

Afstudeerscriptie



Multiplatform front-end development met tools/frameworks

Bedrijfsbegeleider:

Dhr. Ir. L.C. van Mourik

E-mailadres:

L.vanmourik@Flusso.nl

Afstudeerbegeleider:

Dhr. Ir. H.B.A. Beerlage

E-mailadres:

Harry.beerlage@hu.nl

Eerste examiner:

Dhr. drs. G. Ovink

E-mailadres:

Gerald.ovink@hu.nl

Student:	E-mailadres:	Opleiding	Student nummer:
Tom Oosterwijk	Tom.oosterwijk@hotmail.com	Technische Informatica	1608726

Datum: 02-06-2015

Periode: 02-02-2015 / 02-06-2015

Versie:	Datum:	Aanpassingen:
0.1	24-02-2015	Concept Versie
0.2	25-02-2015	Eerste invulling informatie uit PVA
0.3	16-03-2015	Toevoeging deelvragen + Bijlage nummering
0.4	07-05-2015	Verwerking alle deelvragen
0.5	08-05-2015	Conclusie & aanbeveling toegevoegd
0.6	12-05-2015	Aanpassen Conclusie en samenvatting
0.7	13-05-2015	Evaluatie toegevoegd
0.8	19-05-2015	Verwerking feedback
0.9	25-05-2015	Verwerking feedback Harry Beerlage
1.0	02-06-2015	Finale versie

I. Voorwoord

Tijdens mijn afstudeerstage zijn er verschillende mensen geweest die mij hebben ondersteund bij de realisatie van dit onderzoek. Allereerst wil ik mijn docentbegeleider Harry Beerlage en mijn bedrijfsbegeleider Lars van Mourik bedanken. Zij hebben mij geholpen mijn afstudeeropdracht vorm te geven en mij het vertrouwen gegeven dat ik nodig had om het afstudeeronderzoek te kunnen verrichten.

Daarnaast wil ik Frank Staijen, een collega bij Flusso, bedanken. Wanneer Lars afwezig was, kon ik altijd terecht bij hem voor de meest uiteenlopende vragen. Ook heeft Frank de tijd genomen om mij dingen te vertellen en leren die ik niet direct tijdens mijn stage kon inzetten, maar wel in mijn toekomstige carrière.

Als laatste wil ik mijn familie en vrienden bedanken voor de steun die zij mij gaven en voor de tijd en moeite die zij hebben gestoken in het doorlezen van mijn verslagen.

II. Managementsamenvatting

In dit onderzoek naar *multiplatform* front-end development met tools en frameworks is onderzocht hoe de ontwikkeling van hybride mobiele applicaties binnen Flusso kan worden verbeterd. De wens om de hybride applicatie ontwikkeling te versnellen door gebruik te maken van hulpmiddelen was de aanleiding om dit onderzoek uit te voeren. Het doel van dit project was om een advies aan Flusso te geven over het gebruik van bestaande tools/frameworks om de *front-end* realisatie van hybride *multiplatform* applicaties te versnellen. Om tot een gedegen advies te komen is een lunch applicatie gecreëerd met behulp van drie verschillende *tools/frameworks* die het beste bij Flusso passen.

Voor dit onderzoek is de volgende hoofdvraag opgesteld: “Op welke wijze kan aangetoond worden welke bestaande tools en/of frameworks het beste Flusso kan ondersteunen bij het realiseren van een *multiplatform* hybride mobile front-end applicatie?”. Bij de inventarisatie van de tools en frameworks zijn platformen geïdentificeerd die interessant leken voor het onderzoek. Deze tools, frameworks en platformen zijn opgedeeld in drie verschillende type hulpmiddelen en beoordeeld aan de hand van de eisen die met een intern interview tot stand zijn gekomen. Vervolgens is van elke type hulpmiddel het best beoordeelde hulpmiddel geselecteerd.

Type	Ondersteunende hulpmiddelen	Genererende hulpmiddelen	Totale hulpmiddelen
Gekozen	Ionic	Intel XDK	OutSystems

Om de drie geselecteerden hulpmiddelen te kunnen beoordelen, is elk gebruikt voor de realisatie van een “lunch applicatie” proof of concept. Van de lunch applicatie is een technisch/functioneel rapport gemaakt dat aangehouden moest worden tijdens de realisatie. Van elk hulpmiddel zijn de ontwikkelingen bijgehouden, zijn de sterke en zwakke punten geïdentificeerd en is gedocumenteerd in hoeverre de realisatie succesvol is geweest aan de hand van het technisch rapport. Uit de resultaten bleek dat het gratis Intel XDK framework niet geschikt is voor het maken van specifieke applicaties. Hoewel er met de *drag & drop editor* snel een applicatie te ontwikkelen valt, werkt de editor niet wanneer er buiten de editor om gewerkt moet worden. Met het betaalde platform OutSystems is een werkende lunch applicatie gemaakt. Door het kunnen genereren van code, een *drag & drop editor* en *model driven development* werd de applicatie ontwikkeling versneld. Wanneer er voor het eerst met deze functionaliteiten gewerkt wordt, levert dit geen significante tijdswinst op, waardoor de kosten van OutSystems hoger zijn dan de behaalde tijdswinst. Daarnaast staat OutSystems nog teveel in de kinderschoenen om ondersteuning te kunnen geven aan hybride applicaties met diverse *native* functionaliteiten. Het gratis framework Ionic kwam als beste uit de test. Ionic ondersteunt de ontwikkelaar met het grote arsenaal aan userinterface objecten die gebruikt kunnen worden. De lunch applicatie voldeed volledig aan het technisch/functioneel ontwerp en is als enige hulpmiddel binnen de gestelde tijdsperiode ontwikkeld.

Derhalve wordt Ionic aangeraden als framework om te gebruiken binnen Flusso voor het ontwikkelen van hybride *multiplatform* applicaties. Daarnaast wordt aangeraden om het platform OutSystems opnieuw te testen wanneer deze gegroeid is in de ondersteuning van hybride applicaties. Verwacht wordt dat er met OutSystems een grotere tijdswinst behaald kan worden, wanneer het platform vaker wordt gebruikt. Wanneer de behaalde tijdswinst groter is dan de kosten die aan het OutSystems platform verbonden zitten, zou OutSystems het meest geschikte hulpmiddel voor Flusso kunnen worden.

Inhoud

I.	Voorwoord	2
II.	Managementsamenvatting	3
1	Inleiding	6
2	Context	8
2.1	Flusso	8
2.2	De bedrijfsbegeleiding	8
2.3	Het afstudeerproject	9
2.4	Functie en verantwoordelijkheden	9
2.5	Communicatie	9
2.6	Stakeholders	9
3	De (advies/ontwerp)opdracht	10
3.1	De kwestie	10
3.2	Doelstelling	11
3.3	Opdrachtomschrijving	11
3.4	Onderzoeksvragen	12
3.5	Afbakening	12
3.6	Methoden en technieken	13
4	Resultaten	15
4.1	Flusso's eisen/wensen	15
4.1.1	Platform wensen	15
4.1.2	Kosten gewaarborgd	15
4.1.3	Maximale kosten	15
4.1.4	Technieken Flusso & klanten	15
4.1.5	Open source	15
4.1.6	Programmeertalen	16
4.1.7	Native functionaliteiten	16
4.1.8	Type applicaties	16
4.1.9	User interface	16
4.2	Lunch applicatie als referentie	17
4.2.1	Qc4u	17
4.2.2	Inspectandgo	18
4.2.3	Lunch applicatie	18
4.2.4	De overeenkomsten/verschillen	19

Multiplatform front-end development met tools/frameworks

4.3	Bestaande hulpmiddelen	20
4.4	Analyse hulpmiddelen	22
4.5	Hulpmiddelen voor Flusso	24
4.6	Bevindingen hulpmiddelen	27
4.6.1	Ionic	28
4.6.2	Intel XDK	31
4.6.3	OutSystems.....	35
4.7	Aansturen hulpmiddelen.....	40
4.7.1	Gegenereerde lay-outs/schermen	40
4.7.2	Native functies.....	40
5	Conclusies.....	42
6	Aanbeveling.....	44
7	Slotopmerkingen	45
8	Bibliography.....	46
9	Bijlagen	48
Bijlage 1.	Plan van aanpak.....	48
Bijlage 2.	Evaluatie	68
Bijlage 3.	Interview Rapport.....	69
Bijlage 4.	Functioneel & Technisch ontwerp.....	79
Bijlage 5.	Inventarisatie hulpmiddelen Tools.....	90
Bijlage 6.	Applicaties van Flusso.....	94
Bijlage 7.	Afbeeldingen Lunch applicatie Ionic	96
Bijlage 8.	Afbeeldingen Lunch applicatie Intel XDK	98
Bijlage 9.	Afbeeldingen Lunch applicatie OutSystems	100

1 Inleiding

Deze afstudeerscriptie is de rode draad die door het onderzoek loopt; van de context en aanleiding van de opdracht, de manier waarop het onderzoek gedaan werd tot de uiteindelijke resultaten, conclusies en aanbevelingen.

Allereerst beschrijft hoofdstuk 2 de projectcontext; hieronder valt de achtergrond van het bedrijf waar de opdracht uitgevoerd is, de positie en taken van de afstudeerder binnen dit bedrijf en vervolgens de relatie van de afstudeeropdracht tot andere projecten binnen het bedrijf. Hoofdstuk 3 behandelt de projectdefinitie; dit houdt de achtergrond, aanleiding tot de opdracht, de opdracht zelf en de doelstellingen van de opdracht in. Nadat de context van het project duidelijk is, bespreekt hoofdstuk 4 de resultaten die tot stand zijn gekomen bij het beantwoorden van de deelvragen en het realiseren van de proof of concept. De resultaten worden in hoofdstuk 5 gebruikt voor conclusies. Deze conclusies geven antwoord op zowel de deelvragen als de hoofdvraag. Hoofdstuk 6 adviseert Flusso wat zij het beste kunnen doen met de conclusies van dit onderzoek. Tot slot vindt in hoofdstuk 7 een evaluatie plaats over de belangrijkste keuzes die tijdens het onderzoek zijn gemaakt.

Tijdens het uitvoeren van het onderzoek ontstonden twee punten die afweken van het Bijlage 1.

Plan van aanpak. De eerste afwijking was het toevoegen van platformen als potentieel hulpmiddel voor Flusso. De platformen waren in eerste instantie niet opgenomen, maar bleken mogelijk te passen bij Flusso. Hoewel platformen aanzienlijk duurder zijn dan de tools en frameworks, hadden deze ook veel meer beschikbare functionaliteiten, waaronder *Model Driven Development (MDD)*. Met MDD zijn de functionele medewerkers van Flusso in staat een applicatie te ontwikkelen, zonder verdere verstand van de techniek te hebben. Dit wordt geregeld door de platformen. Daarnaast bieden platformen support voor en na de ontwikkeling, en niet alleen tijdens de ontwikkeling zoals de tools en frameworks. Een totaal pakket voor de ontwikkeling/beheer van hybride applicaties.

De tweede afwijking ontstond tijdens het maken van de lunch applicatie met het framework Intel XDK. Het framework Intel XDK werd geselecteerd als genererend hulpmiddel, dat door middel van een *drag and drop* interface de applicatie ontwikkeling sneller moest maken. Binnen Intel XDK applicaties wordt er alleen genavigeerd door een `onClick` methode aan een bepaalde knop te koppelen. Wanneer deze knop wordt ingedrukt, navigeert de applicatie naar de pagina die in de `onClick` methode is aangegeven. Met de lunch applicatie moet navigeren ook mogelijk zijn wanneer er op een lijst 'item' gedrukt wordt. Om deze navigatie te kunnen beheren, is het mogelijk om de binnen Flusso gewenste JavaScript taal Angular te gebruiken. Aangezien Angular als standaard binnen Intel XDK geïmplementeerd is, leek dit de ideale keuze. Wanneer externe Angular code echter werd geïmplementeerd, begonnen functionaliteiten van de app designer niet meer te werken. Dit maakte het niet mogelijk om externe Angular JavaScript toe te voegen aan de applicaties van Intel XDK. Wanneer er vanuit het gewone JavaScript werd geprobeerd de navigatie te forceren, bleek dit niet mogelijk doordat Intel XDK gebruik maakt van Anchors. Met de Anchors worden pagina's aan elkaar gelinkt, zodat er tussen pagina's gewisseld kon worden. Wanneer een paginanaam direct werd benaderd, toonde de applicatie een foutmelding. Wanneer er via een Anchor naar precies dezelfde pagina werd verwezen, gaat het proces wel goed. Dit probleem zorgde er voor dat de navigatie van de applicatie gedaan moest worden via ongewenste knoppen met de Intel XDK `onClick` methodes.

Multiplatform front-end development met tools/frameworks

Naast het navigeren lag er een probleem met het weergeven van de informatie die uit de REST-API werd gehaald. Met Intel XDK zijn REST-API calls te binden aan een data object. Dit data object is vervolgens door de gehele applicatie te gebruiken. De data objecten kunnen echter alleen hun data geven aan specifieke *user interface* objecten, zoals een lijst. De informatie tonen in een simpel tekstveld is niet mogelijk. Hierdoor kon het originele design van de lunch applicatie niet worden aangehouden en kon veel informatie daardoor niet gezien worden. Hoewel deze REST-API calls ook te maken zijn vanuit Angular JavaScript, komt het eerste genoemde probleem weer om de hoek kijken. Om deze twee redenen is de lunch applicatie van Intel XDK niet opgeleverd als proof of concept.

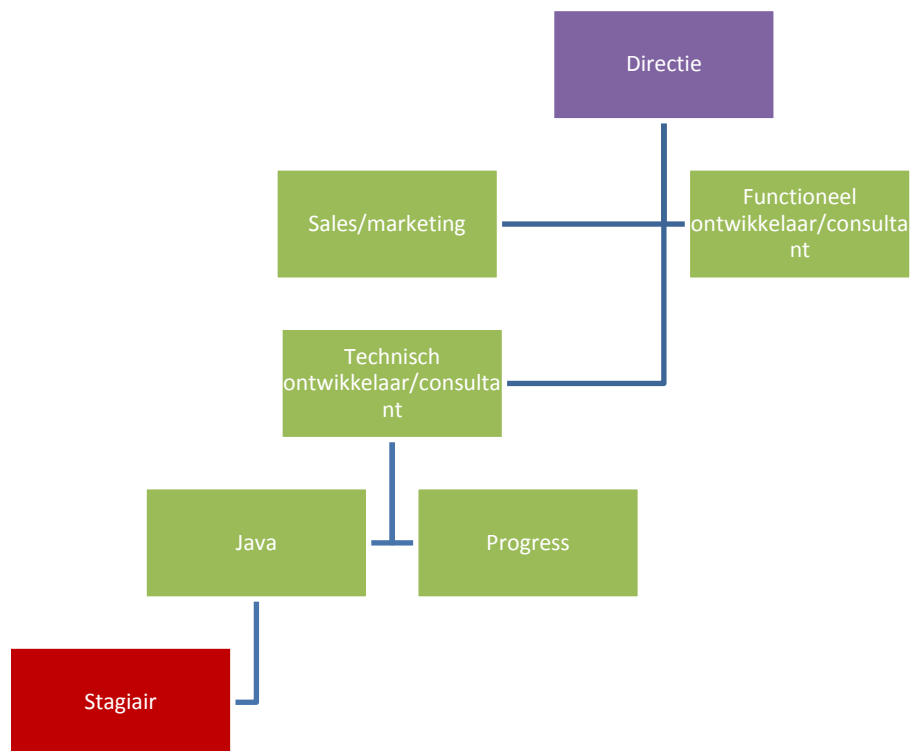
2 Context

In dit hoofdstuk zal achtergrondinformatie gegeven worden over het bedrijf waar de afstudeerstage plaats heeft gevonden, de bedrijfsbegeleiding vanuit Flusso, het projectkader en de positie van de stagiar in het bedrijf.

2.1 Flusso

Het afstudeerproject heeft plaats gevonden bij het bedrijf Flusso. Flusso is een maatwerk applicatieontwikkelaar en ICT dienstverlener die meedenkt en adviseert over de architectuur van applicaties. Flusso ontwerpt, ontwikkelt en verbetert maatwerkapplicaties en zorgt door middel van integratieoplossingen voor een optimale samenwerking tussen IT oplossingen (Flusso, z.d).

Het hoofdkantoor van Flusso is gevestigd in Leusden en een tweede kantoor staat in Groningen. Over de twee kantoren zijn 50 werknemers verdeeld. De afstudeerstage zal plaatshebben in Leusden, waar de stagiar valt onder de afdeling “Java”. De afdeling wordt vooral ingezet bij het realiseren van maatwerk software voor relatief kleine bedrijven met relatief kleine budgetten.



Figuur 1 Organogram Flusso

2.2 De bedrijfsbegeleiding

Tijdens de afstudeerperiode is de student begeleid door Lars van Mourik. Lars heeft na zijn studie Elektrotechniek aan de Universiteit Twente eerst een aantal jaren gewerkt op het snijvlak van hardware en software met vooral embedded systemen. Deze systemen waren veelal slechts een klein onderdeel van het grotere geheel, dus heeft hij zich erg veel bezig gehouden met de integratie van deze systemen. Vervolgens is de overstap gemaakt naar "pure" software ontwikkeling bij Flusso. Lars is erg breed inzetbaar en houdt zich bezig met vrijwel alle facetten van de materie; van de infrastructuur, servers, integratie met andere systemen tot en met het ontwerpen en ontwikkelen van de applicaties zelf, voornamelijk in Java. De contactgegevens van Lars zijn te vinden in paragraaf 2.7 van het Bijlage 1. Plan van aanpak.

2.3 Het afstudeerproject

Het afstudeerproject begon op 02-02-2015 en liep tot en met 02-06-2015. In deze periode kreeg de student een plek binnen Flusso toegewezen waar hij 8 uur per dag kon werken. De student was vrij in het indelen van zijn begin- en eindtijd zolang de vastgestelde uren aan het eind van de maand waren gemaakt. Met het indelen van de tijd moest de student zelf rekening houden met de openingstijd (08:00) en sluitingstijd (18:00) van Flusso.

Het afstudeerproject was een losstaand project dat geen deel uitmaakte van een ander/groter project. De resultaten van het afstudeerproject kunnen echter wel de basis zijn voor toekomstige opdrachten wanneer een gemakkelijk te gebruiken tool of framework wordt gevonden. Het tool/framework zal dan gebruikt kunnen gaan worden voor andere doeleinden dan alleen dit project.

2.4 Functie en verantwoordelijkheden

De positie van de student was die van ontwikkelaar++ (de ++ is, omdat de student ook betrokken was bij het ontwerp en in de architectuurfase). De student was tijdens zijn stageperiode verantwoordelijk voor het uitvoeren van zijn onderzoek en het realiseren van een daarbij behorende proof of concept. Daarnaast draaide de student mee in het ontwikkelaarsteam binnen Flusso.

2.5 Communicatie

Tijdens het afstuderen heeft de student geregeld contact gehad met de bedrijfsbegeleider, Lars van Mourik. Aangezien de student en de begeleider naast elkaar zaten tijdens de werkuren (indien de begeleider niet bij een klant op locatie was) kon de student op ieder gewenst moment vragen stellen, indien nodig. Om de projectvoortgang te waarborgen was er minimaal één keer per week een gesprek gehouden over de uitgevoerde werkzaamheden en de aankomende planning. Hierdoor waren beide partijen op de hoogte van de voortgang van de student en werd de begeleider in staat gesteld input te leveren op de toekomstige werkzaamheden.

2.6 Stakeholders

Binnen de afstudeeropdracht zijn twee groepen stakeholders die belang hebben bij het onderzoek. De eerste groep zijn de software ontwikkelaars binnen Flusso die aan de hand van het resultaat gebruik kunnen gaan maken van de onderzochte tool/framework.

Een tweede groep is het sales/management team binnen Flusso. Wanneer de applicatie ontwikkeling versneld kan worden, hoeft hierin minder tijd geïnvesteerd te worden en worden dus minder kosten gemaakt.

3 De (advies/ontwerp)opdracht

In dit hoofdstuk wordt toegelicht wat de opdracht was, waarom de opdracht tot stand was gekomen, wat het doel van de student was binnen de opdracht en wordt de opdracht verdeeld in verschillende deelvragen die samen het antwoord op de hoofdvraag moet geven.

3.1 De kwestie

Flusso bouwt regelmatig mobiele applicaties voor klanten en deed dit met de bouwstenen Cordova/Phone Gap (zie paragraaf Bijlage 5. Inventarisatie hulpmiddelen Tools) en de gebruikelijke *Integrated Development Environment* (IDE) zoals Eclipse en Sublime. Binnen de ontwikkeling van mobiele applicaties zijn drie verschillende type applicaties te maken (Charland & Leroux, 2011);

- Native¹ applicaties;
- Web- based² applicaties;
- Hybride³ applicaties.

Voor de native en web-based applicaties beschikt Flusso over ontwerp/ontwikkel tools/frameworks die het realiseren van applicaties hielpen te versnellen. Voor de hybride applicaties worden geen specifieke tools of frameworks gebruikt die het ontwerp of de ontwikkeling versnellen.

Flusso merkte echter dat er in de markt wel tools/frameworks beschikbaar zijn en die gaven aan dat het realiseren van hybride mobiele *front-end* applicaties gemakkelijk(er) en snel(ler) te maken zijn. Dit zou voor Flusso kunnen betekenen dat minder tijd en dus geld geïnvesteerd hoeft te worden in het maken van de hybride *multiplatform* applicaties.

Flusso wilde een onderzoek naar deze tools/frameworks, waarbij gezocht moest worden naar een tool/framework dat past bij de (technische) wensen van Flusso. Daarnaast wilde Flusso dat de gevonden tools/frameworks worden gevalideerd door het realiseren van een proof of concept met de tools en frameworks. De resultaten van het onderzoek en de uitvoering van het proof of concept zullen samen de basis leggen voor het uiteindelijke adviesrapport waarin staat welke tool of framework het beste past bij Flusso.

