaties, maar zonder verbinding met de achterliggende API. In de controllertests wordt geverifieerd of de controllers de juiste opdrachten uitvoeren en of de juiste informatie wordt teruggegeven.

Tijdens de ontwikkeling van de models, de verbinding met de API van Greeting, zijn voor alle functies unit-tests gebouwd. In deze tests wordt de verbinding met de API gesi-

muleerd met behulp van zogeheten mock-objecten. Met deze objecten kan in de tests precies worden gecontroleerd of de models de juiste informatie doorsturen naar de API.

Naast de unit-tests voor controllers en models is een reeks integratietests ontwikkeld met behulp van het HttpUnit-framework. Deze tests simuleren een webbrowser die de webapplicatie gebruikt. Door het invullen en versturen van gegevens en het klikken op knoppen worden in deze tests alle mogelijkheden van de applicatie uitgeprobeerd. Hierbij draait de applicatie in de echte webserver en gebruikt een echte verbinding met de API van Greeting. Op deze manier is getest of de verschillende onderdelen van het systeem op de juiste manier samenwerken.

### 6.5.1 Voorbeeld

Codefragment 6.5 laat een voorbeeld zien van één van de integratietests. In de test wordt een telefoonnummer toegevoegd aan een telefoonnummerbegroeting en wordt gecontroleerd of het telefoonnummer inderdaad is toegevoegd.

```
1  /**
2   * Test: add a phone number to an existing profile and click Done.
3   * @throws Exception
4   */
5  @Test
6  public void testAddPhoneNumberClickDone() throws Exception {
7      testSaveNewPhoneNumberGreeting();
8
9      // go to greeting
10     goToGreeting("qweqweqwe");
11
12     // edit the greeting
13     doClick(response.getFirstMatchingLink(WebLink.MATCH_CONTAINED_TEXT, "Wijzig"
14         ));
15     assertResponse(ResponseCode.OK);
16
17     addPhoneNumber("0687651234");
18
19     // click Done
20     WebForm form = response.getFirstMatchingForm(WEB_FORM_MATCH_SUBMIT_VALUE, "
21         Gereed");
22     doSubmit(form);
23
24     // should redirect to greeting
25     assertRedirectedTo("/greetings/[0-9]+");
26     doFollowRedirect();
27
28     // does it show the phone number?
29     assertTextContains("0687651234");
30 }
```

Codefragment 6.5: Een van de HttpUnit-tests

## **6.6 Conclusie**

De iPhone-webapplicatie voor Greeting Voicemail is geïmplementeerd met behulp van het Struts 2-framework. De applicatie is ingedeeld volgens het Model-View-Controller patroon, zodat presentatie, inhoud en gedrag zo veel mogelijk zijn gescheiden. Hierdoor is het relatief eenvoudig om de webapplicatie ook voor andere telefoons beschikbaar te maken. De applicatie is getest met unit- en integratietests. De tests maken het makkelijker om bij wijzigingen te controleren of de applicatie nog goed werkt.

# Conclusie en aanbevelingen 7

---

Tijdens dit project is een visuele applicatie ontwikkeld voor Greeting Voicemail. Via deze voicemailapplicatie kunnen iPhone-gebruikers via het beeldscherm van hun telefoon hun voicemail afluisteren en hun begroetingen beheren.

## Evaluatie probleemdefinitie

De eerste eis uit de probleemdefinitie, dat een iPhone-gebruiker het systeem zonder toelichting moet kunnen gebruiken, stond tijdens het project centraal. Om aan deze eis te voldoen is in het ontwerp zo veel mogelijk aangesloten bij de andere applicaties van de iPhone. De standaardelementen uit de iPhone zijn hergebruikt. Zo zijn de mogelijkheden direct herkenbaar voor ervaren iPhone-gebruikers, terwijl nieuwe gebruikers hun kennis van de voicemailapplicatie ook in andere iPhone-applicaties kunnen gebruiken.

Door middel van gebruikerstests is de bruikbaarheid van de applicatie onderzocht. Uit de tests bleek dat het grootste deel van de applicatie voldoet aan de eisen. Daarnaast leverden de tests suggesties voor verbetering. Zo was er onder andere onduidelijkheid over het onderscheid tussen de verschillende typen begroetingen die Greeting ondersteunt. Ook het verwijderen van berichten bleek een probleem, omdat zelfs de 'iPhone-freaks' in de test de 'swipe' niet kenden. Op deze punten is de interface verbeterd.

De overige drie eisen uit de probleemdefinitie zijn van technische aard. Ook aan deze eisen is voldaan. De interface is geïmplementeerd en gekoppeld aan de bestaande backend van Greeting. Hierdoor zijn alle functionaliteiten van Greeting Voicemail bereikbaar via de mobiele interface. De implementatie volgens het Model-View-Controller pattern zorgt ervoor dat de webapplicatie in de toekomst relatief eenvoudig gebruikt kan worden voor andere telefoons of de gewone website. Met behulp van automatische tests is de applicatie getest.

## Evaluatie proces

We zijn begonnen met ontwerpen voordat we volledige kennis hadden van de mogelijkheden van Greeting Voicemail. Deze volgorde heeft ervoor gezorgd dat wij onze ideeën vrij in het ontwerp konden verwerken zonder te hoeven kijken naar het al dan niet mogelijk zijn van het idee. Sommige ideeën zijn later alsnog geschrapt, andere hebben geleid tot aanpassingen van Greeting Voicemail. Een duidelijk voorbeeld hiervan zijn de contacten die nog niet in Greeting Voicemail aanwezig waren, maar wel onontbeerlijk zijn voor een visueel systeem. Het nadeel is dat onze terminologie soms afwijkt van die van Greeting. Tijdens de integratie leidde dit op enkele punten tot verwarring.

## Afsluiting

Gebruikmakend van de mogelijkheden van de iPhone en de ondersteuning van Apple zijn wij in dit project in staat geweest om een webapplicatie voor Greeting te ontwikkelen. Dit heeft ons geleerd dat er heel wat beperkingen zitten aan mobiele telefoons, maar dat met de juiste keuzes zoveel mogelijkheden geboden kunnen worden dat de beperkingen bijna in

het niet vallen. Dit maakt het ontwerpen van applicaties voor mobiele platforms een apart vak waar nog veel ruimte is voor onderzoek en ontwikkeling.

## 7.1 Aanbevelingen

Door het uitkomen van nieuwe software voor de iPhone en zelfs een nieuwe versie van de iPhone zijn er nog mogelijkheden voor verbetering van de applicatie. Daarnaast lijkt het ons ook goed om Greeting Voicemail op meerdere manieren mobiel bereikbaar te maken. Hieronder doen wij nog enkele aanbevelingen tot verbetering.

**Betere aansluiting bij iPhone** De keuze voor een webapplicatie zorgt voor een aantal beperkingen, bijvoorbeeld bij het afspelen en opnemen van geluid. Er bestaan programma's waarmee een webapplicatie eenvoudig kan worden omgezet in een iPhone-applicatie, waardoor die beperkingen zouden wegvallen. Door deze mogelijkheid, bijvoorbeeld met het programma PhoneGap[10], zou de gebruiksvriendelijkheid van de opneemprocedure kunnen worden verbeterd.

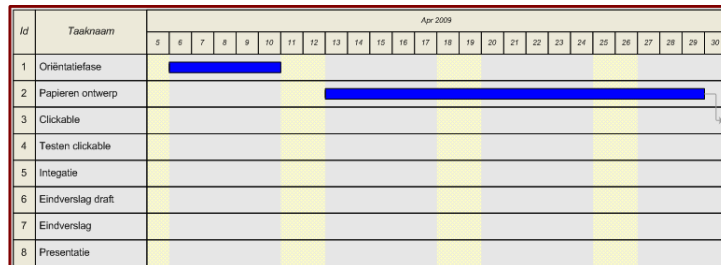
**Nieuwe mogelijkheden** De iPhone is een product dat nog steeds in ontwikkeling is. Wij hebben deze ontwikkelingen tijdens het project niet meer kunnen verwerken. Het lijkt ons goed om de ontwikkelingen bij te blijven houden en de webapplicatie mee te laten groeien met de mogelijkheden van de iPhone.