Bij Flusso was het idee ontstaan om een zogeheten “lunch applicatie” te maken, die gebruikt kon worden als proof of concept. Bij Flusso is er elke dag een bedrijfslunch waar iedereen individueel geld inlegt en een medewerker hiermee inkopen doet. Vanwege de flexibele bezetting zijn er elke dag andere behoeften en omdat is gebleken dat veel mensen geen contant geld op zak hebben, is de lunch pot meestal niet goed genoeg gevuld om boodschappen van te doen. Hierdoor ontstaan allerlei praktische problemen en doordat het saldo niet goed wordt bijgehouden, kan het zijn dat sommige medewerkers (veel) te veel en sommige (veel) te weinig geld inleggen. Door de lunchpot in een applicatie te beheren verwacht Flusso dit praktische probleem op te lossen.

Flusso wilde een technisch/functioneel ontwerp van de lunch applicatie die tijdens het realiseren van het proof of concept aangehouden moet worden. Om zeker te zijn dat de resultaten van het proof of concept relevant is voor Flusso, moest tevens onderzocht worden of de lunch applicatie wel een goed referentiekader was.

Wanneer blijkt dat de tools/frameworks een aanwinst voor het bedrijf kunnen zijn, zullen deze worden geïmplementeerd bij het realiseren van toekomstige applicaties binnen Flusso.

¹ Wordt specifiek voor een bepaald type operating system gemaakt zoals IOS, Android of Windows.

² Wordt gebouwd om ondersteund te worden door browsers.

³ Draait in een native webview met webtechnieken, een combinatie tussen web en native.

3.2 Doelstelling

Het doel van dit project is het opleveren van een advies aan Flusso, te Leusden, op 02-06-2015 over het gebruik van bestaande tools/frameworks om de *front-end* realisatie van hybride *multiplatform* applicaties te versnellen en de lunch applicatie door drie verschillende tools/frameworks die het beste bij Flusso passen op te leveren.

3.3 Opdrachtomschrijving

De opdracht die gerealiseerd is, bestond uit twee onderdelen. Het eerste deel was het onderzoek naar de bestaande *multiplatform* tools/frameworks die het realiseren van mobiele *front-end* applicaties ondersteunen. Daarnaast werd de bruikbaarheid van de geselecteerde tool(s) en framework(s) gevalideerd door het realiseren van de “lunch applicatie”.

De lunch applicatie bestaat uit de volgende componenten:

- Java back-end met REST-interfaces (en onderliggende database)
- Web/mobile *front-end*

Om de web/mobile *front-end* te kunnen realiseren zal gebruik worden gemaakt van een tool/framework. De bestaande tools/frameworks worden gebruikt om code eenmalig te schrijven en vervolgens op meerdere platforms te kunnen gebruiken (Outtier & Leuven, Vergelijkende studie van cross-platform tools voor de ontwikkeling van native mobiele, z.d.). Wanneer dit gebeurt, is er sprake van *multiplatform* development. Het gebruik van *multiplatform* development stelt de maker in staat om op kortere termijn een applicatie te realiseren die meteen op meerdere mobiele platformen werkt, in plaats van voor elk platform een custom applicatie te realiseren.

Om te voorkomen dat onderzoek zou worden gedaan naar elk willekeurige tool/framework die bestaat, is een grove selectie gemaakt. Deze selectie is gemaakt aan de hand van wensen/voorkeuren (het platform, de programmeertaal, het target en hardware features) waar de tool/framework aan moet voldoen (Falk & Både, z.d.).

In het verleden is al onderzoek gedaan naar mobiel *multiplatform* tools/frameworks, waarbij vooral gekeken werd naar het mobiele operating system, de kosten van de tool, de programmeertaal, de mogelijkheden van de API naar de hardware componenten en de architectuur en de IDE omgeving (Palmieri, Singh, & Cicchetti, 2012). De resultaten van dat onderzoek zijn meegenomen in dit onderzoek, al werd rekening gehouden met het feit dat deze data al achterhaald konden zijn aangezien de resultaten ruim 2 jaar oud zijn. Om deze reden zijn de resultaten niet zomaar overgenomen, maar zijn ze opnieuw gevalideerd aan de hand van recente literatuur studies.

Van de drie tools/frameworks die als meest optimaal uit het onderzoek naar voren kwamen is een aparte proof of concept (de lunch applicatie) gemaakt. Tijdens het realiseren van de proof of concept werd bijgehouden hoe het proces is verlopen. De resultaten zijn met elkaar vergeleken om een uiteindelijk advies uit te kunnen brengen over de juiste tool/framework voor Flusso.

Inhoudelijke trefwoorden: mobiel Java back-end, mobiel *front-end* tools/frameworks, technische/functioneel ontwerp, advies en ontwerp, hybride mobiel applicatie.

3.4 Onderzoeksvragen

Bij het uitvoeren van de afstudeeropdracht naar mobiele tools/frameworks is de volgende centrale onderzoeksvraag geformuleerd:

- Op welke wijze kan aangetoond worden welke bestaande tools en/of frameworks het beste Flusso kan ondersteunen bij het realiseren van een *multiplatform* hybride mobile *front-end* applicatie?

Om antwoord te kunnen geven op de onderzoeksvraag zijn er verschillende deelvragen geformuleerd:

- Wat zijn de eisen die Flusso stelt aan een tool/framework?
- Op welke wijze past de lunch applicatie als referentie applicatie binnen het ontwikkelteam van Flusso?
- Welke eigenschappen van bestaande tools/frameworks maken deze geschikt om ingezet te worden tijdens de productie binnen het ontwikkelteam van Flusso?
- Wat zijn de sterke en zwakke punten van elk(e) onderzocht(e) tool/framework voor Flusso?
- Welke top drie tools/frameworks zijn het meest geschikt voor de realisatie van applicaties voor Flusso?
- Op welke wijze ondersteunen de top drie tools/framework de realisatie van de lunch applicatie?
- Op welke wijze kunnen er functies toegevoegd worden aan het bestaande tool/framework?

De bijlage Onderzoekschema in het Bijlage 1. Plan van aanpak bevat een schematische weergave van de onderzoeksvraag, de deelvragen en de subdeelvragen. De uitgewerkte subdeelvragen zullen antwoord geven op de deelvragen, die vervolgens gezamenlijk weer antwoord zullen geven op de centrale onderzoeksvraag.

3.5 Afbakening

Gedurende het project zal een top drie van tools/frameworks samengesteld worden die voldoen aan de opgestelde eisen binnen de organisatie van Flusso met betrekking tot *front-end* development. Met de top drie tools/frameworks zal de lunch applicatie proof of concept gemaakt worden en de bevindingen van het gebruik van elke tool/framework worden genoteerd. De uitkomst van de vergelijking tussen de bevindingen zal resulteren in een adviesrapport dat antwoord geeft op de onderzoeksvraag.

3.6 Methoden en technieken

In deze paragraaf wordt beschreven welke methodes en technieken gebruikt zijn om per deelvraag antwoord te kunnen geven op de onderzoeksvraag.

Wat zijn de eisen die Flusso stelt aan een tool/framework?

- Onderzoeksfunctie:
 - o Ontwerpend, de resultaten van dit onderzoek worden gebruikt om de onderzochte tools/frameworks te valideren aan de hand van de wensen van Flusso.
- Onderzoekseenheid:
 - o Kwalitatief onderzoek, door middel van een gesloten interview. Het interview is bij werknemers van verschillende afdelingen afgenomen (management, programmeurs, sales (in verband met wensen klanten)).

Op welke wijze past de lunch applicatie als referentie applicatie binnen het ontwikkelteam van Flusso?

- Onderzoeksfunctie:
 - o Vergelijkend, de lunch applicatie is vergeleken met eerder gemaakte applicaties van Flusso.
- Onderzoekseenheid:
 - o Kwalitatief onderzoek, door middel van een literatuuronderzoek en ontwerp. Er is gekeken naar de gebruikte technieken binnen de gemaakte applicaties en de overeenkomsten die er zijn met de lunch applicatie. Voor de lunch applicatie is een functioneel/technisch ontwerp worden gemaakt om een goed beeld te krijgen van de benodigde technieken.

Welke eigenschappen van bestaande tools/frameworks maken deze geschikt om ingezet te worden tijdens de productie binnen het ontwikkelteam van Flusso?

- Onderzoeksfunctie:
 - o Verkennend, het selecteren van de tools/frameworks die in aanmerking komen binnen Flusso.
- Onderzoekseenheid:
 - o Kwalitatief onderzoek, door middel van een literatuuronderzoek. Om een selectie van tools/frameworks te kunnen maken is gebruikt gemaakt van een bestaande wizard die frameworks valideert (Falk & Både, z.d.).

Een andere bron die gebruikt werd is een wizard van ThoughtsWorks. ThoughtsWorks houdt bestaande/gebruikte en aankomende tools/frameworks in de gaten en laat zien welke tools/frameworks aan te raden zijn en welke minder goed uit de verf komen (ThoughtWorks, z.d.). Naast de twee websites die beschikken over informatie van nieuwe/huidige tools/frameworks is ook gekeken naar eerder onderzochte en gevalideerde tools/frameworks (Palmieri, Singh, & Cicchetti, 2012).

Multiplatform front-end development met tools/frameworks

Wat zijn de sterke en zwakke punten van elk(e) onderzocht(e) tool/framework voor Flusso?

- Onderzoeksfunctie:
 - Vergelijkend, er is een vergelijking gemaakt binnen de gevonden tools/frameworks.
- Onderzoekseenheid:
 - Kwalitatief onderzoek, door middel van een literatuuronderzoek. Bronnen zijn gebruikt die informatie bevatten over de tools/frameworks die aan Apple en Android platformen ondersteuning bieden.

Welke top drie tools/frameworks zijn het meest geschikt voor de realisatie van applicaties binnen Flusso?

- Onderzoeksfunctie:
 - Vergelijkend, de onderzochte tools/frameworks zijn tegenover de wensen van de lunch applicatie gezet.
- Onderzoekseenheid:
 - Kwalitatief onderzoek, de gemaakte analyse van de zwakte/sterkte punten van elke tool/framework is vergeleken met de eisen aan de lunch applicatie. De drie tools/frameworks die het beste uit de vergelijking kwamen zijn gebruikt voor het realiseren van de proof of concept.

Op welke wijze ondersteunen de top drie tools/frameworks de realisatie van de lunch applicatie?

- Onderzoeksfunctie:
 - Uitvoerend, de tools/frameworks zijn gebruikt om de lunch applicatie te maken.
- Onderzoekseenheid:
 - Kwalitatief onderzoek, de bevindingen tijdens het realiseren zijn bijgehouden en de resultaten verwerkt in een adviesrapport.

Op welke manier kunnen er functies toegevoegd worden aan het bestaande tool/framework?

- Onderzoeksfunctie:
 - Ontwerpend, er is vast gelegd hoe buiten de tool/framework om kan worden gewerkt.
- Onderzoekseenheid:
 - Kwalitatief onderzoek, door middel van een literatuuronderzoek en uitvoerend onderzoek. Om de deelvraag te kunnen beantwoorden moest antwoord gegeven worden op de volgende drie criteria:
 - Op welke manier kunnen gegenereerde lay-out's/schermen worden aangepast buiten het framework om?
 - Op welke manier kunnen native functies van mobiele telefoons worden toegevoegd aan de *tool/framework*?
 - Is de manier waarop de native functies worden toegevoegd *compatible* met andere frameworks?

4 Resultaten

In dit hoofdstuk wordt antwoord gegeven op de gedefinieerde deelvragen.

4.1 Flusso's eisen/wensen

Om de tool/framework te selecteren die het beste past bij Flusso, moest er duidelijkheid zijn over de wensen die Flusso heeft. Om deze wensen te achterhalen is een interview afgenomen met medewerkers van Flusso. Van de 50 medewerkers, hebben 21 personen aan de inventarisatie meegewerkt. De antwoorden van de respondenten zijn verwerkt tot een interviewrapport (zie Bijlage 3. Interview Rapport).

Uit het interviewrapport kwamen de volgende eisen/wensen van Flusso naar voren.

4.1.1 Platform wensen

De tool/framework moet naast Apple en Android ook ondersteuning leveren aan de Windows Phone en het liefst ook een Windows Mobile.

4.1.2 Kosten gewaarborgd

Een tool/framework mag geld kosten wanneer Flusso er in verhouding meer geld mee kan verdienen vergeleken met een gratis open source tool/framework of hand maatwerk. Dit kan indien aan de volgende punten wordt voldaan:

- Snelle *time to market*;
- Grote/actieve community/helpdesk;
- Gevalideerde standaard technieken;
- Goed toekomstperspectief.

4.1.3 Maximale kosten

Wanneer een tool/framework aan de gestelde eisen (zie paragraaf 4.1.2) voldoet, mogen aan de realisatie kosten verbonden zijn. De kosten mogen maximaal per jaar per ontwikkelaar tussen de €200 en €500 bedragen. Hoe beter/meer de tool/framework voldoet aan de eisen, hoe hoger het bedrag mag zijn.

4.1.4 Technieken Flusso & klanten

De medewerkers van Flusso hebben er belang bij om de volgende technieken terug te zien in een tool/framework:

- Een bekende programmeertaal (HTML(5), CSS, Java en JavaScript);
- AngularJS framework;
- Ingebakken Unit-testen voor benodigde platforms;
- UML naar code en code naar UML genereren.

De klanten van Flusso hebben er belang bij om de volgende technieken terug te zien in een tool/framework:

- Native look & feel user interface;
- Develop once, deploy to many;
- Geautomatiseerde testen.

4.1.5 Open source

Binnen Flusso is de wens naar een open source tool/framework erg groot. Het is geen absolute eis, maar een open source tool/framework heeft de voorkeur.

4.1.6 Programmeertalen

Binnen Flusso leeft de wens om een programmeertaal te kiezen die al bekend is bij de ontwikkelaars. De programmeertalen die het beste binnen Flusso passen zijn JavaScript, HTML, CSS en Java. Een kleine groep binnen Flusso ziet graag dat Progress/OpenEdge ook bij de standaard programmeertalen hoort.

4.1.7 Native functionaliteiten

Om een hybride applicatie te kunnen bouwen die voldoet aan de minimale native functie calls binnen Flusso, moet de tool/framework minimale ondersteuning bieden aan de gewenste functionaliteiten:

- De camera;
- Notificatie berichten;
- Geolocation;
- File;
- Het opslaan van bestanden.

Daarnaast werd bij de minder populaire native functies nog steeds veel aangegeven dat deze ook ondersteund moet worden. Een grotere dekking van native functies betekent dat de tool/framework beter kan worden ingezet voor diverse projecten.

4.1.8 Type applicaties

Het is binnen Flusso een eis dat de tool/framework in staat moet zijn om support te leveren bij de totstandkoming van een hybride, native en web-based applicatie. Op deze manier hoeft een ontwikkelaar maar één tool/framework goed te kennen om alle applicaties te kunnen ontwikkelen in plaats van meerdere hulpmiddelen.

4.1.9 User interface

Voor de klanten van Flusso is het de wens dat de applicaties een *native look & feel* hebben. Deze wens wordt gezien en gedeeld door de medewerkers van Flusso. De tool/framework moet in staat zijn per ondersteunde operating systeem(Zie paragraaf 4.1.1) een *native look & feel* applicatie te realiseren.

4.2 Lunch applicatie als referentie

Om te kunnen valideren of de gevonden hulpmiddelen geschikt zijn voor de realisatie van een hybride applicatie, moesten ze in staat zijn om de lunch applicatie te maken. Om te kunnen valideren of de lunch applicatie een goede referentie applicatie is binnen Flusso haar ontwikkelteam, wordt de lunch applicatie vergeleken met bestaande Flusso applicaties. Met name de functionaliteiten die verwacht worden van een tool/framework zijn geanalyseerd. Vervolgens worden de gewenste functionaliteiten van de lunch applicatie bepaald en tot slot worden de bestaande functionaliteiten van de applicaties vergeleken met de functionaliteiten van de lunch applicatie om te bepalen of de lunch applicatie door kan gaan als een referentie applicatie.

Om een beeld te krijgen van de applicaties die voorheen zijn gerealiseerd binnen Flusso, zijn twee applicaties onderzocht die Flusso heeft gemaakt in het jaar 2014/2015. De eerste applicatie heet Qc4u en de tweede applicatie heet Inspectandgo. Beide applicaties hebben als doel om het rapporteren van de omgeving van de gebruiker gemakkelijker te maken. Waar Qc4u gemaakt is om het inspecteren van containers gemakkelijker te maken, wordt Inspectandgo gebruikt om huis bezichtigingen sneller te laten verlopen.

4.2.1 Qc4u

Dankzij de informatie van programmeur Martin Diphooorn is een technische specificatie opgesteld waar een tool/framework aan moet kunnen voldoen om de Qc4u applicatie na te bouwen. Het gaat binnen deze lijst om native functies van een mobiel die worden gebruikt. Voor de applicatie Qc4u zijn de benodigde functionaliteiten weergegeven in Tabel 1. De uitleg waar elke functionaliteit voor wordt gebruikt is terug te lezen in Bijlage 6. Applicaties van Flusso.

	Qc4u
Device propertjes	✓
Network information	✓
Network toegang	✓
Camera	✓
Local notification	✓
Barcode scanner	✓
File transfer	✓
Geo location	✓
Touch screen	✓
Swipe	✓

Tabel 1 Functionaliteiten Qc4u

Met het overzicht van de benodigde functionaliteiten van Qc4u, kan gekeken worden of de lunch applicatie genoeg functionaliteiten bezit om als referentie applicatie te dienen.

4.2.2 Inspectandgo

Dankzij de informatie van programmeur Edwin Lijnzaad is een technische specificatie opgesteld waar een tool/framework aan moet kunnen voldoen om de Inspectandgo applicatie na te bouwen. Het gaat om native functies van een mobiel die worden gebruikt. Voor de applicatie Inspectandgo zijn de benodigde functionaliteiten weergegeven in Tabel 2. De uitleg waar elke functionaliteit voor wordt gebruikt is terug te lezen in Bijlage 6. Applicaties van Flusso.

	Inspectandgo
Device propertjes	✓
Network information	✓
Network toegang	✓
Camera	✓
File transfer	✓
Geo location	✓
Touch screen	✓
Swipe	✓
Battery status	✓
Dialogs	✓
File	✓
Globalization	✓
Inappbrowser	✓

Tabel 2 Functionaliteiten Inspectandgo

Met het overzicht van de benodigde functionaliteiten van Inspectandgo, kan gekeken worden of de lunch applicatie genoeg functionaliteiten bezit om als referentie applicatie te dienen.

4.2.3 Lunch applicatie

Om de benodigde technieken voor het realiseren van de lunch applicatie inzichtelijk te maken, is een technisch/functioneel ontwerp gemaakt (zie Bijlage 4. Functioneel & Technisch ontwerp). Hieruit werd duidelijk dat de tool/framework de technieken moet ondersteunen die zijn weergegeven in Tabel 3.

	LunchApp
Device propertjes	✓
Network information	✓
Network toegang	✓
Touch screen	✓
Swipe	✓
File	✓

Tabel 3 Functionaliteiten Lunch applicatie

De genoemde functionaliteiten zijn minimaal vereist om de lunch applicatie te maken. Met de minimale functionaliteiten van de lunch applicatie kan deze worden vergeleken (zie 4.2.4) met de functionaliteiten van de applicaties Qc4u en Inspectandgo. Wanneer blijkt dat de lunch applicatie geen goede referentie applicatie is voor Flusso, zullen de hulpmiddelen getest worden op extra functionaliteiten die zowel Qc4u als Inspectandgo bezitten, maar de lunch applicatie niet.

4.2.4 De overeenkomsten/verschillen

Waar Inspectandgo en Qc4u zijn ontworpen voor het inspecteren van een ongelimiteerde aantal huizen en containers, is de lunch applicatie gelimiteerd met het versturen van informatie. Bij Qc4u is het mogelijk dat er maar twee of drie containers geïnspecteerd hoeven worden, maar het zouden er net zo goed 100 of meer kunnen zijn. De informatie van de lunch applicatie is altijd gelijk aan het aantal werknemers binnen Flusso. Dit maakt dat de applicaties qua grootte niet te vergelijken zijn, maar er zijn wel overeenkomsten in de gebruikte technieken te vinden.

Alle drie de applicaties bestaan uit de drie core onderdelen; een database, een API en een applicatie. Om vanuit de applicatie te communiceren met de database, is er gebruik gemaakt van een REST-API. Deze is vanuit de applicatie te benaderen, indien er toegang tot het internet is. Zo werken de applicaties in de basis hetzelfde. Wanneer echter gekeken wordt naar de beschikbare functionaliteiten die per applicatie aanwezig zijn, is een verschil merkbaar tussen de gemaakte applicaties en de lunch applicatie. Tabel 4 geeft een overzicht van de aanwezige mobiele functionaliteiten per applicatie. Een groen vakje betekend dat de functionaliteit in de applicatie aanwezig is. Bij rood is de functionaliteit niet aanwezig.

	Qc4u	Inspectandgo	Lunch App
Device propertjes	✓	✓	✓
Network information	✓	✓	✓
Network toegang	✓	✓	✓
Camera	✓	✓	✗
Local notification	✓	✗	✗
Barcode scanner	✓	✗	✗
File transfer	✓	✓	✗
Geolocation	✓	✓	✗
Touch screen	✓	✓	✓
Swipe	✓	✓	✓
Batterij status	✗	✓	✗
Dialogs	✗	✓	✗
File	✗	✓	✓
Globalisatie	✗	✓	✗
Inappbrowser	✗	✓	✗

Tabel 4 Vergelijking van functionaliteiten per applicatie

Van de 15 functionaliteiten, werden er maar 6 (40%) gebruikt in de lunch applicatie. In vergelijking gebruikten Qc4u tien (66,66%) en Inspectandgo 13 (86,66%) van de 15 beschikbare functionaliteiten. In verhouding gebruikt de lunch applicatie een stuk minder functionaliteiten dan de voorbeeld applicaties, en kon daardoor niet gezien worden als een goede referentie applicatie. Om toch de lunch applicatie te kunnen gebruiken als referentie applicatie, moest de lunch applicatie meer functionaliteiten bezitten. Uit Tabel 4 is af te lezen dat de volgende functionaliteiten die in beide voorbeeld applicaties worden gebruikt, niet in de lunch applicatie zitten; de camera, file transfer en het gebruik van geolocation. Deze functionaliteiten zijn meegenomen bij de realisatie van de lunch applicatie. Met de geselecteerde hulpmiddelen moet zowel de standaard lunch applicatie te maken zijn, als het toevoegen en gebruiken van de hierboven genoemde gewenste functionaliteiten binnen een applicatie.

4.3 Bestaande hulpmiddelen

Om de juiste tool/framework te kunnen vinden die Flusso helpt om de front-end development te versnellen, is een inventarisatie gemaakt door middel van literatuuronderzoek.

Naast de bestaande tools/frameworks zijn er platformen uit het onderzoek gekomen die volledige ondersteuning leveren aan het realiseren van multiplatform (hybride) applicaties. De ontwikkelplatformen zijn meegenomen in het onderzoek omdat het mogelijk was dat deze Flusso beter zouden kunnen ondersteunen dan de tools/frameworks.

Het criterium om geselecteerd te worden is dat het hulpmiddel de front-end development kan helpen versnellen door multiplatform ontwikkeling mogelijk te maken voor minimaal Android en iOS hybride applicaties.

De gevonden hulpmiddelen, met een daarbij horende beschrijving zijn terug te vinden in Bijlage 5.

Inventarisatie hulpmiddelen

Tools. De resultaten van de inventarisatie zijn schematisch weergegeven in Tabel 5. In Tabel 5 wordt aangegeven; van welke type(tool, framework of platform) het hulpmiddel is, welke ontwikkeltalen er gebruikt kunnen worden binnen het hulpmiddel, hoe de native gewenste functionaliteiten benaderd kunnen worden, voor welke platformen er applicaties gemaakt kunnen worden en welke type (web, hybride of native) applicaties er gemaakt kunnen worden met het hulpmiddel.

Multiplatform front-end development met tools/frameworks

Type	Naam	Ontwikkeltalen	Toegang tot native functies	Ondersteunde Operating Systems	Type applicaties
Tool	Phone Gap / Cordova	HTML, HTML5, CSS en JavaScript	JavaScript	iOS, Android, Blackberry, Windows Phone	Web/Hybride
	Touchdevelop	drag and drop / JavaScript	drag and drop / JavaScript	iOS, Android, Windows Phone	Web / Hybride
Framework	Sencha Touch	HTML, HTML5, CSS en JavaScript	Phone Gap/Apache Cordova	iOS, Android, Blackberry 10, Windows Phone 8.1	Web/Hybride
	Altova-MobileTogether	XPath, XQuery en drag and drop	XPath, XQuery en drag and drop	iOS, Android, Windows 8 en Windows Phone8	Web/Hybride
	Ionic	HTML, HTML5, CSS, JavaScript(Angular)	Apache Cordova (JavaScript)	iOS 6+ en Android 4.0+	Web/Hybride
	Intel XDK	HTML5, CSS en JavaScript	JavaScript	Android, OIS, Windows 8, Amazon en Nook	Web/Hybride
	Mobile Angular	Phone Gap/Cordova	Phone Gap/Cordova	iOS, Android, Blackberry, Windows Phone	Web/Hybride
	Titanium	JavaScript	JavaScript	iOS, Android en Blackberry	Web/ Hybride
	MoSync	C/C++ of HTML5, CSS en JavaScript	Wormhole (JavaScript)	iOS, Android, Windows Mobile, Symbian os, MeeGo, Moblin	Web/Hybride
	RhoMobile	HTML5, CSS, JavaScript en Ruby	RhoElements	iOS, Android, Windows Mobile	Web/Hybride
	Genuitec	Phone Gap	Phone Gap	iOS, Android	Web/Hybride
	Qt	Qt Quick, C++, en JavaScript	Qt Quick	iOS, Android	Web/Hybride
	Supersonic	HTML5, JavaScript	Supersonic AngularJS	iOS,Android	Hybride
Platform	Mendix	Phone Gap	Phone Gap	iOS, Android en Windows	Web/Hybride
	Telerik AppBuilder	HTML5, CSS en JavaScript	Phone Gap/Cordova	iOS, Android en Windows Phone	Web/Hybride
	Xamarin	C#	C#	iOS, Android Windows	Native/Hybride
	IBM MobileFirst platform	HTML5, CSS en JavaScript	JavaScript gebaseerd op Phone Gap	iOS, Android, Windows	Native/Web/Hybride
	OutSystems	Java, C#, HTML5, CSS en JavaScript	Cordova	iOS, Android, Windows Phone	Hybride/Web

Tabel 5 Overzicht tools, frameworks en platforms

Met de hulpmiddelen van Tabel 5 is er een ruwe groep geselecteerd die allemaal mogelijk door Flusso gebruikt konden worden. Om er achter te komen welk hulpmiddel het beste past bij Flusso, zijn de hulpmiddelen beoordeeld (zie paragraaf 4.4) in hoeverre deze voldeden aan de eisen (zie paragraaf 4.1) die zijn gesteld door de werknemers van Flusso. Met de resultaten van de beoordeling bleven er drie hulpmiddelen over (zie paragraaf 4.5), die vervolgens in de praktijk (zie paragraaf 4.6) getest werden.

4.4 Analyse hulpmiddelen

Na de inventarisatie (zie paragraaf 0) van de hulpmiddelen was het van belang dat de hulpmiddelen van elkaar onderscheiden en met elkaar vergeleken konden worden om uiteindelijk drie hulpmiddelen te selecteren voor de realisatie van de lunch applicatie.

Om de gevonden hulpmiddelen met elkaar te kunnen vergelijken, werden ze getest met de eisen/wensen die uit het interne interview van Flusso zijn gekomen (zie paragraaf 4.1). De punten waarop de tools/frameworks en platforms werden beoordeeld zijn hieronder omschreven:

- Platform: Welke platformen worden er ondersteund door de tool/framework?
- Eisen kosten: Voldoet het hulpmiddel aan de gestelde eisen indien er kosten verbonden zijn?
- Kosten: Liggen de kosten van de hulpmiddel binnen het gestelde bedrag vanuit Flusso?
- Flusso's technieken: In hoeverre ondersteunt het hulpmiddel de technische eisen die de werknemers van Flusso terug willen zien?
- Klanten technieken: In hoeverre ondersteunt de hulpmiddel de technische eisen die de klanten van Flusso terug willen zien?
- Open Source: Is de tool/framework open source?
- Talen: In hoeverre worden de gewenste programmeertalen?
- Native func.: In hoeverre worden (de minimaal gewenste) native functionaliteiten ondersteund?
- Type applicaties: In hoeverre ondersteunt het hulpmiddel het realiseren van (alle) verschillende type applicaties?
- UI: Ondersteunt het hulpmiddel het realiseren van een native *look & feel* user interface?

De resultaten zijn schematische weergegeven in Figuur 2. Voor elk onderdeel waarop de hulpmiddelen zijn beoordeeld zijn zes verschillende resultaten te behalen die links onder in Figuur 2 worden uitgelegd. De resultaten worden in paragraaf 4.5 gebruikt om te bepalen welke hulpmiddelen er gebruikt worden voor de realisatie van de lunch applicatie.

Eis/Wens Hulpmiddel	Platform	Eisen Kosten	Kosten	Flusso's technieken	Klanten technieken	Open Source	Talen	Native func.	Type Applicatie	UI
PhoneGap / Cordova	+	++	++	+	+~	+	++	++	+	+~
Touchdevelop	+	-	++	---	-	-	-	+~	+	-
Sencha Touch	+	+	++	+	+	+	++	++	+	++
MobileTogether	++	+~	+	-	+	-	---	-	+	++
Ionic	+	++	++	+	+	+	++	++	+	++
Intel XDK	+	+	++	++	++	-	++	+	+	++
Mobile Angular	+	++	++	+	+	+	++	++	+	+
Titanium	+~	++	++	+	++	+	+~	+~	+	++
MoSync	---	---	++	+~	-	+	++	---	+	-
RhoMobile	+~	+	---	+	++	+	+	+~	+	++
Genuitec	+~	+	+	+	+	-	++	++	+	+~
Qt	+~	+	++	+~	+	+	-	++	+	++
Supersonic	+~	+	++	+~	+	-	+	+	+~	++
Mendix	+	++	---	++	++	-	++	++	+	++
Telerik	+	+	---	++	+	-	++	++	+	+
Xamarin	+	++	---	-	++	-	-	++	++	++
IBM	+	++	*	+	++	-	++	++	++	++
OutSystems	+	++	---	++	++	+~	++	++	+	++
Symbol	Uitleg									
++	Voldoet volledig aan de eisen.									
+	Voldoet aan merendeels van de eisen.									
+~	Voldoet aan de minimale eisen.									
-	Voldoet aan merendeels van de eisen niet.									
---	Voldoet volledig niet aan de eisen.									
*	niet bekend									

Figuur 2 Schematische weergaven van analyse

Multiplatform front-end development met tools/frameworks

Gelet op het doel van de tool/framework of platform, kunnen de hierboven genoemde hulpmiddelen verdeeld worden in drie groepen. Deze drie groepen zijn: ondersteunende hulpmiddelen, genererende hulpmiddelen en totale hulpmiddelen.

Ondersteunende hulpmiddelen

De ondersteunende hulpmiddelen (zie Figuur 3) bieden ondersteuning door middel van user interface objecten aan te bieden. Dit zijn de zogenoemde user interface tools/ frameworks. Deze objecten zijn gemakkelijk te implementeren en stellen de ontwikkelaar in staat om in een korte termijn een mooi user interface te realiseren. De ontwikkelaar wordt dan niet gesteund in het opzetten/maken van de applicatie, maar krijgt wel de middelen om de realisatie te vergemakkelijken en te versnellen.

Eis/Wens	Platform	Eisen Kosten	Kosten	Flusso's technieken	Klanten technieken	Open Source	Talen	Native func.	Type Applicatie	UI
Hulpmiddel										
Sencha Touch	+	+	++	+	+	+	++	++	+	++
Ionic	+	++	++	+	+	+	++	++	+	++
Mobile Angular	+	++	++	+	+	+	++	++	+	+
Titanium	+~	++	++	+	++	+	+~	+~	+	++
MoSync	---	---	++	+~	-	+	++	---	+	-
RhoMobile	+~	+	---	+~	++	+	+	+~	+	++

Figuur 3 Overzicht van ondersteunende hulpmiddelen

Genererende hulpmiddelen

De genererende hulpmiddelen (zie Figuur 4) zijn in staat codes te generen vanuit (simpele) input van de ontwikkelaar. Een voorbeeld hiervan is een *drag and drop* interfaces. Met een *drag and drop* interface sleept de ontwikkelaar de gewenste functionaliteiten naar de applicatie en het tool/framework genereert daar zelf de code bij. Genererende hulpmiddelen zijn uitstekend te gebruiken bij het snel opzetten of realiseren van een simpel project of snel de basis te leggen voor een uitgebreid project.

Eis/Wens	Platform	Eisen Kosten	Kosten	Flusso's technieken	Klanten technieken	Open Source	Talen	Native func.	Type Applicatie	UI
Hulpmiddel										
PhoneGap / Cordova	+	++	++	+	+~	+	++	++	+	+~
Touchdevelop	+	-	++	---	-	-	-	+~	+	---
MobileTogether	++	+~	+	-	+	-	---	-	+	++
Intel XDK	+	+	++	++	++	-	++	+	+	++
Genuitec	+~	+	+	+	+	-	++	++	+	+~
Qt	+~	+	++	+~	+	+	-	++	+	++
Supersonic	+~	+	++	+~	+	-	+	+	+~	++

Figuur 4 Overzicht van genererende hulpmiddelen

Totale hulpmiddelen

De totale hulpmiddelen (zie Figuur 5) bestaan uit platformen die ondersteuning bieden in het totale ontwikkelproces. De platformen leveren een totaalpakket aan beschikbare functionaliteiten dat niet alleen een ontwikkelaar, maar een heel bedrijfsteam kan ondersteunen. Met de platformen wordt een snelle *time to market* beloofd, maar dit gaat ten koste van de flexibiliteit die de ontwikkelaars hebben. Er moet gewerkt worden met de tools die het platform aanbiedt en op de manier zoals het platform dat verwacht.

Multiplatform front-end development met tools/frameworks

Eis/Wens Hulpmiddel	Platform	Eisen Kosten	Kosten	Flusso's technieken	Klanten technieken	Open Source	Talen	Native func.	Type Applicatie	UI
Mendix	+	++	---	++	++	-	++	++	+	++
Telerik	+	+	---	++	+	-	++	++	+	+
Xamarin	+	++	---	-	++	-	++	++	++	++
IBM MobileFirst	+	++	*	+	++	-	++	++	++	++
OutSystems	+	++	---	++	++	+/-	++	++	+	++

Figuur 5 overzicht van totale hulpmiddelen

Twee opvallende resultaten zijn de kosten van het IBM MobileFirst platform (aangemerkt in de Figuur als *) en score van OutSystems bij Open Source (aangemerkt in de Figuur als +/-). De * betekent dat er geen prijsindicatie is gevonden voor het IBM platform. Hierdoor kan het IBM platform niet op de kosten worden beoordeeld.

Bij de Open Source categorie scoort OutSystems als enige hulpmiddel een +/-, waar de andere hulpmiddelen beoordeeld zijn met een + (open source) of – (niet open source). Toch heeft OutSystems een +/- beoordeling gekregen. De beoordeling is gebaseerd op de afspraak die OutSystems maakt met zijn afnemers over het beëindigen van de relatie. Wanneer een klant van OutSystems besluit om niet langer gebruik te maken van het OutSystems platform, biedt OutSystems de optie aan om de gemaakte applicaties om te generen tot Java of .net code. Deze code wordt vervolgens aan de klant overgedragen. Dit stelt de afnemer in staat om verder te kunnen gaan met de gemaakte applicatie en gaat dit het Vendor lock-in⁴ principe tegen. Om deze reden wordt OutSystems beloond met een +/- scoren.

4.5 Hulpmiddelen voor Flusso

Met de analyse, beoordeling en onderverdeling van de verschillende hulpmiddelen kunnen de juiste tool(s)/framework(s) en platform(s) worden geselecteerd voor verder onderzoek.

Om een zo goed mogelijk beeld te krijgen van wat de verschillende hulpmiddelen kunnen betekenen binnen Flusso, werd van elke groep één hulpmiddel getest. Dit wordt gedaan omdat de verschillende tools/frameworks binnen de groepen op elkaar lijken. De manier waarop deze ondersteuning leveren is hetzelfde, alleen verschilt het per hulpmiddel hoe dit tot uiting komt. Door een hulpmiddel van elke groep te kiezen wordt een duidelijk beeld gecreëerd over de toegevoegde waarde die ieder hulpmiddelen biedt.

Van elke groep zal het hulpmiddel geselecteerd worden dat het beste uit de analyse kwam. Om het beste hulpmiddel te kunnen detecteren, zullen de symbolen ++, +, +/-, - en – worden omgezet naar punten met de waardes 2,1,0,-1, -2. Met deze punten verdeling is een maximale score van 20 punten en een minimale score van -20 te halen.

Ondersteunende hulpmiddelen

In de categorie “ondersteunende middelen” zijn er verschillende tools/frameworks die op papier veel dezelfde dingen beloven. Dit kan dan ook een reden zijn waarom er geen tool/framework uitspringt wanneer naar het overzicht tabel wordt gekeken. In Tabel 6 is een overzicht te zien van de behaalde punten per tool/framework.

⁴ Met vendor lock-in is de afnemer afhankelijk van zijn leverancier. De leverancier levert een product die alleen bij hem kan worden afgenomen. Hierdoor is de afnemer niet in staat van leverancier te wisselen zolang gebruikt moet worden gemaakt van het product.

Multiplatform front-end development met tools/frameworks

Tool/framework	Punten
Ionic	15
Sencha Touch	14
Mobile Angular	15
Titanium	11
RhoMobile	6
MoSync	-2

Tabel 6 Ondersteunende hulpmiddelen score

Hoewel de top drie heel erg dicht bij elkaar zit, is vooral het resultaat van MoSync erg opvallend. Bij het analyseren van het framework bleek dat het gaat om een verouderd en verlaten framework dat in de laatste twee jaar geen product van een bedrijf is, maar puur leeft in de open source wereld met een kleine community. Al met al geen geschikte match voor Flusso.

De userinterface framework Ionic scoort het beste in de categorie “ondersteunende hulpmiddelen” en zal worden gebruikt om de lunch applicatie mee te realiseren.

Genererende hulpmiddelen

In de categorie “Genererende hulpmiddelen” komen de tools/frameworks minder goed uit de verf dan de ondersteunende hulpmiddelen. Zoals te zien in Tabel 7 vallen de resultaten daardoor ook wat lager uit.

Tool/framework	Punten
Intel XDK	13
Phone Gap/Cordova	12
QT	9
Genuitec	8
Supersonic	7
MobileTogether	2
Touchdevelop	-4

Tabel 7 Genererende hulpmiddelen score

Een opvallende naam in vergelijking met de rest is Phone Gap/Cordova. Waar de anderen zich vooral richten op het genereren van front-end code, is Phone Gap/Cordova bezig met het genereren van de software omgeving zodat de applicaties op verschillende besturingssystemen kan draaien. Dit betekent dat Phone Gap/Cordova nooit een optie is geweest om te gebruiken voor front-end development versnelling. Het is echter wel een tool die (meestal) draait op de achtergrond, aangezien het vaak gebruikt wordt in combinatie met front-end tools/frameworks om web-based code in een native webview te plaatsen.

Het framework Intel XDK scoort het beste in de categorie “genererende hulpmiddelen” en zal worden gebruikt om de lunch applicatie mee te realiseren.

Totale hulpmiddelen

In de categorie “totale hulpmiddelen” zitten de grote spelers die qua functionaliteiten veel meer te bieden hebben dan de standaard tools/frameworks. Wanneer de behaalde scores, zoals te zien in Tabel 8, worden vergeleken met Tabel 6 en Tabel 7, is het verschil echter niet zo groot.

Tool/framework	Punten
IBM	13*
OutSystems	12
Mendix	11
Telerik	8
Xamarin	6

Tabel 8 Totale hulpmiddelen score

*: Belangrijk om te vermelden is dat IBM geen punten heeft gekregen over de waardering van de prijs. Dit heeft geresulteerd in een (waarschijnlijk) positievere score dan dat IBM hoort te krijgen. De overige platformen hebben de meeste negatieve score gekregen wat betreft de prijs. Wanneer uit wordt gegaan van een soort gelijke beoordeling, zou IBM net als Mendix op een score van 11 uitkomen en is OutSystems het best scorende platform.

Hoewel de geteste platformen veel meer kunnen dan alleen mobile applicatie versnellen volgens de wensen van Flusso, is dit wel het aspect waarop de platformen geanalyseerd zijn. Alle extra beschikbare functionaliteiten die de platformen bieden, vallen buiten de scope en hebben daardoor geen invloed op de score. Dit betekent dat er voor de beschikbare functionaliteiten van een platform wordt betaald terwijl maar een klein deel (in deze scope) gebruikt wordt. Om deze reden zijn alle platformen veel duurder dan dat Flusso aangaf te willen betalen. Dit resulteert in het krijgen van -2 punten per platform voor de kosten, een andere reden waarom de score enigszins laag is.

Hoewel OutSystems het beste lijkt te passen bij Flusso, zijn er meerdere factoren die bepalen of een platform goed past. Een voorbeeld van één van die factoren is het commerciële team van een platform, er vanuit gaande dat een groter commercieel team meer producten verkoopt dan een kleiner team. Wanneer het platform vaker wordt verkocht, biedt dit meer potentiële klanten waar Flusso ondersteuning aan kan bieden. Dit kan betekenen dat OutSystems op technisch gebied het beste bij Flusso past, maar wanneer naar het totale plaatje gekeken word een ander platform beter is. Aangezien er in dit onderzoek puur gekeken wordt naar de front-end applicatie ontwikkeling, scoort OutSystems het beste en zal worden gebruikt voor de ontwikkeling van de lunch applicatie.

Een Enterprise account van OutSystems begint vanaf 1500 euro per maand, wat neer komt op ongeveer 375 euro per week. Om deze kosten terug te kunnen verdienen, moet er voor minimaal 375 euro aan tijdswinst te behalen vallen. Met een vraagprijs van 75 euro per uur per ontwikkelaar, moet een applicatie met OutSystems in 35 uur ontwikkeld zijn. Een hulpmiddel waar geen kosten aan verbonden zit mag er 40 uur over doen om beiden uit te komen op een prijs van 3000 euro. Om de prijs van OutSystems te kunnen rechtvaardigen, moet de applicatie ontwikkeling per week per ontwikkelaar vijf uur sneller gaan.

4.6 Bevindingen hulpmiddelen

Na de theoretische selectie (zie paragraaf 4.5) zijn drie hulpmiddelen geselecteerd die worden getest in de praktijk. Ionic, IntelXDK en OutSystems zijn ieder gebruikt voor de realisatie van de lunch applicatie. Per realisatie is gekeken naar de volgende punten:

Learning curve

Om de bruikbaarheid van de hulpmiddelen te kunnen meten, is de voortgang van de applicatie ontwikkeling en de learning curve van de programmeur in kaart gebracht. Per dag is de kennis en de voortgang beoordeeld met een cijfer nul tot en met tien. Wanneer de kennis met een tien is beoordeeld is alle kennis aanwezig voor de realisatie van de lunch applicatie. Wanneer de voortgang met een tien is beoordeeld is de lunch applicatie succesvol gerealiseerd.

Pluspunten / Minpunten

Aan de hand van de pluspunten wordt gekeken naar de meerwaarde van het hulpmiddel. Hoe wordt werk uit handen genomen van de ontwikkelaar, zodat er gemakkelijker ontwikkeld kan worden.

Aan de hand van de minpunten wordt in kaart gebracht waar de zwakte punten liggen van het hulpmiddel.

Resultaat

Het resultaat brengt in kaart in hoeverre het mogelijk is om met het hulpmiddel een *custom* (lunch) applicatie te maken. Hoe meer de gemaakte applicatie voldoet aan de gedefinieerde eisen in het technisch/functioneel ontwerp (zie Bijlage 4. Functioneel & Technisch ontwerp), hoe beter de resultaten.