**Andere telefoons** Hoewel de iPhone een populair product is zijn er ook nog veel andere systemen. Vooral de zakelijke markt maakt meer gebruik van deze andere producten. Het lijkt ons daarom goed om ook voor andere telefoons dan de iPhone een webapplicatie te ontwikkelen. Aan Greeting is de keuze voor welke systemen ze willen ontwikkelen of dat er een universele applicatie moet komen. Wij hebben in ieder geval gezorgd dat eventuele nieuwe systemen aansluitbaar zijn.

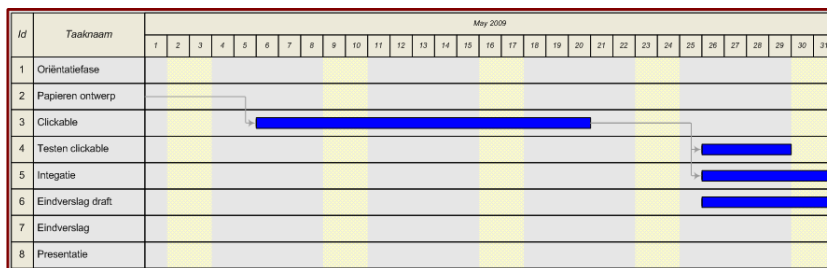
**Flexibele begroetingen** De eisen rond begroetingen zijn nu redelijk strikt. Het is voor de gebruiker niet mogelijk om nummers en landen aan dezelfde begroeting te koppelen. Tijdens de gebruikerstest bleek deze beperking voor veel gebruikers verwarrend te zijn.