Elk gebruikt hulpmiddel is beoordeeld zoals in voorbeeld Tabel 9. Per hulpmiddel wordt aangegeven of een benodigde functionaliteit voor de lunch applicatie (zoals Touch) ondersteunt werd of niet. Het ondersteunen van een functionaliteit wordt uitgedrukt in een "+", het niet ondersteunen wordt aangegeven met een "-". Tevens wordt aangegeven in hoeverre de hulpmiddel het mogelijk maakte om de schermen van de lunch applicatie te maken zoals ze staan aangegeven in het technisch/functioneel ontwerp. Waar 1 niet overeenkomt, is 10 een perfecte match.

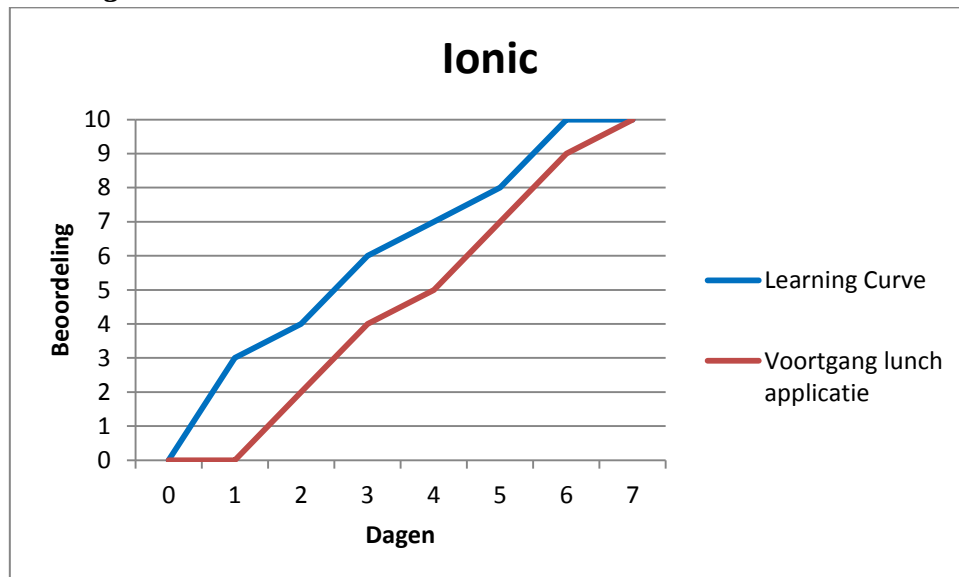
Functioneel	Resultaat
Touch aanraken	+
Swipe aanraken	-
OS detectie	+
Netwerk toegang	-
Netwerk informatie	+
Native look & feel	-
Lees/schrijf files	+
Communicatie REST-API	-
Schermen	1 t/m 10
Selecteer gebruiker	10
Lunch lijst dag	8
Lunch lijst week	6
Deelnemer opties	4
Deelnemer toevoegen	2

Tabel 9 Voorbeeld Resultaten tabel

4.6.1 Ionic

Het Ionic framework biedt ondersteuning door het toevoegen van beschikbare user interface functionaliteiten zoals knoppen, lists en andere web-based objecten. Wanneer de programmeur goed overweg kan met JavaScript, CSS en HTML is Ionic een goed hulpmiddel bij het maken van mooie, native uitziende applicaties. Wanneer de programmeur echter nog niet goed thuis is in JavaScript komt Ionic niet goed tot zijn recht.

4.6.1.1 Learning Curve



Figuur 6 Ionic Learning Curve

Voornamelijk de eerste dagen was de kennis van de door Ionic gebruikte JavaScript taal, Angular, laag. Dit resulteerde in een kleine kennis van het Ionic framework. De vooruitgang gaat echter snel wanneer de taal Angular beter wordt begrepen. Na enkele dagen wordt een dusdanige kennis van Angular gecreëerd waarmee soepel en snel met Ionic gewerkt kan worden. Wanneer de basis van Ionic met Angular is geleerd, gaat de learning curve een stuk langzamer en komt zelfs tot stilstand. Toen de learning curve tot stilstand kwam, was het leerproces met Ionic voor de realisatie van de lunch applicatie voltooid en de applicatie gerealiseerd.

4.6.1.2 Pluspunten

Opstarten

Nieuwe Ionic projecten worden aangemaakt via de command line interface(cli). Via de cli kunnen Cordova native functionaliteiten worden toegevoegd aan het project die vervolgens direct in het project te gebruiken zijn. Hierdoor is er volledige controle over de gewenste native functionaliteiten die de applicatie moet bevatten.

Testen

Het Ionic framework in combinatie met de Cordova tool zorgt ervoor dat er gemakkelijk gebouwd en getest kan worden. Door commando's in de cli in te voeren worden applicaties binnen seconden gebouwd en getest op een aangesloten mobiel, een simulator of via de webbrowser. Wanneer een applicatie gebouwd moet worden voor bijvoorbeeld een Android telefoon, moet de Android Studio (volledig) zijn geïnstalleerd. Hetzelfde geldt voor andere operating systems.

Design en realisatie

Ionic in combinatie met het Angular JavaScript zorgt voor een snelle design van de applicatie. Met het toevoegen van user interface objecten uit Ionic bestaan de (soms complexe) logica meteen, waarop de ontwikkelaar zelf acties kan uitvoeren. Objecten als een search bar op een lijst zijn gerealiseerd binnen minuten zonder zelf (complexe) code te hoeven schrijven, door simpelweg een Ionic lijst met een Ionic search bar toe te voegen. Vervolgens hoeft de lijst alleen maar gefilterd te worden met de waarden van de search bar.

Programmeeromgeving

Standaard werkt Ionic met AngularJS, maar de ontwikkelaar is vrij om een eigen/vertrouwd JavaScript framework toe te voegen. Hiermee wordt de ontwikkelaar niet beperkt in zijn mogelijkheden en hoeft er geen nieuw framework geleerd te worden. Dit betekent ook dat, wanneer bepaalde gewenste functionaliteiten worden vereist maar niet kunnen worden gerealiseerd met AngularJS, er simpelweg een JavaScript framework kan worden toegevoegd, die dat wel ondersteunt.

Write once's, Deploy everywhere

Waar Ionic met JavaScript frameworks, HTML en CSS zorgt voor een web applicatie, zorgt Cordova ervoor dat de web applicatie draait in een native container voor elke device waarop het wordt geïnstalleerd. Vanaf dat moment is het een hybride applicatie die, doormiddel van de Cordova plug-ins, toegang heeft tot de beschikbare native functionaliteiten van het apparaat.

Native look & feel

Door gebruik te maken van de Ionic user interface componenten, krijgt de applicatie vanzelf een native *look & feel*. Het Ionic framework houdt in de gaten op welk platform de applicatie gedraaid wordt en zal de daarbij behorende stijl aanpassen. Hier heeft de ontwikkelaar geen werk aan; dit wordt automatisch geregeld.

Voorbeelden

Ionic is een veel gebruikt en onderhouden framework. Dit zorgt voor een grote community met veel gebruikers die allemaal verschillende applicaties hebben gemaakt. De kennis van de ontwikkeling en de daarbij behorende voorbeeld code worden rijkelijk gedeeld. Hierdoor kan er gewerkt worden vanaf een kant-en-klare applicatie als basis en kunnen problemen vaak snel opgelost worden.

4.6.1.3 Minpunten

Programmeerkennis

Om Ionic te gebruiken voor het maken van web/hybride applicaties, is een minimale programmeerkennis vereist. Om Ionic goed tot zijn recht te laten komen, moet bekend zijn hoe JavaScript samenwerkt met HTML en CSS. Wanneer die basis aanwezig is, zal de ontwikkelaar zich moeten verdiepen in het Angular framework om optimaal gebruik te kunnen maken van Ionic.

Applicatie testen

Om een applicatie te kunnen testen moet de daarbij behorende programmeeromgeving geïnstalleerd zijn. Android Studio is voor elk operating system te installeren, waardoor Android applicaties gemakkelijk te testen zijn. De programmeeromgeving Xcode voor iOS kan alleen geïnstalleerd worden op een Mac. Hierdoor is het bouwen en testen van een applicatie voor een iOS apparaat alleen mogelijk op een Mac of een virtual box die draait als een Mac besturingssysteem.

Browserproblemen

Naast het testen van applicaties op echte mobieltjes is het ook mogelijk om de applicaties te testen in een webbrowser. Omdat Cordova bedoeld is voor de communicatie tussen JavaScript en de beschikbare native functionaliteiten van een mobiel, wordt Cordova niet ingeladen in een standaard browser. Wanneer vervolgens wordt geprobeerd gebruik te maken van JavaScript die een Cordova API aanroept, levert dit foutmeldingen op. Dit maakt het testen van de beschikbare native functionaliteiten via een browser niet mogelijk.

4.6.1.4 Resultaat

In Tabel 10 wordt weergegeven welke functionaliteiten worden ondersteund en zijn toegevoegd aan het Ionic project en hoeverre de applicatie overeen komt met het originele ontwerp.

Functioneel	Resultaat
Touch aanraken	+
Swipe aanraken	+
OS detectie	+
Netwerk toegang	+
Netwerk informatie	+
Native <i>look & feel</i>	+
Lees/schrijf files	+
Communicatie REST-API	+
Schermen	1 t/m 10
Selecteer gebruiker	10
Lunch lijst dag	9
Lunch lijst week	9
Deelnemer opties	10
Deelnemer toevoegen	10

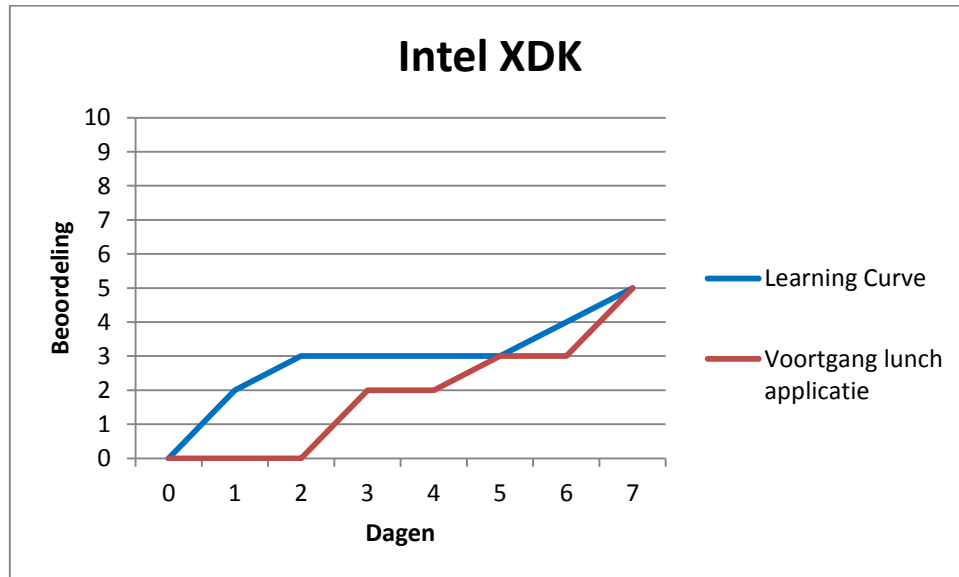
Tabel 10 Resultaten Ionic

Hoewel de realisatie van de schermen volledig is gelukt, wijken de kleuren en vormen van de “aanwezig” checkboxen af van het technisch ontwerp. Ionic past de vormen en kleuren van de checkboxen zo aan, dat ze per besturingssysteem, een native *look & feel* geven. Om deze reden hebben de schermen ‘Lunch lijst dag’ en ‘Lunch lijst week’ geen perfecte scores behaald. Afbeeldingen van de applicatie zijn te zien in Bijlage 7. Afbeeldingen Lunch applicatie Ionic.

4.6.2 Intel XDK

Intel XDK is een *drag and drop* framework die in combinatie met Cordova het mogelijk om hybride applicaties te genereren. Naast de *drag and drop* interface kan er ook gewerkt worden met code.

4.6.2.1 Learning Curve



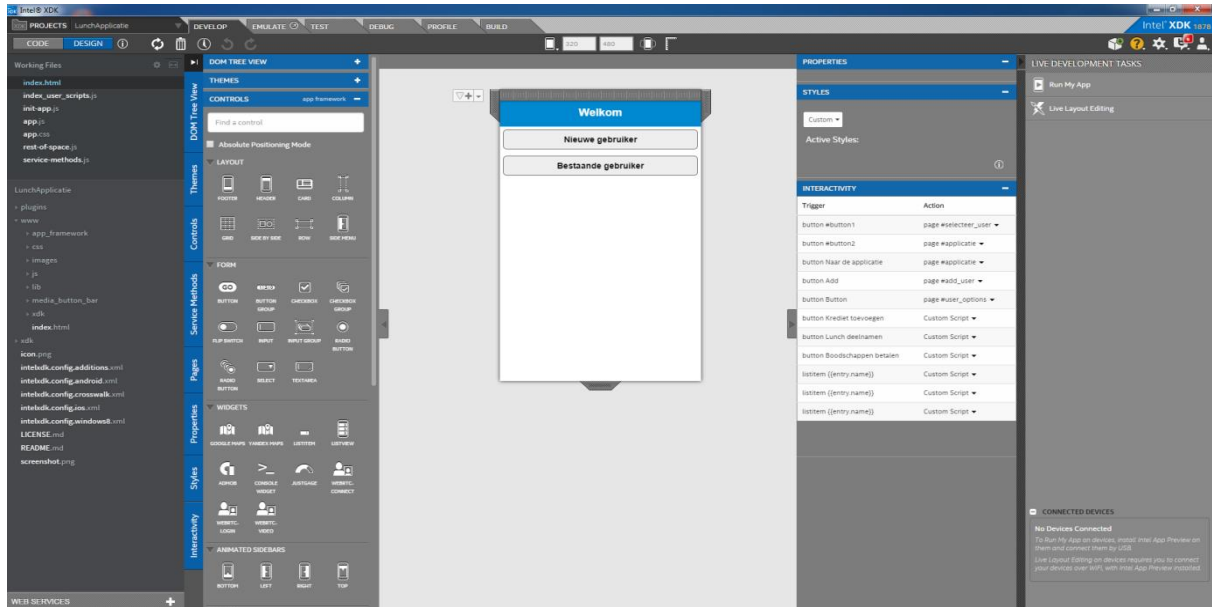
Figuur 7 Learning Curve Intel XDK

De learning curve zoals weergegeven in Figuur 7 toont een teleurstellende lijn. Waar na de eerste 2 dagen de verwachtingen hooggespannen waren, kwamen deze niet uit in de dagen die volgden. De eerste twee dagen werd er gewerkt met de tools en beschikbare *drag and drop* functionaliteiten die het framework te bieden had. Kleine testen werden uitgevoerd, van een simpele “Hello World” tot het uitvoeren van een native Cordova functionaliteit en het maken van een REST-API call. Het werd snel duidelijk hoe er met het framework gewerkt moest worden om een gewenst resultaat te behalen. Vanaf de derde dag kwam de realisatie van de lunch applicatie aan bod. Wanneer een custom applicatie gemaakt moet worden, die meer vereist dan de beschikbare functionaliteiten die het framework biedt, laat het framework het enorm afweten. De kennis die kon worden toegepast bij een standaard applicatie kwam te vervallen omdat deze niet kon worden toegepast in een custom applicatie. Hoewel het framework het toelaat om custom code toe te voegen, bleek het framework de custom code en de auto gegenereerde code van de *drag and drop* editor niet te kunnen combineren. Dit resulteerde in het stagneren van de learning curve en daardoor een lunch applicatie die niet volledig gerealiseerd kon worden.

4.6.2.2 Pluspunten

Drag and Drop

Met de Intel XDK App designer is het mogelijk om zonder een regel te programmeren een applicatie te maken. Dit wordt mogelijk gemaakt door de *drag and drop* functionaliteiten die de App designer beschikt. Hierdoor kan er snel en gemakkelijk een basis applicatie gemaakt worden zonder dat enige kennis van de onderliggende taal (HTML, CSS en JavaScript) nodig is. Een globale weergave van de app designer is te zien in Figuur 88.



Figuur 88 Intel XDK app designer

Web services

Het gebruik maken van web services resultaten, zoals een REST-API call, wordt gemakkelijk geïmplementeerd binnen de applicatie. Hoewel er wat code geschreven moet worden om een REST-API call te kunnen doen, creëert Intel XDK daarna zelf een data binding met de call. Hierdoor is de ontwikkelaar in staat deze data door de gehele applicatie heen te gebruiken met hetzelfde *drag and drop* principe.

Look & Feel

Met het Intel XDK framework zijn verschillende user interface frameworks te koppelen aan een applicatie. Dankzij de keuze uit verschillende UI frameworks is de ontwikkelaar in staat het framework te kiezen die het dichtst in de buurt komt van de wensen van de applicatie.

Native functionaliteiten

Wanneer een nieuw Intel XDK project wordt gestart, is de ontwikkelaar in staat om aan te geven welke gewenste native functionaliteiten nodig zijn binnen het project. Deze functionaliteiten kunnen op een later moment worden gewijzigd. De Cordova plug-ins worden niet alleen geaccepteerd, ze “werken” ook in de test omgeving. Hoewel er bijvoorbeeld geen echte camera wordt opgestart, wordt wel het beeld gegeven alsof een camera zou zijn opgestart. Het toevoegen van *third party* plug-ins wordt ook ondersteund, maar is niet altijd aan te raden. De software zal niet ondersteund worden door de mobiele simulator in de testomgeving en is daardoor alleen op echte apparaten te testen.

Multiplatform front-end development met tools/frameworks

Testen / bouwen

Met het Intel XDK framework wordt het testen van de (hybride) applicaties erg makkelijk gemaakt. De applicatie kan door middel van een mobiele simulator in de web browser worden getest, aan een met USB aangesloten apparaat of door gebruik te maken van de bestaande Enterprise app van Intel XDK. Deze applicatie logt in op het account van de ontwikkelaar en heeft vervolgens meteen toegang tot elke applicatie die de ontwikkelaar beheert. De app designer krijgt vervolgens informatie terug van de Enterprise applicatie, op deze manier kan er getest worden op een apparaat.

Omdat het bouwen van de applicatie gebeurt in de online (Cloud) servers van Intel XDK, hoeft er geen ontwikkelomgeving van Android of iOS op de computer aanwezig te zijn om een applicatie te kunnen maken.



Figuur 9 Intel XDK Simulator

4.6.2.3 Minpunten

Externe software

Hoewel het framework gemaakt is om zowel de *drag and drop* app designer te gebruiken als pure code te schrijven blijkt dit niet goed samen te werken. Wanneer wordt geprobeerd om gebruik te maken van externe JavaScript code, resulteert dit in foutmeldingen en kan het gebeuren dat de app designer vervolgens niet meer werkt. Toen externe Angular JavaScript aan het project wordt toegevoegd, begonnen beschikbare functionaliteiten van de app designer niet meer te werken. De externe Angular JavaScript kon niet worden toegevoegd zolang de app designer werd gebruikt.

Pagina flow

Om met de app designer van pagina te wisselen, kan bijvoorbeeld een knop met daarop een *onClick* actie worden geplaatst die verwijst naar een nieuwe pagina. Dit kan allemaal erg gemakkelijk gedaan worden met de app designer. Wanneer de applicatie zelf moet bepalen dat een nieuwe pagina kan worden ingeladen, is dit niet te doen. Het framework werkt met *Anchors*⁵ die tijdens het bouw proces worden gecreëerd. Hierdoor is dit met voor gedefinieerde code JavaScript niet te bereiken en is het niet mogelijk om zonder user input een pagina te veranderen, tenzij de standaard manier van Intel XDK wordt aangehouden.

⁵ Anchors zijn hyperlinks waarmee webpagina's naar elkaar kunnen verwijzen.

Designer/realiteit

Hoewel de app designer voor een groot deel “what you see is what you get” is, is dit niet in alle gevallen correct. Zo zijn er knoppen die in de designer de volledige breedte vullen, maar in werkelijkheid maar 25% van de breedte in beslag nemen. Daarnaast is het voorgekomen dat er verschil zit tussen de app designer, de mobiele simulatie en de applicatie op de mobiele telefoon.

Instabiel

De Intel XDK applicatie blijft regelmatig hangen of sluit zelfs helemaal af na het geven van een foutmelding. Wanneer dit gebeurt tijdens de ontwikkeling en er werk verloren gaat, levert het frustraties op.

Web services

Het gebruiken van de web service data laat te wensen over. Zo is de data alleen te gebruiken binnen bepaalde soort objecten. Wanneer een simpel tekstveld gevuld moet worden met data van de service call, gaat dit niet.

Oude (gegenereerde) code

Wanneer er met de app designer *drag and drop* acties worden uitgevoerd, wordt ook de bijbehorende JavaScript gegenereerd. Een actie is bijvoorbeeld het maken van een knop, die “Hello World” moet zeggen wanneer op de knop wordt gedrukt. Als deze knop vervolgens wordt weggehaald, verdwijnt deze wel uit de pagina, maar het gegenereerde JavaScript blijft bestaan.

4.6.2.4 Resultaat

In Tabel 11 wordt weergegeven welke functionaliteiten worden ondersteund en zijn toegevoegd aan het Intel XDK project en in hoeverre de applicatie overeenkomt met het originele ontwerp.