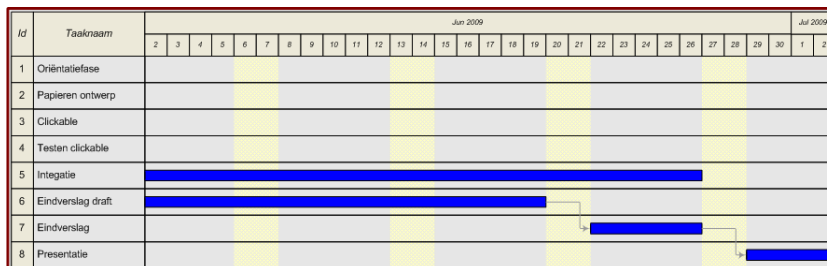
# Planning A



Figuur A.1: Planning April



Figuur A.2: Planning Mei

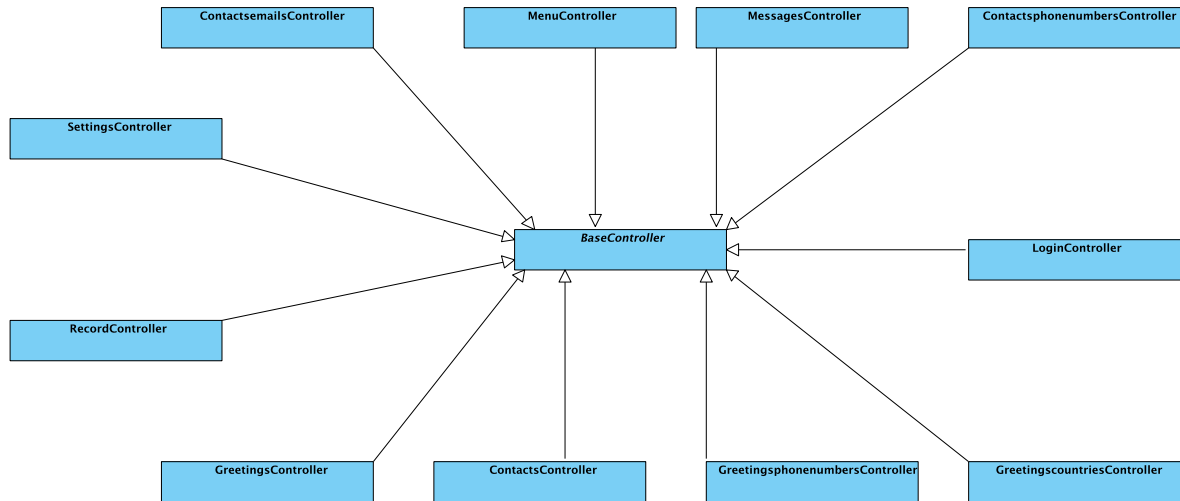


Figuur A.3: Planning Juni

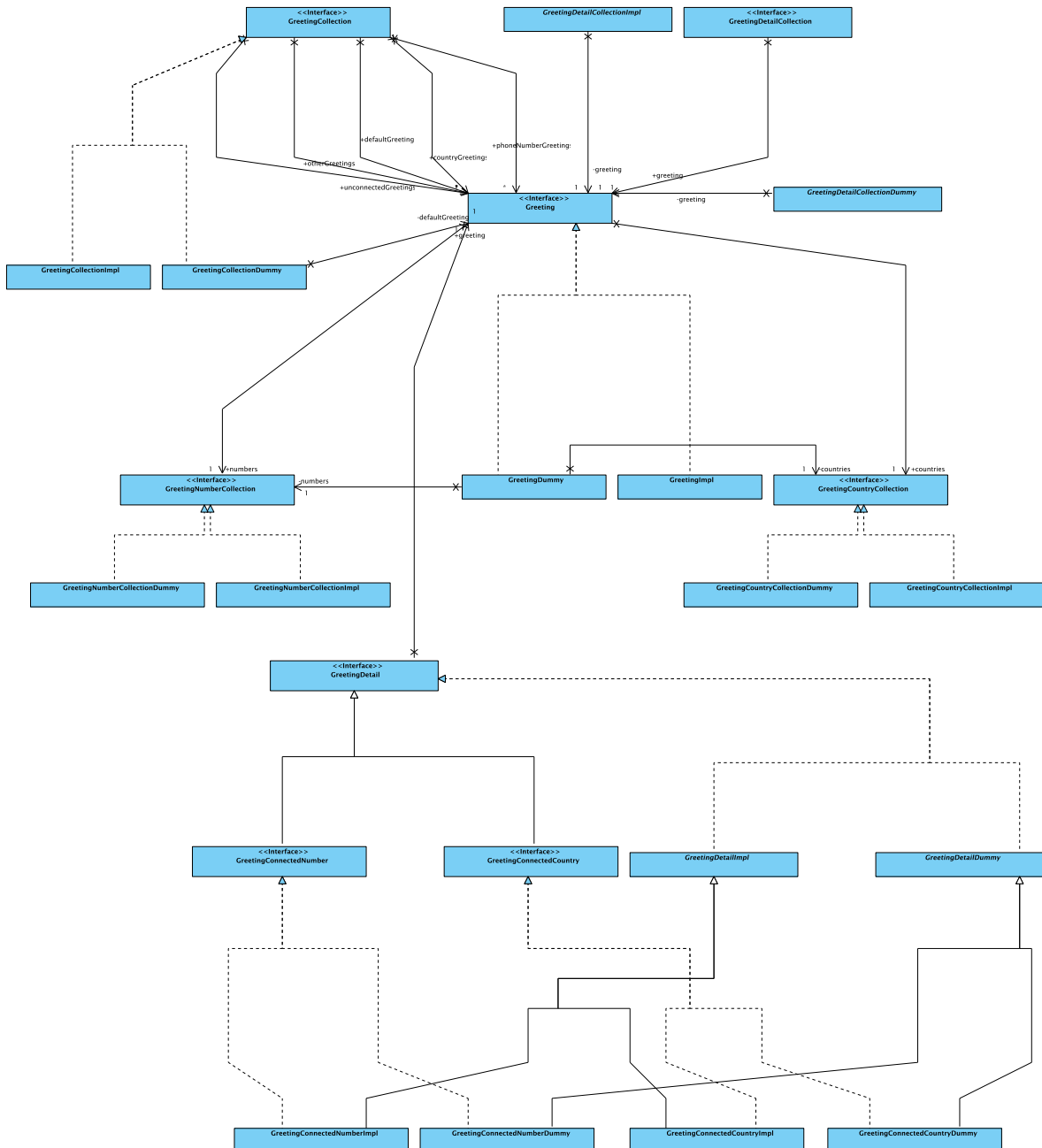


# Klassendiagrammen B

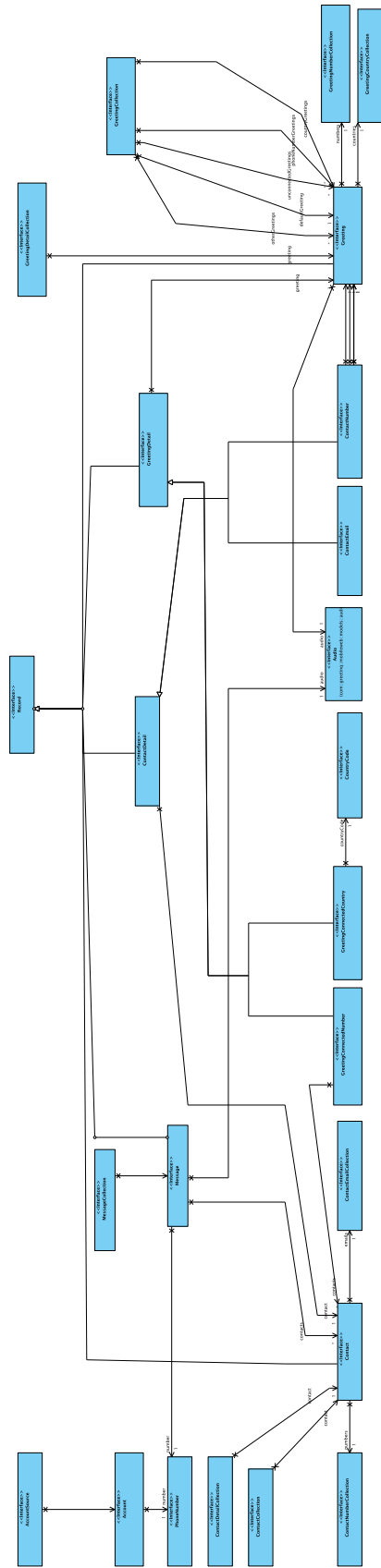
---



Figuur B.1: Klassendiagram van de controllers.



Figuur B.2: Klassendiagram van Greeting en de daar direct aan verbonden klassen.



Figuur B.3: Klassendiagram van het models-package



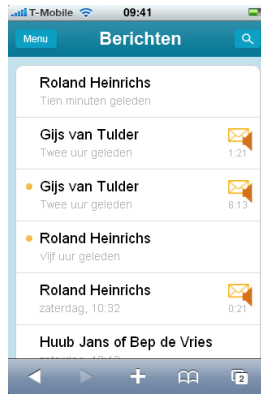
# Clickable 1.6 C



01



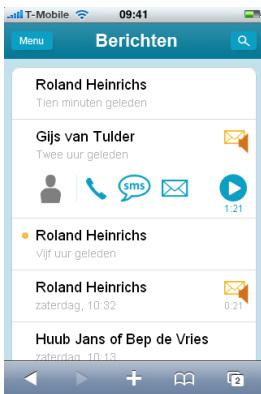
02



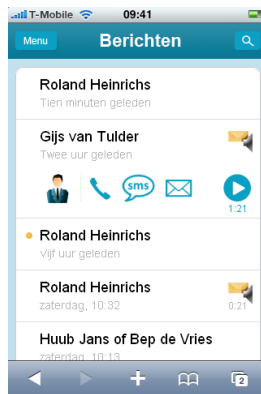
03



04



05



06



07



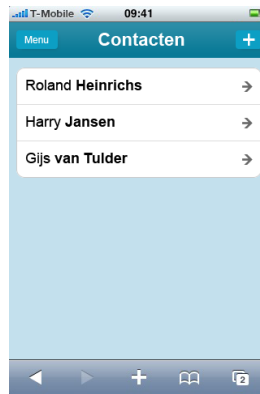
08



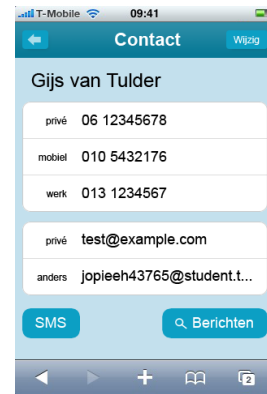
09



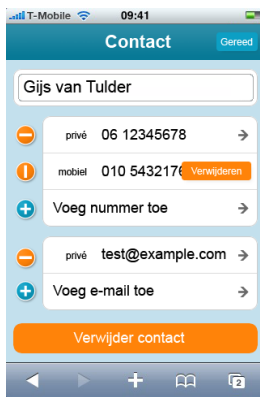
10



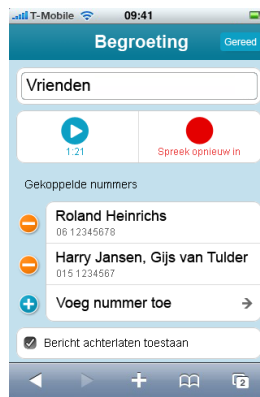
11



12



13



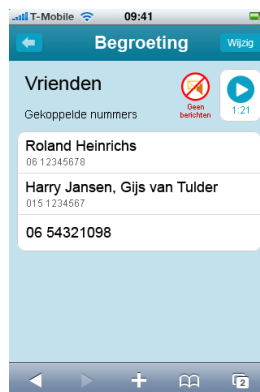
14



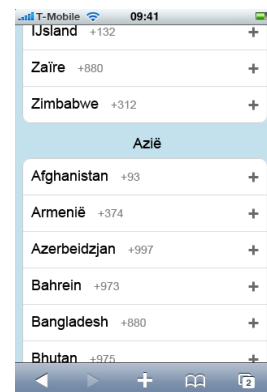
15



16



17



18



# Oriëntatieverslag D

---

## D.1 Inleiding

Dit oriëntatieverslag geeft een beknopt overzicht van de resultaten van de oriëntatiefase van ons bachelorproject bij Greeting. De opdracht en het bedrijf worden nader besproken en enkele belangrijke technische onderwerpen worden van enkele kanttekeningen voorzien.

Het doel van dit bachelorproject is de bouw van een webapplicatie voor de iPhone, waarmee het systeem voor persoonlijke voicemail dat Greeting ontwikkelt met een iPhone kan worden bediend. Via de mobiele applicatie kunnen onder andere voicemailberichten worden opgevraagd en beluisterd, maar kunnen bijvoorbeeld ook de persoonlijke begroetingen worden ingesteld. In dit verslag wordt deze opdracht nader onderzocht.

De eerste fase van het project bestaat uit het ontwerpen van de interface. De precieze uitwerking van het ontwerp aan de serverkant valt daarom buiten dit verslag. De technische aspecten van het bouwen van een webapplicatie en de vraag hoe de webapplicatie aan de backend van Greeting kan worden gekoppeld, worden pas later in het project onderzocht. In dit verslag wordt vooral aandacht besteed aan de ontwikkeling van de user-interface.

Voordat de technische aspecten worden besproken, wordt in hoofdstuk D.2 eerst de opdrachtgever, Greeting, kort geïntroduceerd. Naast de algemene structuur van het bedrijf wordt besproken wat de opdracht globaal inhoudt. Ten slotte wordt beargumenteerd waarom is gekozen voor een webapplicatie en niet voor een 'echte' iPhone-applicatie.

Het tweede deel van dit verslag bestaat uit een kort overzicht van de belangrijkste technische aspecten van het project. De webapplicatie zal gebruik maken van het bestaande voicemailstelsel van Greeting, zodat het voicemail-technische gedeelte geen onderdeel vormt van dit project. De nadruk van het project ligt op het bouwen van de interface van de applicatie.

Door de bijzondere kenmerken van de iPhone is er een aantal belangrijke verschillen tussen het ontwikkelen van een user-interface voor een gewone computer en deze telefoon. Hoofdstuk D.3 bespreekt daarom, aan de hand van de richtlijnen van Apple, welke user-interface-aspecten voor een iPhone-applicatie van belang zijn. Tijdens het ontwikkelen van de interface zullen deze richtlijnen worden gevolgd.