Functioneel	Resultaat
Touch aanraken	+
Swipe aanraken	-
OS detectie	+
Netwerk toegang	+
Netwerk informatie	-
Native <i>look & feel</i>	+
Lees/schrijf files	+
Communicatie REST-API	+
Schermen	1 t/m 10
Selecteer gebruiker	5
Lunch lijst dag	4
Lunch lijst week	0
Deelnemer opties	8
Deelnemer toevoegen	5

Tabel 11 Resultaten Intel XDK

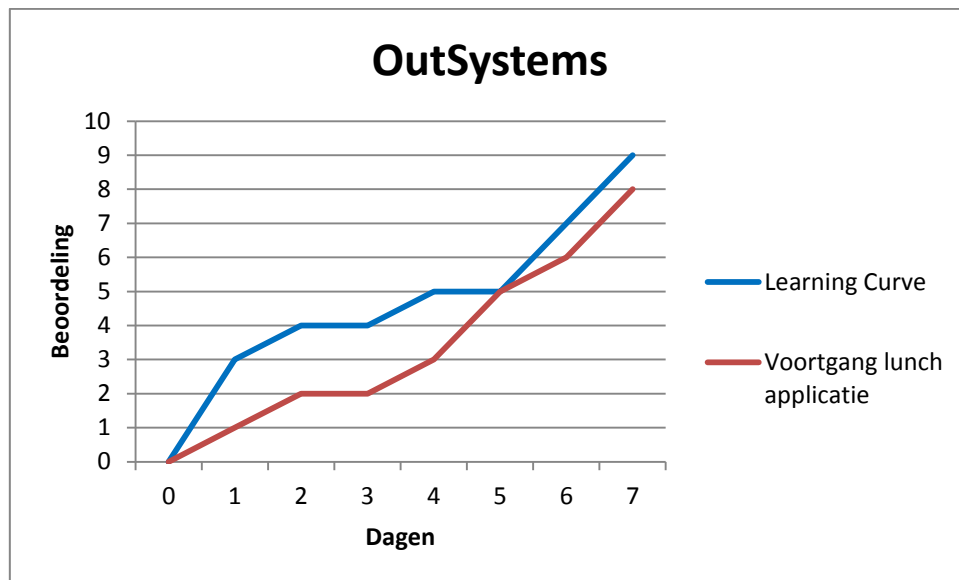
De beschikbare functionaliteiten van Intel XDK en de samenwerking met het Angular JavaScript konden er niet voor zorgen dat de lunch applicatie voldeed de eisen van het technisch ontwerp. Er zijn onnodige knoppen toegevoegd om te kunnen navigeren tussen de pagina's en de vorm van de applicatie komt niet overeen met het gewenste resultaat. Zonder de detectie van een groot toestel (tablet), was het ook niet mogelijk om de 'lunch lijst week' pagina te realiseren. Afbeeldingen van de applicatie zijn te zien in Bijlage 8. Afbeeldingen Lunch applicatie Intel XDK.

4.6.3 OutSystems

OutSystems is een platform dat door middel van *model driven development*, een *drag and drop* editor en het automatisch kunnen genereren van pagina's ondersteuning biedt aan het ontwikkel proces van applicaties.

4.6.3.1 Learning Curve

OutSystems heeft wisselvallige learning curve. Wanneer gekeken wordt naar het leerproces van de lunch applicatie met OutSystems, ontstaat een learning curve zoals te zien in Figuur 100.



Figuur 10 Learning Curve OutSystems

In de eerste twee dagen werd gewerkt met de beschikbare functionaliteiten die het platform te bieden heeft. Deze functionaliteiten zijn gemakkelijk toe te passen en op een begrijpbare manier. Hierdoor ontstaat al snel het idee dat er veel kennis van OutSystems aanwezig is. Wanneer er van de standaard manier wordt afgeweken, zal de ontwikkelaar een paar stapjes terug moeten doen en gaan uitzoeken waarom/hoe bepaalde beschikbare functionaliteiten werken zoals ze werken. Het achterhalen van deze informatie duurde van dag drie tot dag vijf, wat de reden is achter de lage groei van de learning curve. Met het blootleggen van de werking van het OutSystems platform werd kwam de kennis die nodig was om verder te gaan met de applicatie. Vanaf dag vijf ging de learning curve weer goed omhoog, en kon de applicatie verder worden ontwikkeld.

4.6.3.2 Pluspunten

Personal environment

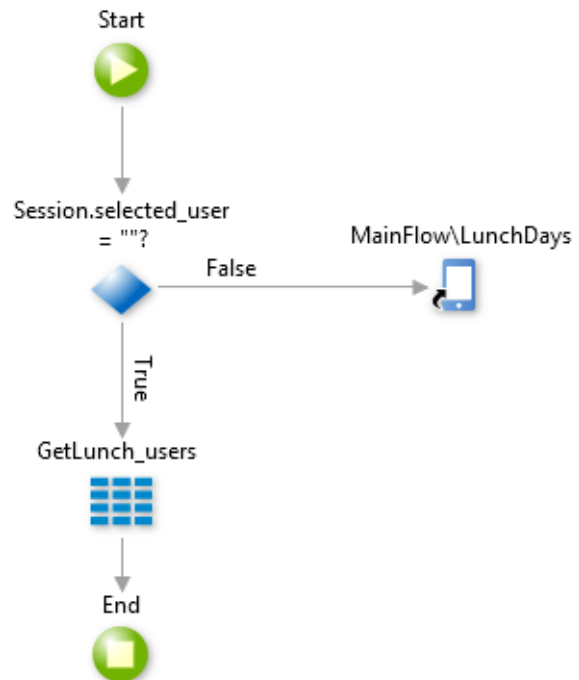
Om met OutSystems te kunnen werken moet een Cloud omgeving beschikbaar zijn waarop OutSystems kan draaien. Wanneer gebruik wordt gemaakt van de gratis OutSystems omgeving, is dit de openbare Cloud Server van OutSystems. In deze Cloud krijgt iedereen een persoonlijke omgeving, met daarin de applicaties van de ontwikkelaar(s). Met de persoonlijke omgevingen wordt het beheer en de controle van de applicaties gemakkelijk gemaakt.

Drag and drop

OutSystems is bedoeld om, zonder ook maar een regel code te schrijven, diverse applicaties te kunnen creëren. Om dit te doen maakt OutSystems onder andere gebruik van de *drag and drop* interface. Sleep simpelweg de benodigde informatie naar de juiste plek en OutSystems regelt de rest.

Model Driven Development

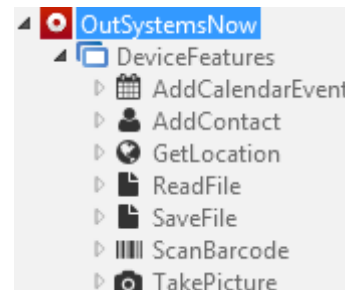
Naast de *drag and drop* interface werkt het OutSystems platform met *Model Driven Development* (MDD). In Figuur 11 is een MDD voorbeeld te zien. Wanneer de session variabele `selected_user` nog geen waarden kent, moeten de lunch users opgehaald worden zodat de gebruiker kan aangeven wie hij is. Wanneer de `selected_user` wel een waarde heeft gekregen, heeft de gebruiker zijn keuze gemaakt en kan het scherm "LunchDays" worden ingeladen.



Figuur 11 Model Driven Development

Native functies

Om native functionaliteiten op een telefoon te kunnen uitvoeren, heeft OutSystems een aparte module gemaakt: OutSystemsNow. Wanneer de OutSystemsNow module aan een project wordt toegevoegd, kunnen alle beschikbare native functionaliteiten binnen het project worden gebruikt. Een overzicht van de beschikbare functionaliteiten is te zien in Figuur 12.



Figuur 12 Native functies

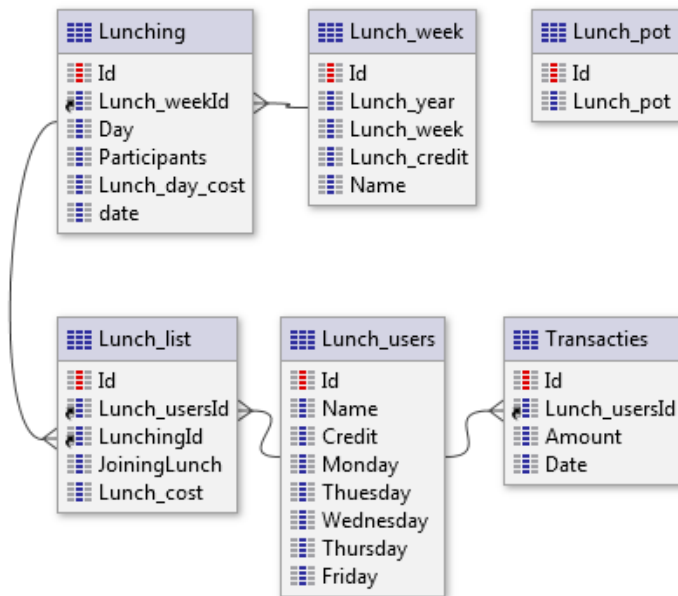
Om een module te kunnen gebruiken, moet deze binnen een scherm gesleept worden. Per module wordt duidelijk aangegeven welk input waarden nodig zijn, en welke return waarden terug zullen komen. Zowel de input waarden als de return waarden moeten zijn gedefinieerd voordat de module te gebruiken is.

Scaffolding schermen

Om het ontwikkel -en creëerproces van een applicatie zo gemakkelijk mogelijk te maken, kan OutSystems automatisch schermen genereren gebaseerd op entiteiten die zijn aangemaakt door de ontwikkelaar. Deze schermen bevatten *Create*, *Read*, *Update* en *Delete* (CRUD) functionaliteiten die kunnen worden uitgevoerd door de gebruiker van de applicatie. De schermen bevatten alle logica die nodig is om de benodigde informatie uit en naar de database te schrijven.

Database

Hoewel de ontwikkelaar binnen OutSystems met entiteiten werkt, verwerkt OutSystems deze naar database tabellen. Hierdoor is het gemakkelijk een database te maken zonder dat er echte database specifieke kennis nodig is. In Figuur 13 is een overzicht te zien van de database zoals die is gecreëerd binnen OutSystems.



Figuur 13 OutSystems Database

Door de database binnen OutSystems te gebruiken kunnen gegevens opgehaald worden uit de database zonder SQL kennis. De kant-en-klare SQL voor *Create*, *Read*, *Update* en *Delete* (CRUD) worden standaard gegenereerd wanneer entiteiten worden aangemaakt. Naast de kant-en-klare query's ondersteunt OutSystems het maken van custom (complexe) SQL query's.

Custom HTML en JavaScript

Het toevoegen van HTML kan op één enkele manier en gebeurt binnen de Service Studio. Door een "expression" te plaatsen met daarin standaard HTML content, zal deze worden weergegeven wanneer de pagina in een browser wordt geladen.

JavaScript kan op verschillende manieren worden opgeslagen en toegevoegd aan de applicatie. De eerste manier is per HTML pagina het JavaScript te plaatsen in de JavaScript editor van OutSystems. Dit JavaScript zal alleen beschikbaar zijn in de pagina waarvoor deze wordt ingeladen.

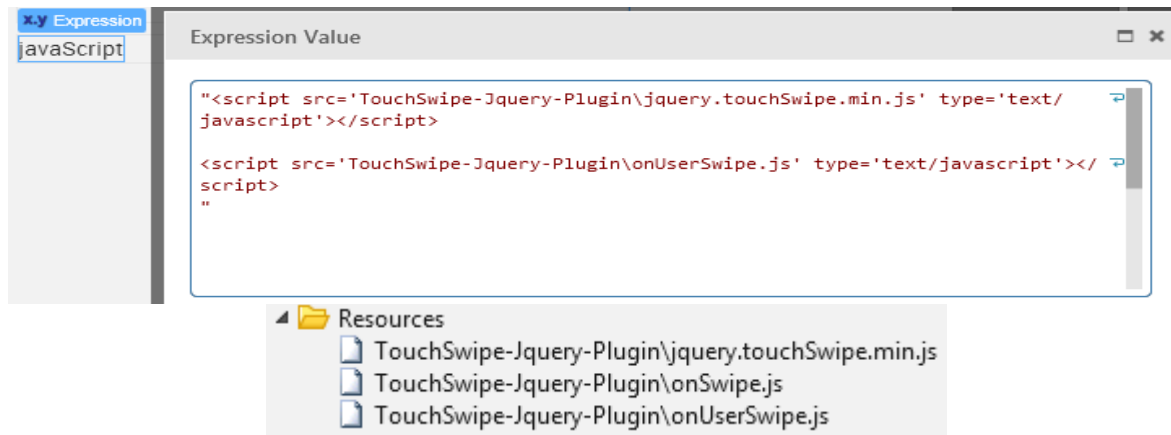


Figuur 14 OutSystems JavaScript

Multiplatform front-end development met tools/frameworks

De tweede manier is het schrijven van JavaScript in de JavaScript Editor van de gehele applicatie. Deze JavaScript zal in de hele applicatie aanwezig zijn en is altijd aan te roepen. Beide manieren zijn weergegeven in Figuur 14.

Het is ook mogelijk om kant-en-klare JavaScript bestanden toe te voegen aan een pagina. De JavaScript bestanden kunnen worden opgeslagen in een Resources mapje. Vandaar kunnen, via standaard HTML, de JavaScript bestanden worden ingeladen zoals te zien in Figuur 15.



Figuur 15 OutSystems JavaScript files inladen

Applicatie testen

Het testen van de ontwikkelde applicatie kan op twee verschillende manieren.

De eerste manier is binnen een browser. De applicatie staat gehost op de Cloud server van OutSystems, die te benaderen is via een URL. Wanneer de applicatie binnen een browser wordt getest, zullen de mobiele native functionaliteiten niet werken.

De tweede manier is door de OutSystemsNow Enterprise applicatie te downloaden op het apparaat dat de applicatie gaat testen. Via de applicatie is een connectie te maken met de Cloud ontwikkel server, waar alle gemaakte applicaties op staan. Vervolgens kan de gewenste applicatie gedraaid worden, die via de OutSystemsNow app in een native webview container gedraaid wordt.

4.6.3.3 Minpunten

Weinig native functies

Voor het maken van hybride applicaties zit OutSystems nog teveel in de kinderschoenen. De native functionaliteiten die worden ondersteund met het OutSystemsNow module zijn (in vergelijking met het aanbod dat Cordova biedt) schaars. Zo biedt OutSystems 7 verschillende native functionaliteiten, waar Cordova standaard begint met 19 beschikbare native functionaliteiten. Ook de acties die met de beschikbare native functionaliteiten van OutSystems zijn uit te voeren zijn minimaal.

Native look & feel

De beloofde native look & feel is in een kleine schaal aanwezig maar pakt niet in alle gevallen goed uit. Zo wordt het standaard submenu op een Android toestel goed getoond, maar werkt het submenu niet op een iPad.

JavaScript en widgets

Wanneer de ontwikkelaar JavaScript wil gebruiken voor het vullen van de content voor OutSystems widgets, ontstaat een probleem. Deze widgets vereisen namelijk een bepaalde Source informatie, bijvoorbeeld een call naar de database. In de call staat vervolgens de informatie die de widget moet gebruiken. Wanneer er rechtstreeks informatie uit JavaScript moet worden gebruikt, kan dit niet aangezien de JavaScript niet als Source kan worden gebruikt.

Aanpassingen

Binnen OutSystems wordt er gewerkt met gedefinieerde attributen en entiteiten. Wanneer deze worden aangemaakt, wordt daar een code bij gegenereerd. Echter, wanneer de attributen en entiteiten worden aangepast, verandert de auto-gegenereerde code niet altijd mee. Wanneer entiteiten nieuwe variabelen krijgen of nieuwe waarden kan dit conflicten geven met de bestaande gegevens. De verwijzingen in de code naar de (oude) entiteiten blijven bestaan en komen niet meer overeen met de nieuwe entiteit. Dit maakt het lastig om in een latere periode aanpassingen te maken aan core entiteiten die door de hele applicatie heen worden gebruikt .

Kosten

Om de kosten die verbonden zitten aan het OutSystems platform terug te verdienen, moest de lunch applicatie te realiseren zijn binnen 35 uur (zie paragraaf 4.5). Na 40 uur zat deze echter rond 80%, met een verwachte uitloop van een of twee dagen. De ontwikkelsnelheid zonder voorkennis van OutSystems gaat te langzaam om de kosten te kunnen waarborgen. Hoewel verwacht wordt dat de snelheid van realisatie met OutSystems omhoog zal gaan wanneer er meer ervaring is opgedaan, kan niet gezegd worden of de gevraagde winst van vijf uur per week gehaald zal worden.

4.6.3.4 Resultaat

In Tabel 12 wordt weergegeven welke functionaliteiten worden ondersteund en zijn toegevoegd aan het OutSystems project en hoeverre de applicatie overeen komt met het originele ontwerp.

Functioneel	Resultaat
Touch aanraken	+
Swipe aanraken	-
OS detectie	+
Netwerk toegang	+
Netwerk informatie	+
Native <i>look & feel</i>	+
Lees/schrijf files	+
Communicatie REST-API	+
Schermen	1 t/m 10
Selecteer gebruiker	8
Lunch lijst dag	10
Lunch lijst week	0
Deelnemer opties	10
Deelnemer toevoegen	10

Tabel 12 Resultaten OutSystems

Van de functionele aspecten van de lunch applicatie werd alleen swipe niet ondersteund. Hoewel deze functionaliteit via externe JavaScript libraries wel kan worden toegevoegd, zit het niet

standaard in het OutSystems pakket. Bij de resultaten van de gemaakte schermen wijken er twee af van het technisch ontwerp. Zo moest er bij het “Selecteer gebruiker” scherm een extra knop komen., die de schrijfactie naar de telefoon zou starten. Zonder deze knop kan de schrijffunctionaliteit niet worden aangeroepen. De pagina “Lunch lijst week” is niet gerealiseerd omdat OutSystems geen ontwikkeloverzicht gaf voor de tablet, of een ander groot apparaat. In plaats van het weekoverzicht op één pagina, is er met OutSystems een lunch dag/week zoek optie geïmplementeerd. Afbeeldingen van de applicatie zijn te zien in Bijlage 9. Afbeeldingen Lunch applicatie OutSystems.

4.7 Aansturen hulpmiddelen

Hoewel hulpmiddelen assisteren bij het realiseren van een applicatie, is de manier waarop dit gedaan wordt niet in elke situatie wenselijk of komt het resultaat niet overeen met de verwachting. Om de hulpmiddelen geschikt te maken voor verschillende situaties, is het belangrijk om te weten hoe er buiten het hulpmiddel om gewerkt kan worden.

4.7.1 Gegenerateerde lay-outs/schermen

De hulpmiddelen Intel XDK en OutSystems zijn allebei in staat om schermen te creëren/genereren. Wanneer de schermen niet overeenkomen met de specifieke wensen aan de applicatie, is het mogelijk om deze aan te passen door het schrijven van HTML, JavaScript en CSS. Hoewel het toevoegen van CSS bij Intel XDK voor de nodige problemen kan zorgen, is het OutSystems platform helemaal voorbereid op het *customizen* van een applicatie(zie paragraaf 4.6.3.2). Zo kan de CSS binnen Intel XDK in conflict raken met de standaard CSS die het framework met zich mee brengt, resulterend in style's die wegvallen/ inconsistent zijn.

Naast het toevoegen van HTML en CSS is het met de Service Studio van Outsystems mogelijk om gegenereerde lay-out's en schermen aan te passen binnen de *drag and drop* editor. Wanneer het scherm is gegenereerd kan deze worden aangepast tot het gewenste resultaat is behaald.

4.7.2 Native functies

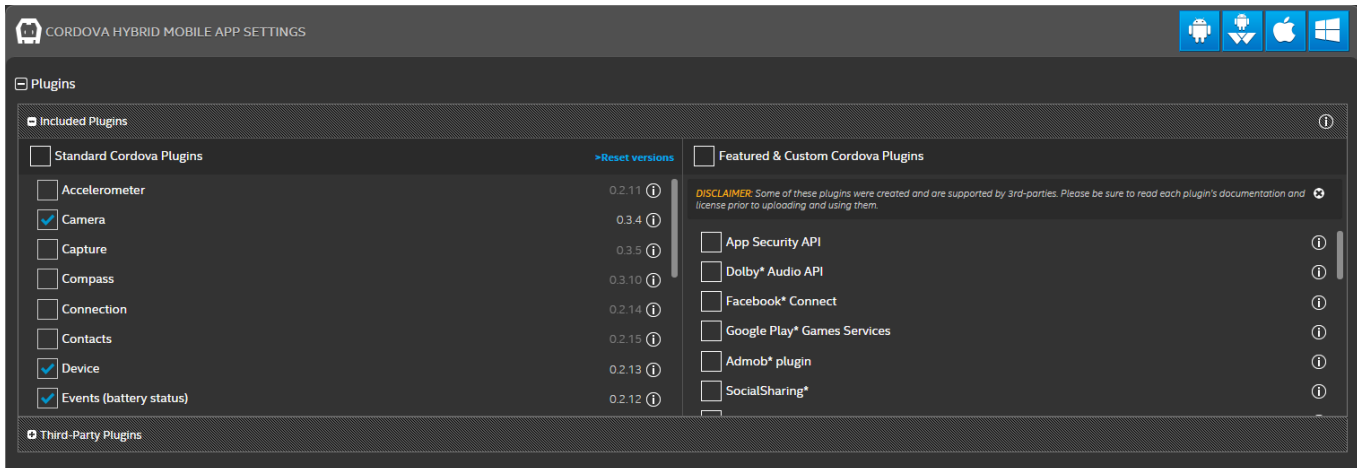
Om een hybride applicatie te kunnen maken, gebruiken alle drie de hulpmiddelen Cordova om vanuit JavaScript de beschikbare native functionaliteiten aan te kunnen roepen. Hoewel het toevoegen/gebruiken van de Cordova plug-ins per hulpmiddel net iets anders gaat, zijn ze wel allemaal compatible met elkaar en met andere hulpmiddelen die Cordova gebruiken.

Ionic

Ionic maakt gebruik van de standaard command line interface (cli) dat Cordova als een tool aanbiedt. Met de cli kunnen nieuwe Cordova projecten worden gemaakt, plug-ins worden beheerd en kunnen applicaties worden gebouwd/gestart. Via de cli kunnen plug-ins vanuit een register worden opgehaald en worden deze opgeslagen binnen het project. Een voorbeeld voor het toevoegen van de Cordova plug-in “console” is als volgt: “\$ cordova plugin add org.apache.cordova.console”. Wanneer de actie is uitgevoerd, wordt de plug-in code binnen het project geïnstalleerd en kan het meteen gebruikt worden binnen de applicatie.