In hoofdstuk D.4 wordt een overzicht gegeven van de belangrijkste eigenschappen van Safari Mobile, de webbrowsen van de iPhone. Aan de hand van dit overzicht van mogelijkheden en beperkingen kan worden vastgesteld of een iPhone-applicatie als webapplicatie kan worden gebouwd en voor welke functionaliteit eventueel een alternatief moet worden gezocht.

## D.2 Greeting

### D.2.1 Het bedrijf en het product

Paragraaf 1 van dit hoofdstuk is een bewerking van een tekst van Greeting. [11]

## De applicatie

Greeting voicemail bestaat uit: meerdere begroetingen en VoicemailOnline. Hierdoor kun je een informele begroeting instellen voor vrienden, een Engelse voor buitenlandse relaties en een speciale begroeting voor je vriendin/vriend. Zo kun je iedereen de aandacht geven die hij of zij nodig heeft.

Voicemail wordt vaak geassocieerd met frustratie, voicemail wordt steeds minder vaak gebruikt, dus er zijn veel mogelijkheden om voicemail te verbeteren. 60% van alle zakelijke gesprekken komt uit bij voicemail, dus voicemail is een contactmoment dat niet verwaarloosd mag worden.

## Hoe werkt het?

Met Greeting voicemail kun je meerdere begroetingen opnemen en deze koppelen aan een telefoonnummer of landcode. Het voicemailsysteem herkent welk nummer er belt (of vanuit welk land iemand belt) en speelt de juiste begroeting af.

Berichten afluisteren of begroetingen instellen kan via het traditionele voicemail-belmenu: je kunt daar een telefoonnummer intoetsen en daarna een passende begroeting inspreken. Wil je dit liever online doen dan log je in op VoicemailOnline en kun je via een microfoontje je begroeting inspreken. Je kunt daar ook meerdere telefoonnummers aan een begroeting koppelen en makkelijk begroetingen aanpassen of wisselen.

## Wie maakt de applicatie?

Greeting BV is begin 2007 opgericht als spin-off van de TU Delft. Langzamerhand is het team gegroeid naar (op dit moment) 8 fte, met de juiste combinatie van ambitieuze starters en ervaren adviseurs. Greeting ontwikkelt innovatieve voicemailediensten en levert zowel aan telecomoperators als aan consumenten.

De betaversie van het product is online sinds maart 2008, sindsdien wordt er hard gewerkt aan de opschaling naar een carrier grade oplossing. Een pilot met honderd gebruikers resulteerde in veel positieve ervaringen.

## D.2.2 Over de opdracht

Telefoon is bij uitstek een mobiel product. Je zou de bijbehorende producten als voicemail dan ook het liefst mobiel willen gebruiken. Vanuit die gedachtegang is de opdracht geformuleerd om een applicatie te maken die specifiek gericht is op de iPhone.

Een applicatie die op de iPhone moet worden geïnstalleerd moet door Apple worden goedgekeurd. Of dat lukt is twijfelachtig, omdat Apple zelf al een product voor visual-voicemail heeft. Daarnaast zit er een patent op visual-voicemail, waarvoor Greeting nog geen licentie heeft. Greeting wil daarom geen applicatie die op de iPhone moet worden geïnstalleerd.

In plaats daarvan is het mogelijk om een website te maken die specifiek gericht is op de iPhone. Het wordt dan een webapplicatie, die Apple niet goed hoeft te keuren. Door alle mogelijkheden van de iPhone te gebruiken kunnen we de webapplicatie laten lijken op een normale applicatie. (Zie paragraaf D.4.1.)

## D.3 iPhone

Apple heeft enkele documenten opgesteld die ontwerpers helpen om producten te maken voor de iPhone. Deze documenten beschrijven de mogelijkheden en de beperkingen van de iPhone. Veel nadruk wordt gelegd op de verschillende mogelijkheden voor applicaties. Het is mogelijk om de lay-out van een website zo op te bouwen dat de website een native applicatie lijkt te zijn. Apple raadt aan om ervoor te zorgen dat de applicatie gebouwd wordt in de huisstijl.

Dit hoofdstuk is geen overzicht van alle mogelijkheden die de iPhone biedt. Wel zijn enkele voorbeelden opgenomen. Het overgrote deel van de informatie komt uit [12].

### D.3.1 Ontwerpen voor de gebruiker

Het voordeel van ontwerpen voor de iPhone is dat je al een beetje weet met wat voor soort gebruiker je te maken hebt. Hij is mobiel en wellicht is hij met meerdere dingen tegelijk bezig. Hij wil dus snel en eenvoudig informatie kunnen opvragen.

### D.3.2 Adviezen van Apple

Om het ontwerp inzichtelijk te maken, adviseert Apple om gebruik te maken van metaforen. Als de gebruiker voor het eerst een webapplicatie opstart moet direct duidelijk zijn wat hij allemaal kan doen. Bij het opstarten dient de belangrijkste informatie als eerste weergegeven te worden. Om het geheel overzichtelijk te houden is het handig om de lay-out van de standaardapplicaties van Apple aan te houden. De gebruiker weet al wat hij hiervan kan verwachten en hoe hij er mee om moet gaan.

#### Top down thinking

Een belangrijk onderdeel van het ontwerp is het zogeheten 'top down thinking'. Hiervoor moeten we eerst uitleggen hoe een iPhone vaak wordt bediend.

Er zijn verschillende gangbare manieren om een iPhone vast te houden.

- Je houdt de iPhone in je linkerhand en gebruikt je rechterhand voor de bediening. Dit kan natuurlijk ook andersom.
- Je houdt de iPhone in één hand en je gebruikt de duim van die hand voor de bediening.
- Je houdt de iPhone in twee handen en je gebruikt beide duimen voor de bediening.

In elk van bovenstaande gevallen kijk je naar het scherm terwijl tenminste een of meer van de vingers voor het onderste deel van het scherm hangen. Vooral bij de aanpak met twee duimen komt dit naar voren. Wanneer de duimen in rust zijn hangen ze boven de telefoon. Dit betekent dat je de belangrijkste informatie op de website bovenin het scherm moet plaatsen. Dit kan de gebruiker immers lezen zonder zijn vingers te hoeven verplaatsen. Navigatieknoppen moeten juist weer onderop het scherm. Daar kan de gebruiker bij zonder zijn vingers ver te hoeven verplaatsen.

## Navigeren met je vingers

Hiermee is een groot voordeel van de iPhone aangestipt. Waar bij de gangbare apparaten een pennetje is meegeleverd voor het navigeren, is de iPhone ontworpen voor gebruik van je vingers. Het is zelfs mogelijk om meerdere vingers tegelijk te gebruiken. In figuur D.1 worden de invoermogelijkheden weergegeven. Het grootste nadeel van je vingers, vooral je duimen, is dat ze veel groter zijn dan een pennetje. Er wordt dan ook geadviseerd om de minimale grootte van een knop te houden op  $44 \times 44$  pixels.

Gesture	Action
Tap	To press or select a control or link (analogous to a single mouse click). You receive the <code>onClick</code> event for this gesture.
Double tap	To zoom in and center a block of content or an image. To zoom out (if already zoomed in).
Flick	To scroll or pan quickly.
Drag	To move the viewport or pan.
Pinch open	To zoom in.
Pinch close	To zoom out.
Touch and hold	To display an information bubble, magnify content under the finger, or perform specific actions in built-in iPhone applications and features.
Two-finger scroll	To scroll up or down within a text area, an inline frame, or an element with overflow capability, depending on the direction of the movement. You can receive a <code>mousewheel</code> event for this gesture.

Figuur D.1: Invoermogelijkheden (bron: [12])

### D.3.3 Beperkingen

Aangezien de iPhone een mobiel apparaat is kent hij nogal wat beperkingen. Wij geven hier eerst een overzicht van de beperkingen om ze daarna te bespreken.