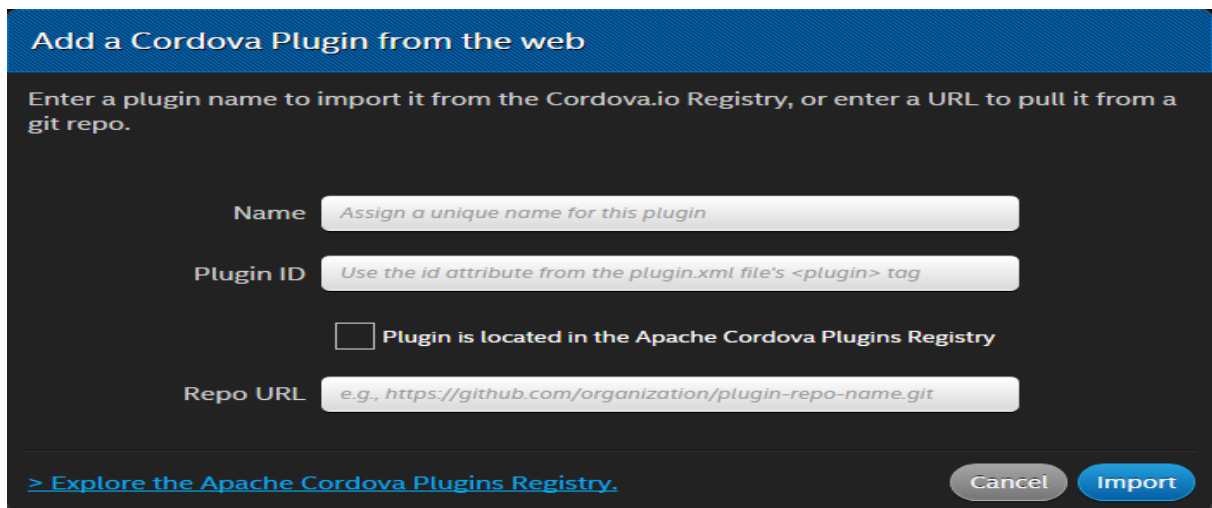
Intel XDK

Met Intel XDK wordt er bij het maken van elke nieuwe applicatie een lijst met plug-ins getoond. In deze lijst bevinden zich standaard Cordova plug-ins en *third party* plug-ins die worden ondersteund door Intel XDK. Er kan gekozen worden welke plug-ins benodigd zijn voor de applicatie zoals te zien op Figuur 16.



Figuur 16 Intel XDK Cordova Plug-ins

Naast de Cordova plug-ins ondersteunt Intel XDK het toevoegen van *third party* plug-ins zoals te zien op Figuur 17. Hiermee is het mogelijk om custom plug-ins te schrijven die kunnen worden toegevoegd aan de applicatie.



Figuur 17 Intel XDK Toevoegen van Plug-ins

OutSystems

Een OutSystems applicatie functioneert in twee onderdelen: de OutSystems platform kant en de OutSystemsNow applicatie kant. Om gewenste native functionaliteiten aan het platform toe te voegen, zullen de benodigde plug-ins aan het OutSystemsNow Cordova applicatie moeten worden toegevoegd. De OutSystemsNow applicatie zet de hele Cordova omgeving klaar en laadt vervolgens de echte applicatie binnen een webview in van het OutSystems platform.

Wanneer de applicatie vervolgens een Cordova-API wilt aanroepen, is dit nog niet mogelijk. De applicatie heeft nog geen kennis van het bestaan van Cordova. Om dit probleem op te lossen heeft OutSystems een CordovaLoader plug-in geschreven. De CordovaLoader zorgt ervoor dat wanneer er een applicatie wordt ingeladen binnen de OutSystemsNow applicatie, ook de daarbij horende Cordova JavaScript files worden klaargezet voor de applicatie.

5 Conclusies

Dit hoofdstuk presenteert de conclusies die aan de hand van de onderzochte deelvragen en het realiseren van de lunch applicatie te trekken zijn. Tevens wordt er antwoord gegeven op de hoofdvraag van dit onderzoek.

De hoofdvraag van dit onderzoek luidt: *“Op welke wijze kan aangetoond worden welke bestaande tools en/of frameworks het beste Flusso kan ondersteunen bij het realiseren van een multiplatform hybride mobile front-end applicatie?”*.

Om deze vraag te kunnen beantwoorden zijn de opgestelde deelvragen beantwoord. Met het uitwerken van deelvraag A *“Wat zijn de eisen die Flusso stelt aan een tool/framework?”* kwam aan de hand van een intern interview met Flusso medewerkers naar voren dat de tool/framework er voornamelijk voor moet zorgen dat een snelle *time to market* ontstaat. Door een snelle *time to market* kunnen eventuele kosten aan de tool/framework worden terugverdiend. Daarnaast moet de tool/framework beschikken over een gevalideerde techniek (met toekomst perspectief) die bekend is bij de medewerkers van Flusso. Voor de uitstraling van de applicatie is het wenselijk dat er per platform (Android, iOS, Windows) een native *look & feel* gecreëerd wordt.

Aan de hand van deelvraag B *“Op welke wijze past de lunch applicatie als referentie applicatie binnen het ontwikkelteam van Flusso?”* worden de gewenste functionaliteiten van de lunch applicatie vergeleken met de door Flusso gemaakte applicaties: Qc4u en Inspectandgo. In de vergelijking ontstond een duidelijk beeld waarin de lunch applicatie te weinig native functionaliteiten bezat om als referentie applicatie te dienen. Dit was een te verwachten risico met als gevolg dat er bij het testen van de tools/frameworks verder gekeken moest worden dan de realisatie van de lunch applicatie. Extra functionaliteiten zoals het gebruik van de camera, file transfer en de *geolocation* zullen beschikbaar moeten zijn.

Uit deelvraag C *“Welke eigenschappen van bestaande tools/frameworks maken deze geschikt om ingezet te worden tijdens de productie binnen het ontwikkelteam van Flusso?”* kwam een lijst van potentiële tools, frameworks en platforms naar voren die Flusso zouden kunnen ondersteunen bij het realiseren van hybride applicaties. Van de tools, frameworks en platforms kwamen eigenschappen naar voren zoals het genereren van code, het programmeren zo gemakkelijk mogelijk maken door middel van een *drag and drop* editor en ondersteuning leveren bij het maken van native *look & feel* web/hybride applicaties door user interface objecten. Daarnaast werd duidelijk dat de meeste tools/frameworks en platformen werken met de programmeertalen HTML en JavaScript.

Met het uitwerken van de deelvragen D *“Wat zijn de sterke en zwakke punten van elk(e) onderzocht(e) tool/framework voor Flusso?”* en E *“Welke top drie tools/frameworks zijn het meest geschikt voor de realisatie van applicaties voor Flusso?”* zijn de tools, frameworks en platformen verdeeld in drie groepen: ondersteunende hulpmiddelen, genererende hulpmiddelen en totale hulpmiddelen. Binnen de drie groepen zijn alle hulpmiddelen gevalideerd aan de hand van de wensen zoals geformuleerd in deelvraag A. Uit elke groep is het hulpmiddel dat als beste uit de test kwam geselecteerd voor realisatie van de lunch applicatie en verder onderzoek. De geselecteerde hulpmiddelen per groep zijn:

Ondersteunende hulpmiddelen – Ionic;

Genererende hulpmiddelen – Intel XDK;

Totale hulpmiddelen – OutSystems.

Multiplatform front-end development met tools/frameworks

Met deelvraag F “Op welke wijze ondersteunen de top drie tools/frameworks de realisatie van de lunch applicatie?” zijn de drie geselecteerde hulpmiddelen gebruikt om de lunch applicatie te realiseren. Per hulpmiddel kwamen zowel positieve als negatieve punten naar voren, waarop de hulpmiddelen zijn beoordeeld.

Ionic kreeg de beste beoordeling. In samenwerking met Cordova en AngularJS, was er met Ionic een native look & feel applicatie te ontwikkelen die vrijwel elk bestaande native functionaliteit kan beschikken. Hoewel Ionic snel en gemakkelijk te gebruiken is, is programmeerkennis vereist om een applicatie te kunnen creëren. Daarnaast moet er per gewenst platform de juiste software geïnstalleerd zijn om applicaties te kunnen bouwen/testen voor dat platform.

Het OutSystems platform is ideaal voor het creëren van web applicaties met weinig tot geen programmeerkennis. Dankzij het automatisch genereren van schermen en snel logica toevoegen door middel van *model driven development*, is het maken van applicaties een kwestie van dagen in plaats van weken. Hoewel applicatie ontwikkeling versneld wordt met OutSystems, levert dit geen winst op door de initiële kosten die aan OutSystems verbonden zitten. Daarnaast laat het platform wat steekjes vallen wanneer het gaat om de realisatie van hybride applicaties. Het platform biedt niet genoeg hybride functionaliteiten die nodig zijn bij een gemiddelde applicatie van Flusso. Zo worden gewenste functionaliteiten als netwerk informatie, swipe en het gebruik van file transfer niet ondersteund, die in Flusso applicaties wel gebruikt worden.

Intel XDK had de slechtste resultaten. Het schrijven van Angular JavaScript code botste met de automatische gegenereerde code van de app designer. Hierdoor werd het niet mogelijk om de tekortkomingen van Intel XDK te vervangen met custom Angular JavaScript, zonder de hele app designer weg te laten.

Uit de resultaten van het onderzoek naar deelvraag G “Op welke manier kunnen er functies toegevoegd worden aan het bestaande tool/framework?” blijkt dat alle drie de onderzochte hulpmiddelen gemakkelijk zijn aan te passen. Door het toevoegen van custom HTML, CSS en JavaScript zijn de resultaten van de hulpmiddelen te beïnvloeden. Alle drie de hulpmiddelen maken gebruik van Cordova om beschikbare native functionaliteiten vanuit JavaScript aan te kunnen roepen. Met Cordova kan het toevoegen van nieuwe gewenste native functionaliteiten snel, gemakkelijk en op een universele Cordova wijze.

Met de antwoorden op de deelvragen kan de hoofdvraag worden beantwoord. Door het valideren van de wensen en eisen binnen Flusso, was het mogelijk om een top drie aan hulpmiddelen samen te stellen die hieraan het beste voldoen. Om aan te tonen welk hulpmiddel er het beste past bij Flusso, werden de drie theoretisch beste hulpmiddelen in de praktijk getest door de realisatie van de lunch applicatie. De lunch applicatie gemaakt met Ionic kwam het meest overeen met het voorafgaand aan het onderzoek opgestelde technisch/functioneel ontwerp. Geconcludeerd kan worden dat het Ionic framework op het moment van schrijven het beste past bij Flusso voor ondersteuning bij het realiseren van hybride multiplatform applicaties.

6 Aanbeveling

Om Flusso te ondersteunen bij de realisatie van hybride multiplatform applicatie, wordt het user interface framework Ionic aanbevolen. Met Ionic zijn applicaties te vormen zoals de programmeur die wil, terwijl Ionic zorgt dat elk platform zijn eigen native *look & feel* krijgt. Daarnaast is het beheren en toevoegen van functionaliteiten per applicatie makkelijk gemaakt. De ontwikkelaar hoeft alleen aan te geven welke functionaliteiten aanwezig moeten zijn, Ionic zet deze vervolgens per platform klaar. Zo is Ionic snel, gemakkelijk en goed te gebruiken voor hybride applicaties.

Wanneer er puur gekeken wordt naar welk hulpmiddel het meeste werk uithanden neemt van de ontwikkelaar, komt OutSystems met de meeste voordelen. Met model driven development en het automatisch genereren van codes zijn applicaties zonder programmeerkennis snel te realiseren. Toch had OutSystems geen betere resultaten dan Ionic. OutSystems ondersteunt op het moment van schrijven nog te weinig native mobiele functionaliteiten om een Flusso applicatie mee te realiseren. Zo worden functionaliteiten als netwerkinformatie, swipe en het gebruik van file transfer niet ondersteund, die in Flusso applicaties wel gebruikt worden. Daarnaast is het met OutSystems niet gelukt om de applicatie binnen een week volledig volgens het Technisch document te realiseren. Met de kosten die OutSystems met zich mee brengt, moest de tijd van realisatie juist een krachtpunt van OutSystems zijn. Verwacht wordt dat de snelheid van realisatie omhoog gaat wanneer OutSystems vaker wordt gebruikt. Beschikbare functionaliteiten zoals *model driven development* leveren op langere termijn een tijdwinst, maar in het begin kost het tijd om de werking ervan goed te begrijpen.

Door de positieve ervaring die tijdens de ontwikkeling binnen OutSystems is opgedaan, wordt er aanbevolen om OutSystems in de gaten te houden en in de toekomst opnieuw te testen. Wanneer OutSystems een breder aanbod aan functionaliteiten biedt, en de kostenbesparing door het snel te kunnen ontwikkelen de kosten van OutSystems dekt, zal OutSystems de rol van Ionic kunnen overnemen als hulpmiddel voor de ontwikkeling van hybride multiplatform applicaties binnen Flusso.

7 Slotopmerkingen

Binnen dit onderzoek zijn er verschillende keuzes gemaakt die in dit hoofdstuk worden geëvalueerd.

Toevoegen platformen

Hoewel het onderzoek in eerste instantie alleen oog had voor tools en frameworks, is er een derde groep bij gekomen: platformen. Verschillende platformen boden functionaliteiten aan die goed overeen kwamen met de wensen vanuit Flusso. Omdat er een aanzienlijke kans was dat een platform een goede keuze voor Flusso kon zijn, zijn deze toegevoegd aan het onderzoek.

Lunch applicatie

Tijdens het onderzoek kwam naar voren dat de lunch applicatie niet genoeg functionaliteiten bezat om een goede referentie applicatie te zijn. Dit risico was van te voren verwacht en een oplossing stond paraat. Hoewel geen extra functionaliteiten zijn toegevoegd aan de lunch applicatie, zijn de onderzochte hulpmiddelen op extra functionaliteiten zoals *geolocation* en camera getest. Op deze manier werd gevalideerd dat de hulpmiddelen ook een bredere applicatie zouden kunnen ontwikkelen.

Resultaten

Dit onderzoek richt zich op front-end hybride applicatie ontwikkeling. Om dit te kunnen testen is de realisatie van de lunch applicatie gebruikt. De back-end van de lunch applicatie werkte nog niet volledig wat zou resulteren in niet volledig werkende applicaties. Om de werking van hulpmiddelen Ionic en Intel XDK toch te kunnen valideren, werd gebruik gemaakt van lokale test data. Ook zijn verschillende webservice calls gemaakt naar de al wel beschikbare REST-API om te valideren dat de hulpmiddelen om kunnen gaan met externe data. Met OutSystems is een lokale database gebruikt.

8 Bibliography

- Altova. (z.d.). *mobiletogether/develop-apps-fast*. Retrieved 02 27, 2015, from Altova: <http://www.altova.com/mobiletogether/develop-apps-fast.html>
- Appcelerator. (z.d.). *Titanium*. Retrieved Maart 2, 2015, from Appcelerator: <http://www.appcelerator.com/titanium/>
- Appgyer. (z.d.). *Supersonic*. Retrieved Maart 5, 2015, from Appgyer.
- Branas, R. (2014). *AngularJS Essentials*. Livery Place: Packt Publishing Ltd.
- Charland, A., & Leroux, B. (2011). Mobile application development: web vs. native. *Communications of the ACM*, 49-53.
- Christ, A. M. (2011). Bridging the Mobile App Gap. *Connectivity and the User Experience*(11(1),27).
- cordova.apache. (z.d.). *Home*. Retrieved Maart 11, 2015, from cordova.apache: <http://cordova.apache.org/>
- Dalmaso, I., Datta, S. K., Bonnet, C., & Nikaein, N. (2013). Survey, Comparison and Evaluation of Cross Platform Mobile Application Development Tools. *Wireless Communications and Mobile Computing Conference (IWCMC)*(9th International), 323-328.
- Danielsson, J. (2014). *Comparison study of cross-platform*. Mälardalen University.
- Falk, M., & Både, S. (z.d.). *Home*. Retrieved Februari 5, 2015, from Mobile Frameworks Comparison Chart: <http://mobile-frameworks-comparison-chart.com>
- Flusso. (z.d.). *Over ons*. Retrieved December 27, 2014, from Flusso: <http://www.flusso.nl/over-ons/>
- Genuitec: The Cloud Control Company. (n.d.). *myeclipse/mobile*. Retrieved Maart 3, 2015, from Genuitec: <https://www.genuitec.com/products/myeclipse/mobile/>
- Hadley, E. (2014, Oktober 21). *no-code-enterprise-mobile-app-development*. Retrieved Maart 2, 2015, from Mendix: <http://www.mendix.com/press/no-code-enterprise-mobile-app-development/>
- Hasselknippe, S. (2014). *Developing native applications on modern devices: A cross-platform approach*.
- Hogeschool Utrecht. (2014). *Afstudeerleidraad*. Utrecht.
- Horspool, N., & Tillmann, N. (2013). *TouchDevelop: Programming on the Go*.
- IBM Corporation. (2014). *An overview of IBM MobileFirst Platform*. United States of America: IBM Corporation.
- Ionicframework. (z.d.). *Ionic Documentation Overview*. Retrieved Maart 2, 2015, from Ionicframework: <http://ionicframework.com/docs/overview/#download>

Multiplatform front-end development met tools/frameworks

- Karadimce, A., & Bogatinoska, D. C. (2014). Using hybrid mobile applications for adaptive multimedia content delivery. *In Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, 686-691.
- Kim, H. J., Karunaratne, S., Regenbrecht, H., & Warren, I. (2015). *Evaluation of Cross-platform Development Tools for Patient Self-Reporting on Mobile Devices*.
- MobileAngularui. (z.d.). *home*. Retrieved Maart 4, 2015, from MobileAngularui: <http://mobileangularui.com/>
- OutSystems. (z.d.). *platform/madp*. Retrieved Maart 6, 2015, from OutSystems: <http://www.outsystems.com/platform/madp/>
- Outtier, B., & Leuven, B. (n.d.). *Vergelijkende studie van cross-platform tools voor de ontwikkeling van native mobiele*.
- Outtier, B., & Leuven, B. (z.d.). *Vergelijkende studie van cross-platform tools voor de ontwikkeling van native mobiele*.
- Palmieri, M., Singh, I., & Cicchetti, A. (2012). Comparison of cross-platform mobile development tools. *Intelligence in Next Generation Networks (ICIN)*, 179-186.
- Qt.io. (z.d.). *qt-Framework*. Retrieved Maart 3, 2015, from Qt.io: <http://www.qt.io/qt-framework/>
- ThoughtWorks. (n.b.). *Languages-and-frameworks*. Retrieved Februari 11, 2015, from thoughtworks: <http://www.thoughtworks.com/radar/languages-and-frameworks>
- ThoughtWorks. (z.d.). *Languages-and-frameworks*. Retrieved Februari 11, 2015, from thoughtworks: <http://www.thoughtworks.com/radar/languages-and-frameworks>
- Wasserman, A. I. (2010, November). Software engineering issues for mobile application development. *In Proceedings of the FSE/SDP workshop on Future of software engineering research*, 397-400.

9 Bijlagen

Bijlage 1. Plan van aanpak



Multiplatform front-end development met tools/frameworks

Bedrijfsbegeleider:

Dhr. Ir. L.C. van Mourik

E-mailadres:

l.vanmourik@Flusso.nl

Afstudeerbegeleider:

Dhr. Ir. H.B.A. Beerlage

E-mailadres:

Harry.beerlage@hu.nl

Student:	E-mailadres:	Opleiding	Student nummer:
Tom Oosterwijk	tom.oosterwijk@student.hu.nl	Technische Informatica	1608726

Datum: 24-02-2015

Periode: 02-02-2015 / 02-06-2015

Versie:	Datum:	Aanpassingen:
0.1	02-02-2015	Concept Versie
0.2	03-02-2015	Invulling + eerste spelling Controle
0.3	05-02-2015	PVA format wijziging formulier "Hoe stel ik PVA op"
0.4	06-02-2015	Theoretisch kader versterkt + toevoegen Afbakening
0.5	08-02-2015	Verwerking feedback meelezer
0.6	09-02-2015	Subvragen aangepast en feedback bedrijfsbegeleider toegepast
0.7	11-02-2015	Verwerking feedback docentbegeleider
0.8	18-02-2015	Verwerking feedback bootcamp
1.0	24-02-2015	Finale versie

Inhoud

1	Inleiding	50
2	Context	51
2.1	Flusso	51
2.2	De bedrijfsbegeleiding.....	51
2.3	Het afstudeerproject	51
2.4	Functie en verantwoordelijkheden	51
2.5	Communicatie	52
2.6	Stakeholders	52
3	De (advies/ontwerp)opdracht.....	53
3.1	De kwestie	53
3.2	Doelstelling.....	54
3.3	Omschrijving opdracht	54
3.4	Onderzoeksvragen.....	55
3.5	Afbakening.....	55
4	Projectactiviteiten	56
4.1	Methoden en technieken.....	57
4.2	Fasering/planning.....	59
4.2.1	Fasering	59
4.2.2	Planning	60
5	Producten	61
6	Risico's	62
7	Projectorganisatie	63
8	Bibliography.....	64
9	Bijlage Onderzoek schema	66

1 Inleiding

In het kader van het afstuderen wordt er van elke student aan de Hogeschool Utrecht verwacht dat er een afstudeeropdracht uitgevoerd wordt. Het doel van de afstudeeropdracht is om de competities van de student te testen aan de hand van een scriptie en een uit te voeren onderzoek. Echter, voordat de student kan beginnen met zijn onderzoek binnen de afstudeeropdracht, dient er een plan van aanpak gerealiseerd te worden. Dit document heeft als doel om duidelijkheid over de afstudeeropdracht te creëren voor zowel de bedrijfsbegeleider, de docentbegeleider als de student zelf.

Dit document gaat over de afstudeeropdracht “Multiplatform front-end development met tools/frameworks” bij het bedrijf Flusso te Leusden, die uitgevoerd zal worden door de student Tom Oosterwijk. In het plan van aanpak komt naar voren welke werkzaamheden er verwacht worden van de student, hoe deze werkzaamheden tot stand zullen komen en wat er aan het eind van de afstudeerperiode moet worden opgeleverd.

In Hoofdstuk 2 wordt de context van de opdracht geschetst. Hierin wordt duidelijk waar de afstudeerstage zal plaatsvinden, wie de bedrijfsbegeleider is en wat de rol van de student is binnen het bedrijf.

Hoofdstuk 3 gaat over de opdracht die de student zal gaan uitvoeren. Naast de kwestie zoals deze is opgesteld door het bedrijf wordt er door de student beschreven welke onderzoeks- of deelvragen er onderzocht gaan worden en wat de uiteindelijke doelstelling van het onderzoek is.