- Klein beeld:  $480 \times 320$  pixels
- Gelimiteerd werkgeheugen
- Eén applicatie tegelijk
- Eén scherm tegelijk
- Beperkte internetverbindingen

Het kleine beeld is een gevolg van de draagbaarheid van de telefoon. Dit betekent dat er veel beter nagedacht moet worden over wat er te zien is op een scherm. Apple geeft als advies om alleen de strikt noodzakelijke functies weer te geven. Alles moet ten alle tijden overzichtelijk zijn. Een gevolg van het kleine beeld is dat er maar één scherm tegelijk weergegeven kan worden. Een desktop-computer kan wel goed twee schermen weergeven

die dan naast elkaar verschijnen. De iPhone laat dit niet toe. Een scherm staat altijd op de volledig grootte van het beeld en kan niet kleiner gemaakt worden.

Aangezien het een mobiel apparaat betreft moet er rekening gehouden worden met de duur van de batterij en de ruimte die de telefoon bevat. Het werkgeheugen is daarom erg gelimiteerd. Het is ook niet mogelijk om dingen weg te schrijven op het vaste geheugen tijdens het rekenen. Een (web)applicatie dient dus niet veeleisend te zijn.

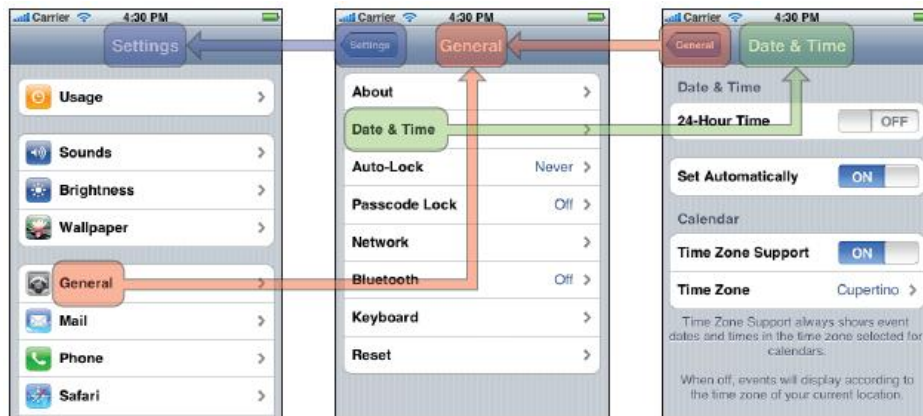
De iPhone kan ook maar één applicatie tegelijk uitvoeren. Als de telefoon gebeld wordt zal hij de applicatie afsluiten. Het is daarom belangrijk dat alle informatie altijd wordt opgeslagen. Wanneer je de applicatie dan weer opstart moet je gewoon weer in het scherm komen waar je gebleven was. Er zijn mogelijkheden geïmplementeerd om zelfs de scrollpositie in het scherm op te slaan. Als de applicatie deze mogelijkheden goed gebruikt hoeft de gebruiker niet eens te merken dat de applicatie even was afgesloten.

De iPhone maakt zijn verbinding via draadloze netwerken. Hij ontvangt wifi-signalen maar kan ook werken via het UMTS-netwerk. De snelheid en de datacapaciteit zijn beperkter dan vast internet. Ook is de verbinding nooit te garanderen (denk aan tunnels etc.). Dit kan bijvoorbeeld problemen opleveren met het afspelen van audiobestanden (paragraaf D.4.1).

### D.3.4 Ontwerpen voor de iPhone

#### Navigatiestructuur

Applicaties op de iPhone gebruiken meestal een navigatiebalk om aan te geven waar in het programma de gebruiker zich bevindt. In de navigatiebalk staan meestal een titel en een knop om terug te gaan naar de vorige pagina. Figuur D.2 laat hiervan een voorbeeld zien. De 'vorige pagina'-knop heeft de titel van de vorige pagina.

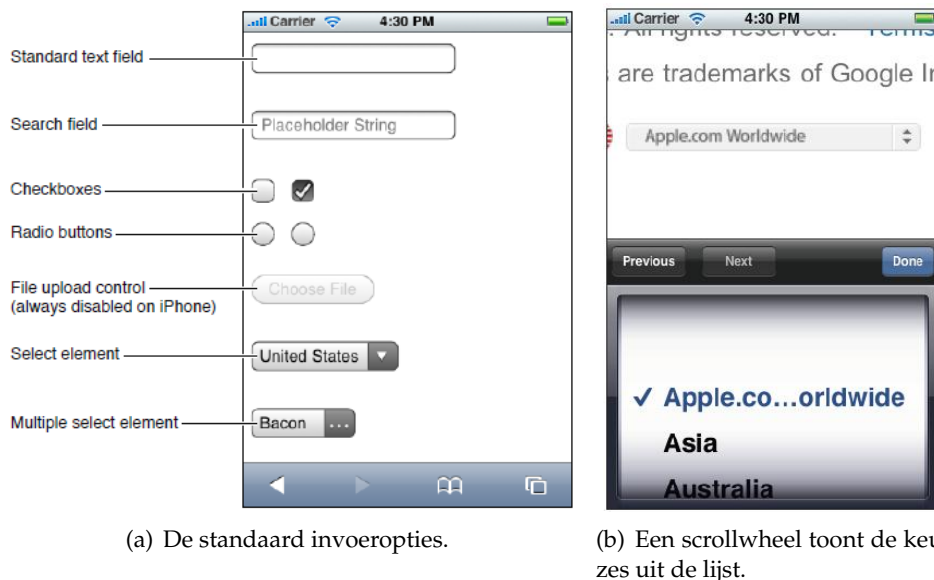


Figuur D.2: De navigatiestructuur die in veel applicaties wordt gebruikt (bron: [12])

#### Standaardelementen

Apple heeft al een aantal standaardelementen ontwikkeld, die zijn geoptimaliseerd voor gebruik met de iPhone en die in veel ingebouwde applicaties worden gebruikt (zie figuur D.3). Een voorbeeld is het scrollwheer. (Een overzicht van alle elementen is terug te vinden

in [12]). Als een website een lijst met optionele keuzes bevat, die op een normale desktop-computer zou uitklappen, dan maakt de iPhone er een soort scrollwheer van (zie voor een voorbeeld figuur D.3(b)). Dat maakt het mogelijk om makkelijk door de hele lijst te lopen en aan te klikken wat je wil hebben.



Figuur D.3: De iPhone biedt een aantal standaard invoeropties (bron: [12]).

## D.4 Safari Mobile

De webbrowser op de iPhone is Safari Mobile. Een webapplicatie voor de iPhone is afhankelijk van de technische mogelijkheden die deze browser biedt. In dit hoofdstuk worden de voor dit project belangrijkste kenmerken, mogelijkheden en beperkingen van Safari Mobile onderzocht. Hierbij wordt uitgegaan van de versie van Safari Mobile die is ingebouwd in iPhone OS 2.

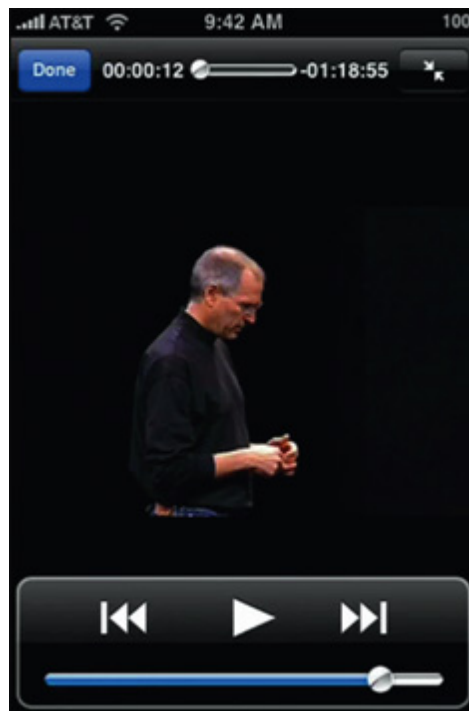
De meeste informatie in dit hoofdstuk is afkomstig uit het document van Apple over Safari Mobile. [13]

### D.4.1 Kenmerken en beperkingen

#### (X)HTML, CSS en JavaScript

Safari Mobile werkt volgens de normale standaarden voor XHTML en CSS. Een webpagina die volgens die standaarden is ontwikkeld zou het in principe op de iPhone ook goed moeten doen. De iPhone biedt voldoende mogelijkheden voor het bouwen van een moderne JavaScript-applicatie: de normale DOM-methodes en ook de voor AJAX-achtige toepassingen benodigde XMLHttpRequest-functies worden door Safari Mobile ondersteund.

Vanwege het beperkte geheugen van de iPhone worden aan een webpagina voor de iPhone iets striktere eisen gesteld dan aan een gewone webpagina. Een volledige webpa-



Figuur D.4: Video en audio worden door Safari alleen full-screen afgespeeld. (bron: [13])

gina, met plaatjes, scripts et cetera, mag voor de iPhone maximaal 10 MB groot zijn. Voor de meeste webpagina's is dit geen probleem.

### Multimedia

De iPhone en Safari Mobile bieden beperkte mogelijkheden om geluids- en videobestanden af te spelen. Het afspelen gebeurt full-screen en pas als de gebruiker het geluids- of videobestand heeft aangeklikt. De rest van de webpagina is op dat moment even niet zichtbaar. Zie figuur D.4.

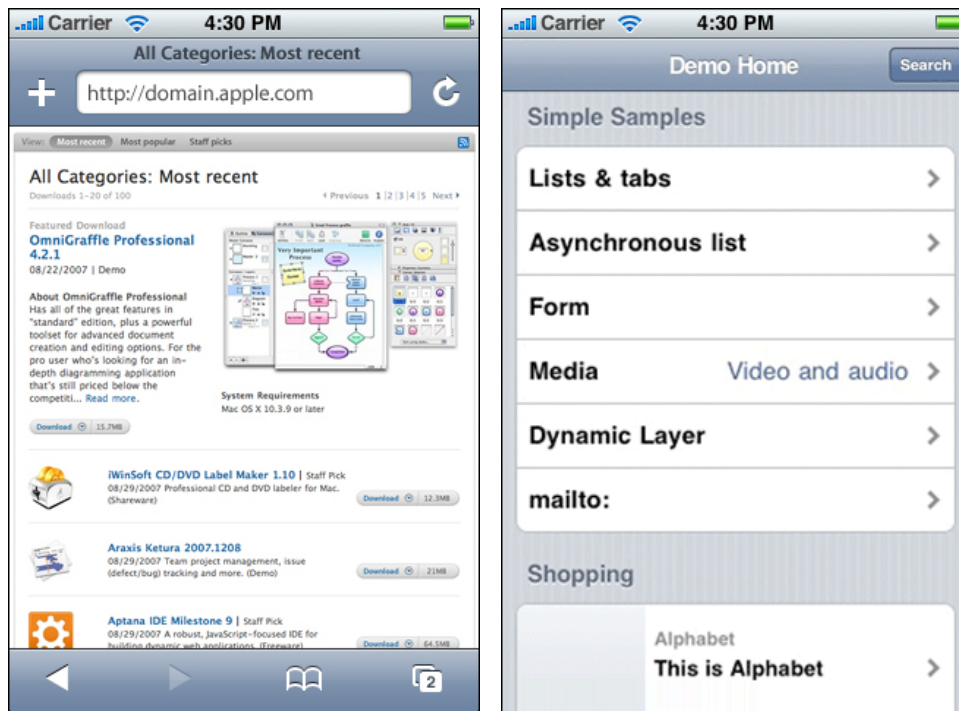
Safari Mobile kan overweg met diverse multimedia-formaten. Voor geluid zijn dit onder andere MP3, AAC en wave. Niet-gestreamde mediabestanden mogen niet groter zijn dan 10 MB. De iPhone gebruikt voor het opvragen van de audio of video zogeheten HTTP byte-range requests (ook wel content-range of partial-range). Hiermee kan op verzoek slechts een deel van het mediabestand worden gedownload. De HTTP-server die de mediabestanden aanbiedt moet dit type requests dan ook ondersteunen.

De iPhone werkt niet met RTP-streaming. Op dit moment heeft de iPhone ook geen ondersteuning voor Macromedia Flash en de bijbehorende mediaformaten.

Het is niet mogelijk om via Safari geluid op te nemen of om een eerder opgenomen geluidsbestand te uploaden.

### Webapplicaties

Safari Mobile biedt dus goede ondersteuning voor standaard-webpagina's. Deze pagina's worden op de iPhone getoond tussen de boven- en onderbalken van Safari (met onder meer



(a) Een gewone website (bron: [12]).

(b) Een webapplicatie met een eigen navigatiebalk (bron: [14]).

Figuur D.5: De navigatiebalk van Safari kunnen worden uitgeschakeld.

navigatieknoppen en de mogelijkheid om een URL in te voeren). Voor een webapplicatie is die weergave minder geschikt. Safari Mobile biedt daarom de mogelijkheid om de weergave van webapplicaties aan te passen, zodat een webapplicatie bijna hetzelfde werkt als de native iPhone-applicaties.

Safari biedt hiervoor de volgende mogelijkheden:

- De status- en navigatiebalken die Safari standaard laat zien, kunnen worden uitgeschakeld of aangepast.
- De webapplicatie kan in plaats daarvan een eigen titelbalk weergeven, met een voor de applicatie geschikte terug-knop.
- De webapplicatie kan, als een zogeheten 'web clip', worden geïnstalleerd op het homescherm van de telefoon.
- Met behulp van de juiste viewportinstellingen kunnen webpagina-functies zoals zoomen worden uitgeschakeld.

## URL-schemes

De webapplicatie kan met de ingebouwde applicaties op de telefoon worden geïntegreerd door in de links in de applicatie speciale URL-formaten te gebruiken. Hiermee kan bijvoor-



beeld een telefoongesprek worden gestart of kunnen sms'jes of e-mails worden verstuurd: [15]

- `mailto:` opent het e-mailprogramma
- `tel:` begint een telefoongesprek
- `sms:` begint een sms-bericht

### Offline gebruik

Safari Mobile biedt de mogelijkheid om webapplicaties offline te gebruiken. De applicatie en de gegevens worden dan opgeslagen op de telefoon, zodat de applicatie ook zonder internet-toegang kan worden gebruikt. De applicatiebestanden kunnen hiervoor worden gecached via een zogeheten Manifest-bestand. De data van de gebruiker kunnen worden opgeslagen in een 'JavaScript Database'. [16] Met die voorzieningen is het mogelijk om een eenvoudige webapplicatie offline te gebruiken, maar omdat de verbinding met internet ontbreekt is dit waarschijnlijk niet voor iedere applicatie een goede optie.

### Ondersteuning voor iPhone-specifieke interacties

Via een speciale voorziening kunnen webapplicaties reageren op de specifieke iPhone-invoer, zoals touch-events of oriëntatieveranderingen. Safari Mobile biedt een aantal extra events waarmee de applicatie via JavaScript op dit soort invoer kan reageren. (Deze events zijn overigens Apple-specifiek: ze behoren niet tot de officiële JavaScript-standaard en worden dus op andere telefoons niet zonder meer ondersteund.)

De volgende iPhone-specifieke events zijn beschikbaar in iPhone OS 2:

- `orientationchange` vindt plaats als de telefoon wordt gedraaid
- `touchstart`, `touchmove`, `touchend`, `touchcancel` vinden plaats als het scherm wordt aangeraakt
- `gesturestart`, `gesturechange`, `gestureend` vinden plaats als de gebruiker een specifieke beweging maakt over het scherm

Safari Mobile bevat daarnaast de mogelijkheid om een aantal visuele effecten te gebruiken. Transitie, animaties of de bekende 'Apple-reflectie' onder een plaatje kunnen via JavaScript of CSS eenvoudig worden aangeroepen. [17]

## D.5 Framework

In dit hoofdstuk doen wij verslag van het onderzoek naar de geschikte frameworks om de user-interface te koppelen aan de API van Greetingq.

### D.5.1 Criteria

De frameworks moet aan de volgende eisen voldoen: (De criteria zijn deels ontleend aan het overzicht in [18].)

**Java.** De bestaande software van Greeting is ontwikkeld in Java, dus is het handig om de nieuwe webapplicatie ook in Java te ontwikkelen.

**MVC-architectuur.** De Model-View-Controller-indeling is een veelgebruikt model dat een goede structuur biedt voor de ontwikkeling van webapplicaties.