In Hoofdstuk 4 gaat de student dieper in op het onderzoeksplan om antwoord te kunnen geven op de onderzoeks- en deelvragen. In het onderzoeksplan komt naar voren welke activiteiten de student zal moeten uitvoeren, welke methodes per deelvraag gebruikt gaan worden, welke middelen nodig zijn om dit tot stand te brengen en welke verwachte resultaten dit zal opleveren.

Hoofdstuk 5 gaat over de producten die tijdens de afstudeerperiode gemaakt zullen worden en de deelproducten die daarvoor nodig zijn.

In Hoofdstuk 6 worden de risico's van het onderzoek beschreven en hoe er met de risico's wordt omgegaan indien deze tijdens het onderzoek naar voren komen.

In Hoofdstuk 7 wordt samengevat welke personen betrokken zijn binnen het afstudeer project en welke rol zij daarin vervullen.

De bronnen en referenties die zijn gebruikt bij het maken van het plan van aanpak staan verwerkt in Hoofdstuk 8.

2 Context

In dit hoofdstuk zal er achtergrondinformatie gegeven worden over het bedrijf waar de afstudeerstage plaatsvindt, de bedrijfsbegeleiding van uit Flusso, het projectkader en de positie van de student in het bedrijf.

2.1 Flusso

Het afstudeerproject zal plaats vinden bij het bedrijf Flusso. Flusso is een maatwerk applicatieontwikkelaar en ICT dienstverlener die meedenkt en adviseert over de architectuur van applicaties. Flusso ontwerpt, ontwikkelt en verbetert maatwerkapplicaties en zorgt door middel van integratieoplossingen voor een optimale samenwerking tussen IT oplossingen (Flusso, z.d).

Het hoofdkantoor van Flusso is gevestigd in Leusden en een tweede kantoor staat in Groningen. Over de twee kantoren zijn 50 werknemers verdeeld. De afstudeerstage zal plaatshebben in Leusden, waar de stagiair valt onder de afdeling "Java". De afdeling wordt vooral ingezet bij het realiseren van maatwerk software voor relatief kleine bedrijven met relatief kleine budgetten.

2.2 De bedrijfsbegeleiding

Tijdens de afstudeerperiode zal de student worden begeleid door Lars van Mourik. Hij heeft na zijn studie Elektrotechniek aan de Universiteit Twente eerst een aantal jaren gewerkt op het snijvlak van hardware en software met vooral embedded systemen. Deze systemen waren veelal slechts een klein onderdeel van het grotere geheel, dus heeft hij zich erg veel bezig gehouden met de integratie van deze systemen. Vervolgens is de overstap gemaakt naar "pure" software ontwikkeling bij Flusso. Hij is erg breed inzetbaar en houdt zich bezig met vrijwel alle facetten van de materie: van de infrastructuur, servers, integratie met andere systemen tot en met het ontwerpen en ontwikkelen van de applicaties zelf, voornamelijk in Java. De contact gegevens van Lars zijn te zien in Hoofdstuk 7 onder het kopje Begeleider Flusso.

2.3 Het afstudeerproject

Het afstudeerproject begint op 02-02-2015 en loopt tot en met 02-06-2015. In deze periode wordt er voor de student een plek binnen Flusso aangewezen waar hij 8 uur per dag kan werken. De student is vrij in het indelen van zijn begin- en eindtijd zolang de vastgestelde uren aan het eind van de maand zijn gemaakt. Met het indelen van de tijd moet de student zelf rekening houden met de openingstijd (08:00) en sluitingstijd (18:00) van Flusso.

Het afstudeerproject is een losstaand project dat geen deel uitmaakt van een ander/groter project. De resultaten van het afstudeerproject kunnen echter wel de basis zijn voor toekomstige opdrachten wanneer een snel/gemakkelijk te gebruiken tool/framework wordt gevonden. Het tool/framework zou dan gebruikt worden voor andere doeleinden dan alleen dit project.

2.4 Functie en verantwoordelijkheden

De positie van de student is die van ontwikkelaar++ (de ++ is omdat de student ook het ontwerp en de architectuurfase meemaakt) die tijdens zijn stageperiode verantwoordelijk is voor het uitvoeren van zijn onderzoek en het realiseren van een daarbij behorende proof of concept. Daarnaast draait de student mee in het ontwikkelaarsteam binnen Flusso.

2.5 Communicatie

Tijdens het afstuderen zal de student geregeld moeten communiceren met de bedrijfsbegeleider Lars van Mourik. Aangezien de student en de begeleider naast elkaar zitten tijdens de werkuren (indien de begeleider niet bij de klant op locatie is) kan op ieder moment vragen gesteld worden, indien nodig. Om de projectvoortgang te waarborgen zal er ook minimaal één keer per week een gesprek gehouden worden over de uitgevoerde werkzaamheden en de aankomende planning. Hierdoor zijn beide partijen op de hoogte van de voortgang van de student en wordt de begeleider in staat gesteld input te leveren op de toekomstige werkzaamheden.

2.6 Stakeholders

Binnen deze afstudeeropdracht zijn er twee groepen stakeholders die belang hebben bij het onderzoek. De eerste groep zijn de software ontwikkelaars binnen Flusso die aan de hand van het resultaat gebruik kunnen gaan maken van de onderzochte tool/framework.

Een tweede groep is de sales/management team binnen Flusso. Wanneer de applicatie ontwikkeling versneld kan worden hoeft hierin minder tijd geïnvesteerd te worden en worden dus minder kosten gemaakt.

3 De (advies/ontwerp)opdracht

In dit hoofdstuk komt naar voren wat de opdracht is, waarom de opdracht tot stand is gekomen, wat de doel van de student is binnen de opdracht en wordt de opdracht verdeeld in verschillende deelvragen die samen het antwoord op de opdracht moeten geven.

3.1 De kwestie

Flusso bouwt regelmatig mobiele applicaties voor klanten en doet dit met de bouwstenen Cordova/PhoneGap, AngularJS en de gebruikelijke *Integrated Development Environment* (IDE) ontwikkel software. Binnen de ontwikkeling van mobiele applicaties zijn er drie onderscheidingen te maken. Er zijn native⁶, web based⁷ en hybride⁸ applicaties (Charland & Leroux, 2011). Voor de native en webbased applicaties beschikt Flusso over ontwerp/ontwikkel tools/frameworks die het realiseren van applicaties helpt te versnellen. Voor de hybride applicaties worden geen specifieke ontwerp tools of frameworks gebruikt die het ontwerp of de ontwikkeling versnellen.

Flusso merkt echter dat er in de markt wel tools/frameworks beschikbaar zijn en deze claimen het realiseren van hybride mobiele front-end applicaties gemakkelijk(er) en snel(ler) te maken. Dit zou voor Flusso kunnen betekenen dat er minder tijd en dus geld geïnvesteerd hoeft te worden in het maken van de hybride multiplatform applicaties.

Flusso wilt een onderzoek naar deze tools/frameworks waarbij gezocht moet worden naar een tool/framework die past bij de (technische) wensen van Flusso. Daarnaast wil Flusso dat de gevonden tools/frameworks worden gevalideerd met het realiseren van een proof of concept. De resultaten van het onderzoek & de uitvoering van het proof of concept zullen samen de basis leggen aan het uiteindelijke advies rapport waarin staat welke tool/framework het beste past bij Flusso.

Recentelijk is er bij Flusso het idee ontstaan om een zogeheten “lunch app” te maken. Bij Flusso is er elke dag een bedrijfslunch waar iedereen individueel geld inlegt en een medewerker hiermee inkopen doet. Vanwege de flexibele bezetting zijn er elke dag andere behoeften en omdat is gebleken dat veel mensen geen contant geld op zak hebben, is de lunch pot meestal niet goed genoeg gevuld om boodschappen van te doen. Hierdoor ontstaan allerlei praktische problemen en doordat het saldo niet goed wordt bijgehouden kan het zijn dat sommige medewerkers (veel) te veel en sommige (veel) te weinig geld inleggen. Door de lunch pot in een applicatie te administreren verwacht Flusso dit praktische probleem te tackelen.

Flusso wil een technisch/functioneel ontwerp van de lunch app die tijdens het realiseren van de proof of concepts aangehouden moet worden. Om er zeker van zijn dat de resultaten van de proof of concepts relevant zijn voor Flusso moet er onderzocht worden of de lunch app wel een goed referentie kader is.

Wanneer blijkt dat de tools/frameworks een aanwinst voor het bedrijf kunnen zijn, zullen deze worden geïmplementeerd bij het realiseren van toekomstige applicaties.

⁶ Wordt specifiek voor een bepaald soort apparaat gebouwd.

⁷ Wordt gebouwd om ondersteunt te worden door browsers.

⁸ Combinatie tussen native en web based

3.2 Doelstelling

Het doel van dit project is het opleveren van een advies aan Flusso te Leusden op 02-06-2015 over het gebruik van bestaande tools/frameworks om de front-end realisatie van hybride multi platform applicaties te versnellen en het opleveren van de lunch applicatie die gerealiseerd is door drie verschillende tools/frameworks die het beste bij Flusso passen.

3.3 Omschrijving opdracht

De opdracht die gerealiseerd dient te worden bestaat uit twee onderdelen. Het eerste deel is het onderzoek naar de bestaande multiplatform tools/frameworks die het realiseren van mobiele front-end applicaties ondersteunen. Daarnaast wordt de bruikbaarheid van de geselecteerde tool(s)/framework(s) gevalideerd met het realiseren van de “lunch app”.

De lunch app bestaat uit de volgende componenten:

- Java backend met REST interfaces (en onderliggende database)
- Web/mobile front-end

Om de web/mobile front-end te kunnen realiseren zal er gebruik worden gemaakt van een tool/framework. De bestaande tools/frameworks worden gebruikt om code eenmalig te schrijven en vervolgens op meerdere platforms te kunnen gebruiken (Outtier & Leuven, Vergelijkende studie van cross-platform tools voor de ontwikkeling van native mobiele). Wanneer dit gebeurt is er sprake van multiplatform development. Het gebruik van multiplatform development stelt de maker in staat om op kortere termijn een applicatie te realiseren die meteen op meerdere mobiele platformen werkt, in plaats van voor elke platform een custom applicatie te realiseren.

Om te voorkomen dat er onderzoek wordt gedaan naar elke willekeurige tool/framework die er bestaat, wordt er een grove selectie gemaakt. Deze selectie wordt gemaakt aan de hand van wensen/voorkeuren (de platform, de programmeertaal, de target en hardware features) waar de tool/framework aan moet voldoen (Falk & Både, z.d.).

Bovendien is er al onderzoek gedaan naar mobiele multiplatform tools/frameworks waar er vooral gekeken werd naar de mobiele operating system, de kosten van de tool, de programmeertaal, de mogelijkheden van de API naar de hardware componenten, de architectuur en de IDE omgeving (Palmieri, Singh, & Cicchetti, 2012). Deze resultaten zullen worden meegenomen in het onderzoek, maar er wordt rekening gehouden met het feit dat deze data al achterhaald kan zijn aangezien de resultaten al ruim 2 jaar oud zijn.

Van de drie tools/frameworks die als het meest optimaal uit het onderzoek naar voren komen moet een aparte proof of concept (de lunchapp) gemaakt worden. Met het realiseren van de proof of concept moet er bijgehouden worden hoe het proces is verlopen. De resultaten worden met elkaar vergeleken om een advies rapport op te stellen.

Inhoudelijke trefwoorden: mobiel Java back-end, mobiel front-end tools/frameworks, technische/functioneel ontwerp, advies & ontwerp, hybride mobiel applicatie.

3.4 Onderzoeksvragen

Bij het maken van de afstudeeropdracht naar mobiele tools/frameworks is de volgende centrale onderzoeksvraag geformuleerd:

- Op welke wijze kan aangetoond worden welke bestaande tools en/of frameworks het beste Flusso kan ondersteunen bij het realiseren van een multiplatform hybride mobile front-end applicatie?

Om antwoord te kunnen geven op de onderzoeksvraag zijn er verschillende deelvragen geformuleerd:

- Wat zijn de eisen die Flusso stelt aan een tool/framework?
- Op welke wijze past de lunch app als referentie applicatie binnen het ontwikkelteam van Flusso?
- Welke eigenschappen van bestaande tools/frameworks maken deze geschikt om ingezet te worden tijdens de productie binnen het ontwikkelteam van Flusso?
- Wat zijn de sterke en zwakke punten van elk(e) onderzocht(e) tool/framework voor Flusso?
- Welke top drie tools/frameworks zijn het meest geschikt voor de realisatie van applicaties voor Flusso?
- Op welke wijze ondersteunen de top drie tools/framework de realisatie van de lunch app?
- Op welke wijze kunnen er functies toegevoegd worden aan het bestaande tool/framework?

De bijlage **Onderzoek schema** bevat een schematische weergave van de onderzoeksvraag, de deelvragen en de subdeelvragen. De uitgewerkte subdeelvragen zullen antwoord geven op de deelvragen, die vervolgens gezamenlijk weer antwoord zullen geven op de centrale onderzoeksvraag.

3.5 Afbakening

Gedurende het project zal een top drie van tools/frameworks samengesteld worden die voldoen aan de opgestelde eisen binnen de organisatie van Flusso met betrekking tot front-end development. Met de top drie tools/frameworks zal de lunch app proof of concept gemaakt worden en de bevindingen van het gebruik van elke tool/framework worden genoteerd. De uitkomst van de vergelijking tussen de bevindingen zal resulteren in een adviesrapport dat antwoord geeft op de onderzoeksvraag.

4 Projectactiviteiten

Tijdens het afstuderen zullen verschillende projectactiviteiten worden verricht.

Wekelijkse bespreking met de bedrijfsbegeleider Lars van Mourik

Om de voortgang van het project te bewaken zal er minimaal één keer per week *face to face* contact zijn met Lars van Mourik. Tijdens deze contacturen worden de resultaten van de voorgaande week besproken en wordt gekeken naar de planning voor de aankomende week. Daarnaast zal er een nieuwe bespreking zijn wanneer een (deel)product wordt opgeleverd om deze te reviewen.

Besprekingen met de afstudeerbegeleider Harry Beerlage

Vanuit de Hogeschool Utrecht is Harry Beerlage aangewezen om de afstudeerbegeleider te zijn. Harry Beerlage heeft tijdens het onderzoek de rol als begeleider en bij de afstudeerzitting zal hij de tweede examinerator zijn. Er zal maandelijks een *face to face* contact moment plaats vinden en indien de student het nodig lijkt om extra besprekingen in te plannen kan dit in overleg. Het tot stand komen van afspraken zal geschieden door telefonisch contact, Skype of per e-mail.

Realiseren functioneel/technische ontwerp

Voordat gekeken kan worden naar eventuele tools en frameworks zal er eerste een duidelijk beeld moeten ontstaan over de lunch app die gerealiseerd moet worden. Dit zal worden gedaan door middel van een functioneel / technische ontwerp.

Onderzoeksvraag/deelvragen beantwoorden

Om antwoord te kunnen geven op de doelstelling van Flusso zal er een onderzoek gedaan worden naar de kwestie. Om het project realiseerbaar te houden is het onderzoek opgesplitst in zeven deelvragen die tezamen antwoord zullen geven op de onderzoeksvraag.

Realiseren van de proof of concept applicatie

Om de tools/frameworks die door middel van het literatuuronderzoek worden geïdentificeerd, te valideren, zullen de tools/frameworks in de praktijk getest worden. Dit wordt gedaan met de realisatie van de lunch app. Voor elke tool/framework uit de samengestelde top drie zal er een proof of concept gemaakt worden. De bevindingen en resultaten uit de proof of concept in combinatie met de bevindingen uit de andere deelvragen zullen uiteindelijk antwoord geven op de onderzoeksvraag.

Scriptie schrijven

De inhoud van de scriptie bevat het gehele werkproces van de student. De scriptie zal geschreven worden in het Nederlands en dient op 02-06-2015 te worden ingeleverd. De scriptie moet voldoen aan de criteria die zijn opgesteld in bijlage 6.3, 6.4 en 6.5 van de afstudeerleidraad (Hogeschool Utrecht, 2014).

Advies rapport schrijven

In het advies rapport komen de bevindingen van het onderzoek en de realisatie van de proof of concepten samen. Dit rapport dient er voor om Flusso inzicht te geven over het gebruik van de tools/frameworks en of beantwoordt de centrale onderzoek hoofdvraag.

Maken van de eindpresentatie:

De afstudeerperiode wordt afgesloten met een 20 minuten durende presentatie. In de presentatie worden de bevindingen van de student gepresenteerd en toegelicht. Na de presentatie zal de student in discussie gaan over zijn afstudeerproject met de eerste en tweede examinerator.

4.1 Methoden en technieken

In dit hoofdstuk wordt beschreven welke methodes en technieken er gebruikt gaan worden om per deelvraag antwoord te kunnen geven op de onderzoeksvraag.

Wat zijn de eisen die Flusso stelt aan een tool/framework?

- Onderzoek functie:
 - o Ontwerpend, de resultaten van dit onderzoek zullen worden gebruik gebruikt om in een later stadium de onderzochte tools/frameworks te valideren aan de wensen van Flusso.
- Onderzoek eenheid:
 - o Kwalitatief onderzoek, door middel van een gesloten interview. Het interview zal bij werknemers van verschillende afdelingen worden afgenomen (management, programmeurs, sales (in verband met wensen klanten)).

Op welke wijze past de lunch app als referentie applicatie binnen het ontwikkelteam van Flusso?

- Onderzoek functie:
 - o Vergelijkend, de lunchapp zal moeten worden vergeleken met eerder gemaakte applicaties van Flusso.
- Onderzoek eenheid:
 - o Kwalitatief onderzoek, door middel van een literatuur onderzoek en ontwerp. Er zal worden gekeken naar de gebruikte technieken binnen de gemaakte applicaties en de overeenkomsten die er zijn met de lunchapp. Voor de lunch app zal een functioneel/technisch ontwerp worden gemaakt om een goed beeld te krijgen van de benodigde technieken voor de realisatie van de lunch app.

Welke eigenschappen van bestaande tools/frameworks maken deze geschikt om ingezet te worden tijdens de productie binnen het ontwikkelteam van Flusso?

- Onderzoek functie:
 - o Verkennend, het selecteren van de tools/frameworks die in aanmerking komen binnen Flusso.
- Onderzoek eenheid:
 - o Kwalitatief onderzoek, door middel van een literatuuronderzoek. Om een selectie van tools/frameworks te kunnen maken zal er gebruik gemaakt worden van een bestaande wizard die frameworks valideert (Falk & Både, z.d.).

Een andere bron die gebruikt zal worden is een wizard van ThoughtsWorks. ThoughtsWorks houdt bestaande/gebruikte en aankomende tools/frameworks in de gaten en laat zien welke tools/frameworks er aan te raden zijn en welke minder goed uit de verf komen (ThoughtWorks, n.b.). Naast de twee websites die beschikken over informatie van nieuwe/huidige tools/frameworks wordt ook gekeken naar eerder onderzochte en gevalideerde tools/frameworks (Palmieri, Singh, & Cicchetti, 2012).

Het hierboven geschreven onderzoek komt uit 2012 en kan op sommige punten accuraat zijn maar het kan niet meer als 100% betrouwbaar gezien worden.

Multiplatform front-end development met tools/frameworks

Wat zijn de sterke en zwakke punten van elk(e) onderzocht(e) tool/framework voor Flusso?

- Onderzoek functie:
 - o Vergelijkend, er wordt een vergelijking gemaakt binnen de gevonden tools/frameworks.
- Onderzoek eenheid:
 - o Kwalitatief onderzoek, door middel van een literatuuronderzoek. Er zullen bronnen gebruikt worden die informatie bevatten over de frameworks die aan Apple en Android platformen ondersteuning bieden.

Welke top drie tool/framework zijn het meest geschikt voor de realisatie van applicaties binnen Flusso?

- Onderzoek functie:
 - o Vergelijkend, de onderzochte tools/frameworks worden tegenover de wensen van de lunch app gezet.
- Onderzoek eenheid:
 - o Kwalitatief onderzoek, de gemaakte analyse van de zwakte/sterkte punten van elke tool/framework zal worden vergeleken met de eisen aan de lunch app. De drie tools/frameworks die het beste uit de vergelijking komen zullen worden meegenomen voor verder onderzoek en realisatie van de proof of concept.

Op welke wijze ondersteunen de top drie tools/frameworks de realisatie van de lunch app?

- Onderzoek functie:
 - o uitvoerend, de tools/frameworks worden gebruikt om de lunch app te maken.
- Onderzoek eenheid:
 - o Kwalitatief onderzoek, de bevindingen tijdens het realiseren worden bijgehouden en verwerkt in een advies rapport.

Op welke manier kunnen er functies toegevoegd worden aan het bestaande tool/framework?

- Onderzoek functie:
 - o Ontwerpend, er moet worden vast gelegd hoe er buiten de tool/framework kan worden gewerkt.
- Onderzoek eenheid:
 - o Kwalitatief onderzoek, door middel van een literatuuronderzoek en uitvoerend onderzoek. Om de deelvraag te kunnen beantwoorden moet er antwoord gegeven worden op de volgende drie criteria:
 - Op welke manier kunnen gegenereerde lay-outs/schermen worden aangepast buiten het framework om?
 - Op welke manier kunnen er native functies van mobiele telefoons worden toegevoegd aan de *tool/framework*?
 - Is de manier waarop de native functies worden toegevoegd *compatible* met andere frameworks?

4.2 Fasering/planning

Om het overzicht te behouden is het project opgedeeld in verschillende fases en is er een globale planning gemaakt.

4.2.1 Fasering

De fases:

- Project oriëntatie
 - Plan van aanpak
- Onderzoek
 - Beantwoorden deelvragen
 - Proof of concept functioneel/technische ontwerp
- Proof of concept
 - Realiseren
- Eindfase
 - Scriptie
 - Adviesrapport
 - Presentatie

Project oriëntatie

In de project oriëntatie zal de student kennis maken met het bedrijf, de opdracht duidelijk krijgen en deze verwerken in een plan van aanpak. De deadline voor het plan van aanpak is 20-03-2015.