**Uitgebreide documentatie.**

**Veelgebruikt framework.**

**Goede testbaarheid.** Het framework moet goede ondersteuning bieden voor unit-testing.

**Goede URL's.** Het framework moet voldoende mogelijkheden bieden om de URL's op een goede manier in te delen.

**Flexibele templates.** Het templatesysteem van het framework moet flexibel genoeg zijn om de interface te kunnen implementeren.

**In-/uitvoer moet controleerbaar zijn.** Om het contact tussen de server en de iPhone goed te kunnen regelen, moet de in- en uitvoer van het framework te beïnvloeden zijn.

Aan de hand van deze eisen hebben we de onderstaande twee frameworks geëvalueerd. De frameworks zijn gekozen omdat het de meestgebruikte frameworks zijn en bij alle commentaren als goed naar boven komen.

### D.5.2 Struts 2

Struts 2 [19] voldoet redelijk goed aan de bovenstaande eisen (zie ook het boek [20]). Struts 2 is een veelgebruikt framework met goede documentatie. Er zijn, via het Spring-framework, voorzieningen voor unit-testing. Struts 2 geeft voldoende mogelijkheden om, waar nodig, de in- en uitvoer in detail te beïnvloeden. Het framework gebruikt JSP-templates, maar kan ook worden gebruikt in combinatie met andere templatesystemen.

### D.5.3 Spring MVC

Spring MVC [21] is het Model-View-Controller-framework van het Spring-platform. Ook Spring MVC is een veelgebruikt framework met goede documentatie [22]. Het voldoet aan de gestelde criteria.

### D.5.4 Vergelijking

Struts 2 en Spring MVC lijken allebei geschikt voor dit project. Beide frameworks bieden ongeveer dezelfde functionaliteit en verschillen vooral wat betreft aanpak en architectuur. Welk framework de voorkeur heeft, hangt af van ontwerpkeuzes die pas op een later moment in het project worden gemaakt.

## D.6 Conclusie

Greetingq ontwikkelt Greetingq Voicemail: persoonlijke voicemail met veel extra functionaliteiten. Een Greetingq-gebruiker kan bijvoorbeeld per contact een persoonlijke welkomstboodschap inspreken. Tot nu toe kunnen de boodschappen alleen via een gewone computer worden ingesteld. Voicemail is echter bij uitstek een product dat je ook mobiel zou willen gebruiken. Greetingq wil de applicatie daarom ook voor mobiele telefoons beschikbaar maken.

De opdracht was dan ook om een mobiele applicatie te ontwikkelen. Vanwege de uitgebreide mogelijkheden die deze telefoon biedt, is er voor gekozen om de applicatie in eerste instantie specifiek te bouwen voor de iPhone.

Aan de ontwikkeling van een applicatie die op de iPhone kan worden geïnstalleerd kleven enkele nadelen, onder andere dat de applicatie door Apple moet worden goedgekeurd. Het alternatief, een webapplicatie die niet geïnstalleerd hoeft te worden, lijkt ons geschikt.

Apple heeft veel aandacht besteed aan het oplossen van problemen die ontstaan als je op een klein scherm en met beperkte geheugencapaciteit werkt. Ook is er goed nagedacht over een iPhone-stijl. Om een goede webapplicatie te bouwen zullen wij zo veel mogelijk moeten aansluiten bij de bestaande oplossingen.

De keuze voor een webapplicatie zorgt voor een aantal beperkingen. Zo is het niet mogelijk om geluid op te nemen of bestanden te uploaden. Hierdoor zal het moeilijk zijn om via de webapplicatie nieuwe begroetingen in te stellen. Ook is het niet zonder meer mogelijk om gegevens op te vragen uit ingebouwde applicaties zoals het telefoonboek.

Wij denken dat wij een product kunnen ontwerpen dat voldoet aan de eisen van een mobiele applicatie van Greetingq Voicemail van Greetingq.



# Plan van aanpak E

---

## E.1 Managementsamenvatting

Dit Plan van Aanpak beschrijft de aanpak van het bachelorproject uitgevoerd bij Greeting met als einddoel een webapplicatie voor de iPhone waarmee het product Greeting Voicemail gebruikt kan worden. Het bachelorproject loopt over een tijd van twaalf weken en zal eind juni afgerond zijn. In de laatste week zal een werkend eindproduct worden gepresenteerd.

## E.2 Inleiding

### E.2.1 Aanleiding

Deze opdracht is opgesteld door Greeting omdat ze het product Greeting Voicemail beschikbaar willen maken voor de mobiele telefoon. Deze opdracht zal worden uitgevoerd in het kader van het bachelorproject van de opleiding Media en Kennistechnologie van de Technische Universiteit Delft.

### E.2.2 Accordering en bijstelling

Dit Plan van Aanpak is opgesteld door Roland Heinrichs en Gijs van Tulder. Het plan wordt besproken met de projectbegeleiders, Richard Stronkman en de heer Rothkrantz.

### E.2.3 Toelichting op de opbouw van het plan

Dit Plan van Aanpak bestaat uit vijf hoofdstukken. In het hoofdstuk 'Projectopdracht' worden de doelstelling van het project en de projectopdracht vastgesteld. Er wordt een overzicht gegeven van de op te leveren producten, de eisen en beperkingen waaraan tijdens het project moet worden voldaan. Het hoofdstuk 'Aanpak' beschrijft de globale aanpak en indeling van het project. Vervolgens worden in het hoofdstuk 'Projectinrichting en voorwaarden' de voorwaarden beschreven waaraan opdrachtgever en opdrachtnemer moeten voldoen. De voorlopige planning van het project wordt gegeven in het hoofdstuk 'Planning'. Het hoofdstuk 'Kwaliteitsborging' beschrijft ten slotte aan welke kwaliteitseisen het project moet voldoen en welke maatregelen worden genomen om aan die eisen te voldoen.

Dit Plan van Aanpak is gebaseerd op het oriëntatieverslag, waarin de opdracht is onderzocht en nader wordt toegelicht.

## E.3 Projectopdracht

### E.3.1 Projectomgeving

Greeting ontwikkelt een persoonsgerichte voicemaildienst, waarin onder andere persoonlijke begroetingen kunnen worden ingesteld. Omdat voicemail bij uitstek een product is dat ook mobiel gebruikt zou kunnen worden, wil Greeting graag een mobiele webapplicatie ontwikkelen.

### E.3.2 Doelstelling project

Greeting wil graag een toegankelijk product bieden aan de klant. Als het product mobiel kan worden gebruikt zou dit de toegankelijkheid en bruikbaarheid sterk vergroten. De doelstelling van dit project is het ontwikkelen van een mobiele interface voor zakelijke gebruikers.

### E.3.3 Opdrachtformulering en op te leveren product

De opdracht is het ontwerpen en vervolgens bouwen en aansluiten van een webapplicatie voor de iPhone. Deze applicatie moet de belangrijkste functies van het Greeting Voicemail mobiel bereikbaar maken.

Het resultaat van het project is een bruikbare webapplicatie die Greeting Voicemail beschikbaar maakt voor de iPhone.

### E.3.4 Eisen en beperkingen

1. De webapplicatie bevat de belangrijkste functionaliteiten van Greeting Voicemail
2. De webapplicatie is bruikbaar door de doelgroep
3. Bij het ontwerp van de webapplicatie is, waar mogelijk, rekening gehouden met een toekomstige uitbreiding naar andere mobiele telefoons.

De opdrachtgever wil het project graag in twee delen uitvoeren. Het eerste deel van het project bestaat uit een ontwerpfase, waarin eerst wordt onderzocht welke functies de applicatie zou moeten bevatten en vervolgens de interface voor deze applicatie wordt ontworpen en getest. In ieder geval wordt bekeken welke van de bestaande functies in de mobiele versie geïmplementeerd moeten en kunnen worden. Daarnaast wordt gekeken welke functies zouden kunnen worden toegevoegd.

Het tweede deel van het project bestaat uit de implementatie van het gemaakte ontwerp en de koppeling van die implementatie aan de backend van Greeting.

### E.3.5 Cruciale succesfactoren

Het optimale gebruik, en dus de kennis, van de mogelijkheden van de iPhone en Safari Mobile zijn cruciaal voor het ontwerpen van een goede interface. Vervolgens zal energie moeten worden gestoken in de implementatie van de serverkant van de interface en de koppeling met de bestaande systemen.

### E.3.6 Aanpak

Het project wordt aangepakt in de volgende drie fasen: oriëntatie, ontwerpen en bouwen en koppeling.

#### Oriëntatie

In het eerste deel van de oriëntatiefase onderzoeken we het product Greeting Voicemail en de mogelijkheden van de iPhone en de bijbehorende webbrowsers Safari Mobile. In het tweede deel onderzoeken we de frameworks die voor de koppeling van de interface met de bestaande backend zorgen.

## Ontwerpen en bouwen

Deze fase bestaat uit drie stappen. Daarbij is het mogelijk om een stap terug te nemen als er belangrijke nieuwe inzichten worden verkregen. Iedere stap bestaat uit het opstellen en vervolgens testen van het ontwerp. De testresultaten worden gebruikt bij het ontwerp in de volgende iteratie. De ontwerpen zullen worden besproken met de opdrachtgever en zullen, als mogelijk, worden getest met enkele potentiële gebruikers.

De eerste stap is het inventariseren van de eigenschappen en wensen van de doelgroep. Op basis daarvan worden de benodigde functionaliteiten opgesteld.

De tweede stap is het maken van een papieren user interface. Hierdoor is het mogelijk om inzicht te krijgen in de verwachtingen van de gebruiker.

De derde stap resulteert in een clickable user interface. Hierdoor kan de gebruiksvriendelijkheid worden getest.

## Koppeling

Deze laatste fase zal bestaan uit het koppelen van de clickable interface aan de backend van Greeting. Hiervoor zal een bestaand framework worden gebruikt. In deze fase zal het product als geheel worden getest en zullen er wellicht nog enkele kleine ontwerpveranderingen worden gemaakt.

## E.4 Projectinrichting en voorwaarden

### E.4.1 Projectinrichting

#### Organisatie

Roland Heinrichs. Projectuitvoerder.

Gijs van Tulder. Projectuitvoerder.

Richard Stronkman. Projectbegeleider vanuit Greeting.

Prof. dr. Leon Rothkrantz. Projectbegeleider vanuit TU Delft.

#### Informatie

De uitvoerders overleggen bijna dagelijks met Richard Stronkman. Enkele keren tijdens het project vindt overleg plaats met Leon Rothkrantz.

#### Techniek

De benodigde hardware bestaat uit het beschikbaar zijn van een iPhone en een (test-)server. De benodigde software is vrij beschikbaar.

### E.4.2 Voorwaarden aan opdrachtnemer

Aan de opdrachtnemer worden de volgende voorwaarden gesteld: het project moet eind juni 2009 zijn afgerond met een werkend eindproduct; de geschreven programmacode moet goed gedocumenteerd zijn om verdere uitbreiding van het programma mogelijk te maken en het project wordt afgerond met een eindverslag en een presentatie.

### E.4.3 Voorwaarden aan opdrachtgever

Aan de opdrachtgever worden de volgende voorwaarden gesteld: de opdrachtgever zorgt voor een geschikte werkomgeving; de opdrachtgever zorgt voor de informatie en documentatie die nodig is; de opdrachtgever zorgt dat de bestaande systemen, waar nodig en indien mogelijk, tijdig worden uitgebreid en de opdrachtgever zorgt voor een iPhone (of vergelijkbaar) om het product te kunnen testen.

## E.5 Planning

Het project kent een globale planning, aan het begin staat nog niet exact vast wanneer welke deadlines gaan vallen. De planning staat in bijlage A.

Zelf hebben we geen financiële middelen tot onze beschikking. Wij hebben deze ook verder niet nodig. Vanaf de tweede ontwerpfase (week 4) zullen wij moeten beschikken over een iPhone of op zijn minst een iPod touch, zodat de clickable user interface goed getest kan worden.

## E.6 Kwaliteitsborging

### E.6.1 Productkwaliteit

**Functionaliteit** Het product moet de belangrijkste functionaliteit van Greeting Voicemail bevatten. Het systeem moet correct werken. Bij eventuele externe fouten moet het systeem op de juiste manier reageren en de gebruiker een foutmelding laten zien.

**Betrouwbaarheid** Het is belangrijk dat het product betrouwbaar werkt, omdat de zakelijke gebruiker niet zonder zijn voicemail kan. Als de telefoon geen internetverbinding heeft of in het geval van een storing bij Greeting kan de betrouwbaarheid niet worden gegarandeerd.

**Onderhoudbaarheid** De code van het systeem moet goed worden gedocumenteerd.

**Bruikbaarheid** De zakelijke gebruiker moet de webapplicatie intuïtief en eenvoudig kunnen gebruiken.

**Portabiliteit** De webapplicatie wordt in principe ontwikkeld voor de iPhone. De applicatie zal zo worden ingericht dat hij uitbreidbaar is naar andere telefoontypes.

### E.6.2 Proceskwaliteit

Voor een efficiënt ontwerp- en ontwikkelproces moet aan de volgende eisen worden voldaan: er moet een goed versiebeheer plaatsvinden; de software moet goed worden gedocumenteerd; de software moet automatisch worden getest en er moet zo veel mogelijk gebruik worden gemaakt van bestaande software.



### **E.6.3 Voorgestelde maatregelen**

Bij het ontwikkelen van de software wordt voor het versiebeheer gebruik gemaakt van Subversion. De Java-code wordt gedocumenteerd volgens het JavaDoc-formaat. Voor het testen van de ontwikkelde software zal een geautomatiseerd systeem worden gebruikt. Bij het ontwikkelen van de webinterface wordt zo veel mogelijk gebruik gemaakt van de bestaande standaarden.

Voor de koppeling van de user interface aan de backend zal een bestaand framework worden gebruikt.



# Bibliografie

---

- [1] 37signals, *Getting Real: The smarter, faster, easier way to build a successful web application*. 37signals, 2006.
- [2] G. van Tulder and R. Heinrichs, "Oriëntatieverslag," april 2009.
- [3] H. Sharp, Y. Rogers, and J. Preece, *Interaction design: Beyond human computer interaction*. John Wiley & Sons, 2007.
- [4] G. van Tulder and R. Heinrichs, "Eiseninventarisatie," april 2009.
- [5] G. van Tulder and R. Heinrichs, "Scenario's," april 2009.
- [6] G. van Tulder and R. Heinrichs, "Ontwerpideeën," april 2009.
- [7] G. van Tulder and R. Heinrichs, "Ontwerpideeën 2," april 2009.
- [8] Apple, *iPhone Human Interface Guidelines*. 2009.
- [9] Greeting, "Greeting huisstijl document," april 2009.
- [10] PhoneGap. 2009. <http://www.phonegap.com/>.
- [11] A. Tolsma, *Greeting Voicemail in 400 woorden*. 2009.
- [12] Apple, *iPhone Human Interface Guidelines for Web Applications*. 2009.
- [13] Apple, *Safari Web Content Guide for iPhone OS*. 2009.
- [14] Ajaxian, *iPhone Web Apps running at full screen*. <http://ajaxian.com/archives/iphone-full-screen-webapps>.
- [15] Apple, *Apple URL Scheme Reference*. 2009.
- [16] Apple, *Safari Client-Side Storage and Offline Applications Programming Guide*. 2009.
- [17] Apple, *Safari Visual Effects Guide*. 2009.
- [18] M. Raible, *Comparing Web Frameworks*. 2006. <http://www.chariotsolutions.com/slides/pdfs/spring-forward-2006-web-frameworks.pdf>.
- [19] *Struts 2.1.6*. 2009. <http://struts.apache.org/2.1.6/>.
- [20] I. Roughley, *Starting Struts 2*. InfoQ, 2006. <http://infoq.com/minibooks/starting-struts2>.
- [21] *Spring*. 2009. <http://www.springsource.org/>.
- [22] *The Spring Framework - Reference Documentation*. 2008. <http://static.springframework.org/spring/docs/2.5.x/reference/>.