Onderzoek

Tijdens het onderzoek zal de student antwoord geven op deelvragen, voor zover dat mogelijk is. De laatste deelvraag is afhankelijk van de resultaten uit de proof of concept realisatie en zal tijdens de proof of concept fase worden beantwoord.

Proof of Concept

Tijdens de proof of concept fase zullen er maximaal drie proof of concept worden gerealiseerd. Het aantal proof of concept dat zal worden gemaakt hangt af van het aantal gevonden tools/frameworks die goed zijn in te zetten voor de lunch app. De bevindingen van de realisatie van het proof of concept zullen worden verwerkt in de laatste deelvraag.

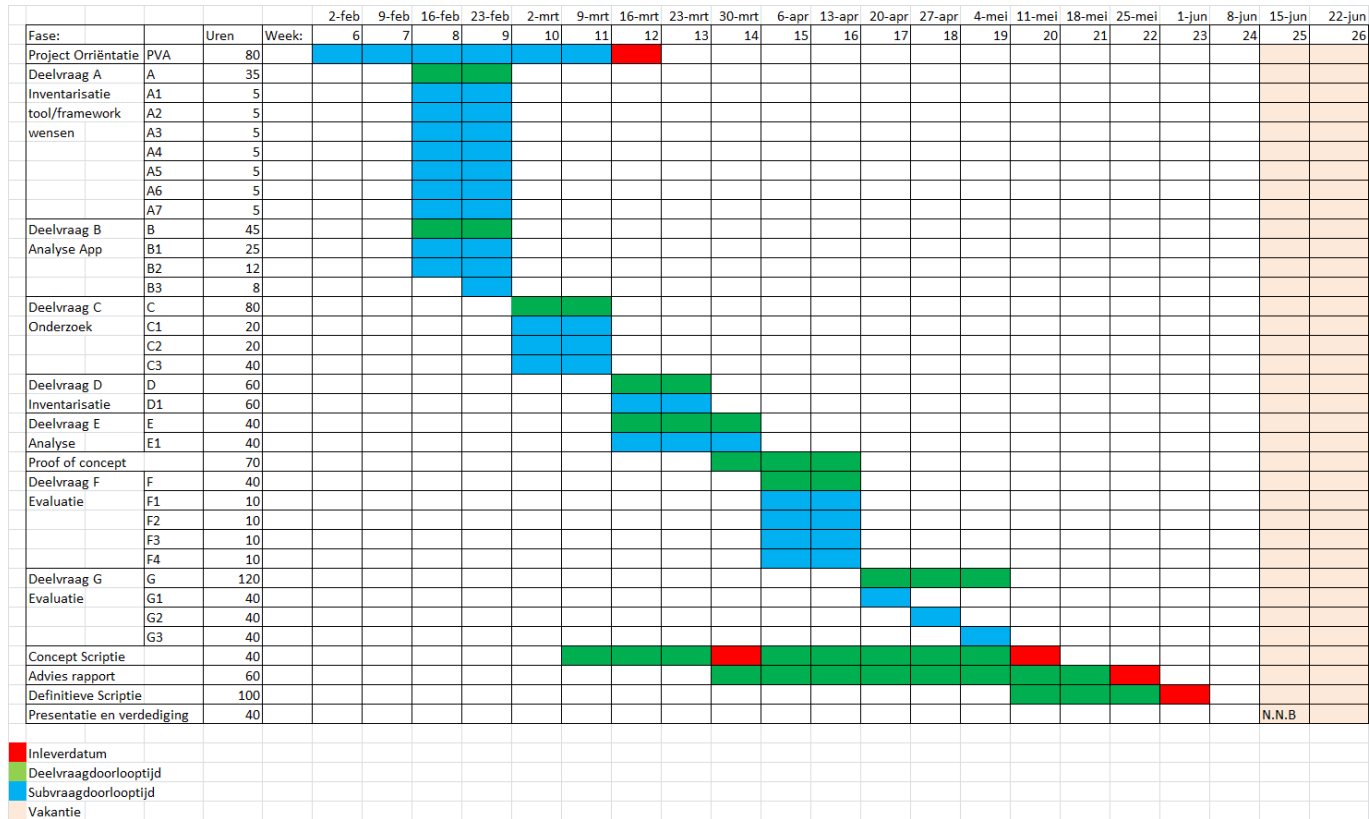
Eindfase

In de eindfase zullen de resultaten van het onderzoek en de proof of concept worden samen gevoegd tot een adviesrapport. Het adviesrapport zal uiteindelijk dienen om Flusso een advies te geven over welke tools/frameworks er goed te gebruiken zijn binnen de organisatie en welke de student hen aanraadt.

In de scriptie wordt de hele stageperiode uitvoerig omschreven, wordt het onderzoek uitgelegd en de resultaten toegelicht.

4.2.2 Planning

Om een goed beeld te krijgen van wat er allemaal in het project gedaan moet worden is de planning in figuur 1 weergegeven.



Figuur 18 De onderzoeksplanning

Hoewel de hele planning voor het project belangrijk is, zijn er vier data die apart vermeld worden. Namelijk, de inleverdata van de producten: plan van aanpak(20-03), concept scriptie 1(20-04), concept scriptie 2 (11-05) en de definitieve scriptie (02-06). In de planning van figuur 1 zijn al deze data met rood gearceerd. Het werk dat van feedback zal worden voorzien, zal op de maandag van de week waarop het materiaal dient te worden ingeleverd aan de begeleider worden toegezonden.

5 Producten

In dit hoofdstuk komt naar voren welke producten er aan het einde van de stage moeten worden opgeleverd.

- De scriptie
- Het adviesrapport
- Proof of concept(s)

De Scriptie

In de scriptie worden alle bevindingen van het onderzoek en de opdracht genoteerd en dient als verantwoording voor het tot stand komen van het uiteindelijke product.

Het adviesrapport

Het adviesrapport heeft een informerende functie over de tools/frameworks die zijn gebruikt voor het realiseren van de proof of concept(s).

Proof of concept(s)

De lunch applicatie die per tool/framework is gerealiseerd met daarbij een beschrijving over de resultaten van de lunch applicatie.

Om deze drie producten te kunnen realiseren zijn de er deelproducten toegevoegd.

- De scriptie
 - Interview Flusso werknemers
 - Rapport resultaten uit literatuur/veld onderzoek
- Het adviesrapport
 - Proof of concept bevindingen
- Proof of concept(s)
 - Functioneel/technisch ontwerp lunch app
 - Tool/framework analyseren op front-end development

6 Risico's

Tijdens het afstudeerproject kunnen er verschillende problemen ontstaan die de planning in de war kunnen brengen. In tabel 2 presenteert een overzicht van eventuele risico's.

Risico	Kans	Impact	Gevolg	Maatregel
De gevonden tools/frameworks zijn (te) duur om aan te schaffen.	10%	8	De andere deelvragen en het realiseren van een proof of concept kunnen niet tot stand komen zonder een tool/framework.	De eisen waaraan de tool/framework moet voldoen verlagen zodat goedkopere tool(s)/framework(s) in aanmerking komen.
Het realiseren van het rapport over de native functies duurt langer dan gepland.	50%	3	Er ontstaat geen goed beeld van de top 3 frameworks. Het zal onduidelijk blijven of het framework ondersteuning biedt om functies toe te voegen.	Indien blijkt dat er te weinig tijd is om het hele verslag te realiseren, zullen er per framework minder criteria gevalideerd worden. Zo zal er in grote tijdsnood alleen het eerste criterium worden gevalideerd.
De lunch app blijkt geen goede referentie applicatie te zijn voor Flusso.	20%	2	De lunch app kan niet gebruikt worden bij het testen van de frameworks omdat het geen goed beeld zou geven over hoe de frameworks Flusso kunnen helpen bij de ontwikkeling van applicaties door Flusso.	De ontbrekende onderdelen die de lunch app wel een goed referentie applicatie zouden maken worden meegenomen bij het onderzoeken naar de tools/frameworks, maar zullen niet geïmplementeerd worden in de applicatie zelf.

Tabel 2: Risico analyse

Risico:	Omschrijving van het risico
Kans:	De kans uitgedrukt in percentage over de mogelijkheid dat dit risico optreedt
Impact:	De impact die dit risico zal hebben op het project, waar 1 laag is en 10 hoog
Gevolg:	Omschrijving van het gevolg als het risico dat beschreven staat, uitkomt
Maatregel:	Omschrijving om het risico te minimaliseren

7 Projectorganisatie

Tijdens het afstuderen zijn de volgende personen betrokken bij het proces.

Student

Naam: T. Oosterwijk
E-mail: tom.oosterwijk@student.hu.nl
Telefoon: 06-10 018 230
Adres: Varenstraat 81
Plaats: 3765 WK Soest

Bedrijf

Naam: Flusso B.V.
E-mail: info@flusso.nl
Telefoon: +31 (0)33 – 4347 680
Adres: Hamersveldseweg 126
Plaats: 3833 GV Leusden

Begeleider Flusso

Naam: Dhr. Ir. L.C. van Mourik
Functie: Consultant
E-mail: l.vanmourik@flusso.nl
Telefoon: 06-23 432 032
Adres: Nieuweweg 28
Plaats: Soest

Afstudeerbegeleider Hogeschool Utrecht

Naam: Dhr. Ir. H.B.A. Beerlage
E-mail: harry.beerlage@hu.nl
Telefoon: (088) 484 8054
Adres: Neijenoord 1, Kamer D01.22
Plaats: Utrecht

Gesloten interview groep

Naam: Werknemers van Flusso
E-mail: iedereen@flusso.nl
Telefoon: +31 (0)33 – 4347 680
Adres: Hamersveldseweg 126
Plaats: 3833 GV Leusden

8 Bibliography

- Altova. (z.d.). *mobiletogether/develop-apps-fast*. Opgeroepen op 02 27, 2015, van Altova: <http://www.altova.com/mobiletogether/develop-apps-fast.html>
- Appcelerator. (z.d.). *Titanium*. Opgeroepen op Maart 2, 2015, van Appcelerator: <http://www.appcelerator.com/titanium/>
- Appgyer. (z.d.). *Supersonic*. Opgeroepen op Maart 5, 2015, van Appgyer.
- Branas, R. (2014). *AngularJS Essentials*. Livery Place: Packt Publishing Ltd.
- Charland, A., & Leroux, B. (2011). Mobile application development: web vs. native. *Communications of the ACM*, 49-53.
- Christ, A. M. (2011). Bridging the Mobile App Gap. *Connectivity and the User Experience*(11(1),27).
- cordova.apache. (z.d.). *Home*. Opgeroepen op Maart 11, 2015, van cordova.apache: <http://cordova.apache.org/>
- Dalmasso, I., Datta, S. K., Bonnet, C., & Nikaein, N. (2013). Survey, Comparison and Evaluation of Cross Platform Mobile Application Development Tools. *Wireless Communications and Mobile Computing Conference (IWCMC)*(9th International), 323-328.
- Danielsson, J. (2014). *Comparison study of cross-platform*. Mälardalen University.
- Falk, M., & Både, S. (z.d.). *Home*. Opgeroepen op Februari 5, 2015, van Mobile Frameworks Comparison Chart: <http://mobile-frameworks-comparison-chart.com>
- Flusso. (z.d.). *Over ons*. Opgeroepen op December 27, 2014, van Flusso: <http://www.flusso.nl/over-ons/>
- Genuitec: The Cloud Control Company. (sd). *myeclipse/mobile*. Opgeroepen op Maart 3, 2015, van Genuitec: <https://www.genuitec.com/products/myeclipse/mobile/>
- Hadley, E. (2014, Oktober 21). *no-code-enterprise-mobile-app-development*. Opgeroepen op Maart 2, 2015, van Mendix: <http://www.mendix.com/press/no-code-enterprise-mobile-app-development/>
- Hasselknippe, S. (2014). *Developing native applications on modern devices: A cross-platform approach*.
- Hogeschool Utrecht. (2014). *Afstudeerleidraad*. Utrecht.
- Horspool, N., & Tillmann, N. (2013). *TouchDevelop: Programming on the Go*.
- IBM Corporation. (2014). *An overview of IBM MobileFirst Platform*. United States of America: IBM Corporation.
- Ionicframework. (z.d.). *Ionic Documentation Overview*. Opgeroepen op Maart 2, 2015, van Ionicframework: <http://ionicframework.com/docs/overview/#download>

Multiplatform front-end development met tools/frameworks

- Karadimce, A., & Bogatinoska, D. C. (2014). Using hybrid mobile applications for adaptive multimedia content delivery. *In Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, 686-691.
- Kim, H. J., Karunaratne, S., Regenbrecht, H., & Warren, I. (2015). *Evaluation of Cross-platform Development Tools for Patient Self-Reporting on Mobile Devices*.
- MobileAngularui. (z.d.). *home*. Opgeroepen op Maart 4, 2015, van MobileAngularui: <http://mobileangularui.com/>
- OutSystems. (z.d.). *platform/madp*. Opgeroepen op Maart 6, 2015, van OutSystems: <http://www.outsystems.com/platform/madp/>
- Outtier, B., & Leuven, B. (sd). *Vergelijkende studie van cross-platform tools voor de ontwikkeling van native mobiele*.
- Outtier, B., & Leuven, B. (z.d.). *Vergelijkende studie van cross-platform tools voor de ontwikkeling van native mobiele*.
- Palmieri, M., Singh, I., & Cicchetti, A. (2012). Comparison of cross-platform mobile development tools. *Intelligence in Next Generation Networks (ICIN)*, 179-186.
- Qt.io. (z.d.). *qt-Framework*. Opgeroepen op Maart 3, 2015, van Qt.io: <http://www.qt.io/qt-framework/>
- ThoughtWorks. (n.b.). *Languages-and-frameworks*. Opgeroepen op Februari 11, 2015, van thoughtworks: <http://www.thoughtworks.com/radar/languages-and-frameworks>
- ThoughtWorks. (z.d.). *Languages-and-frameworks*. Opgeroepen op Februari 11, 2015, van thoughtworks: <http://www.thoughtworks.com/radar/languages-and-frameworks>
- Wasserman, A. I. (2010, November). Software engineering issues for mobile application development. *In Proceedings of the FSE/SDP workshop on Future of software engineering research*, 397-400.

9 Bijlage Onderzoek schema

Onderzoeksvraag:			
“Op welke wijze kan aangetoond worden welke bestaande tools en/of frameworks het beste Flusso kan ondersteunen bij het realiseren van een multiplatform hybride mobile front-end applicatie?”			
	Deelvraag	Nr.	Subvraag
A	Wat zijn de eisen die Flusso stelt aan een tool/framework?	A1	Welk platform moet de tool/framework ondersteunen naast Apple en Android?
		A2	Wat is de jaarlijkse kosten die de licentie van de tool/framework mag zijn?
		A3	Welke technieken moet de tool/framework ondersteunen?
		A4	Welke programmeertalen moet de tool/framework ondersteunen?
		A5	Welke soort typen applicaties moet de tool/framework ondersteuning aan leveren naast hybride?
		A6	Welke native hardware functies moet de tool/framework kunnen ondersteunen?
		A7	Welke user interface moet de tool/framework opleveren?
B	Op welke wijze past de lunch app als referentie applicatie binnen het ontwikkelteam van Flusso?	B1	Welke technieken/functies zijn er bij Flusso voorheen gebruikt?
		B2	Welke technieken/functies zijn er nodig voor de realisatie van de lunchapp?
		B3	Wat zijn de overeenkomsten tussen de gemaakte applicaties en de lunchapp?
C	Welke eigenschappen van bestaande tools/frameworks maken deze geschikt om ingezet te worden tijdens de productie binnen het ontwikkelteam van Flusso?	C1	Welke tool/framework bied ondersteuning voor multiplatform(minimaal Apple en Android) front-end realisatie?
		C2	Welke tool/framework bied ondersteuning aan de geselecteerde programmeertalen uit deelvraag A4?
		C3	Hoe onderscheidt de tool/framework zich van de rest?
D	Wat zijn de sterke en zwakke punten van elk(e) onderzocht(e) tool/framework voor Flusso?	D1	Welke tool/framework past het beste bij de eisen die tot stand komen uit deelvraag A?

Multiplatform front-end development met tools/frameworks

E	Welke top drie tools/frameworks zijn het meest geschikt voor de realisatie van applicaties voor Flusso?	E1	Welke tool/framework ondersteunt het realiseren van de lunchapp?
F	Op welke wijze ondersteunen de top drie tools/frameworks de realisatie van de lunch app?	F1	Wat is de learning curve van de tool/framework?
		F2	Welke functionele beperkingen heeft de tool/framework laten zien tijdens de productie?
		F3	Op welke wijze heeft de tool/framework werk uit de handen van de programmeur genomen?
		F4	In hoeverre is het eindresultaat uit de tool/framework gelijk aan het functioneel/technisch ontwerp?
G	Op welke manier kunnen er functies toegevoegd worden aan het bestaande tool/framework?	G1	Op welke manier kunnen gegenereerde lay-outs/schermen worden aangepast buiten het framework om?
		G2	Op welke manier kunnen er native functies van mobiele telefoons worden toegevoegd aan de <i>tool/framework</i> ?
		G3	Is de manier waarop de native functies worden toegevoegd <i>compatible</i> met andere frameworks?

Bijlage 2. Evaluatie

In de eerste paar dagen voor en tijdens mijn afstudeerstage had ik een gevoel van twijfel en zenuwen. Hoewel ik wel wist dat ik dit proces tot een goed einde kon brengen, bleef er een naar gevoel hangen. Ik ervaar het als een groot geluk dat ik begeleid werd door zowel Harry Beerlage en Lars van Mourik. Niet alleen hebben ze mij ondersteund bij de realisatie van mijn stage/scriptie, zij gaven mij ook het vertrouwen en de zekerheid die ik nodig had om dit allemaal te kunnen. Ik ervaar het samenwerken met beide dan ook als zeer positief.

De voorbereiding van het onderzoek, het plan van aanpak (pva), verliep erg goed. Harry Beerlage benadrukte hoe belangrijk een pva is, en hoeveel tijd het kan besparen wanneer deze goed wordt gemaakt. Dit is in de praktijk ook zo gebleken. Door het goed/duidelijk opstellen van een pva werd mij helemaal duidelijk wat ik moest gaan doen, en in welke tijdsperiode dit gedaan moest worden. Tijdens mijn onderzoek was ik in staat om steeds voor de deadline's te blijven en daardoor mooi op schema te zitten. Hierdoor had ik op het einde genoeg tijd om een mooie scriptie te kunnen opstellen waarin mijn onderzoek goed naar voren komt.

Om tijdens mijn onderzoek zo goed mogelijke resultaten te kunnen garanderen, heb ik elk essentieel onderdeel gevalideerd. Om een duidelijk beeld te krijgen van de wensen binnen Flusso heb ik een intern interview gehouden. Om bestaande hulpmiddelen binnen te halen is een literatuuronderzoek gedaan naar eerdere studies. De gevonden hulpmiddelen zijn uiteindelijk gefilterd aan de hand van de wensen binnen Flusso. Om de realisatie van de hulpmiddelen te valideren werd de lunch applicatie gemaakt. Om te valideren dat de lunch applicatie een goed referentie applicatie is binnen Flusso, is deze vergeleken met eerder gemaakte applicaties. De uiteindelijke resultaten van de realisatie zijn getest op de punten van het functioneel/technisch ontwerp waaraan de lunch applicatie moet voldoen.

Over de resultaten van het onderzoek ben ik in het algemeen tevreden, alleen komen de genererende hulpmiddelen niet zo goed tot hun recht. Omdat de realisatie van de lunch applicatie totaal niet ging met het Intel XDK framework, kon er ook geen applicatie opgeleverd worden. Hoewel ik mij realiseer dat dit ook een resultaat is, voelt het toch verkeerd. Het liefst had ik drie werkende lunch applicaties opgeleverd, om vervolgens te kijken welke realisatie het beste tot stand is gekomen.

Samenvattend heb ik mijn afstudeerstage bij Flusso ervaren als een hele leerzame tijd waar ik veel geleerd heb over de ontwikkeling van hybride applicaties. Naast een succesvolle stage kan ik ook met trots zeggen dat Flusso mij een baan heeft aangeboden, zodat ik direct na mijn school aan de slag kan als Consultant bij Flusso!

Bijlage 3. Interview Rapport



Wensen tool/framework

Student:	E-mailadres:	Opleiding	Student nummer:
Tom Oosterwijk	tom.oosterwijk@student.hu.nl	Technische Informatica	1608726

Datum: 20-03-2015

Periode: 16-02-2015 / 27-02-2015

Versie:	Datum:	Aanpassingen:
0.1	19-02-2015	Concept Versie
0.2	20-02-2015	Eerste invulling
0.3	26-02-2015	Verwerking resultaten
0.4	16-03-2015	Verwerking feedback derde partij
1.0	20-03-2015	Finale versie

1 Inleiding

In februari 2015 is aan de werknemers/werkgevers van Flusso een gesloten interview gehouden in de vorm van een vragenlijst. Het interview heeft als doel om de wensen over een tool/framework vanuit Flusso duidelijk te krijgen. Het interview is gemaakt via het standaard google formulier met een Flusso google drive account. Hiervoor is gekozen omdat elke werknemer binnen Flusso een google account heeft en hierdoor selectief kunnen worden uitgenodigd voor deelname.

Van de 40 verstuurde vragenlijsten zijn er 26 reacties gekomen. Hieronder vielen vijf personen die zich afmeldden met als reden dat hij of zij niet genoeg technische kennis of belangen had in het onderzoek naar de tools/frameworks. De overige 21 reacties werden gebruikt om de wensen/eisen inzichtelijk te maken.

Het interview wordt in principe anoniem afgenomen, toch kan er vanuit worden gegaan dat ieder ingevuld interview een reactie is van een werknemer binnen Flusso. Het google formulier laat één reactie per gebruiker toe en de gebruiker moet ingelogd zijn met een Flusso werk account. De werknemers hebben overigens wel de keuze om hun naam op te geven, maar dit is vrijblijvend. Omdat het interview meerdere open vragen bevat, kan het handig zijn om open antwoorden te verduidelijken met een gesprek. Om dit te kunnen doen moet bekend zijn wie het interview had ingevuld.

De interviewvragen zijn tot stand gekomen in overleg met Lars van Mourik, Frank Staijen en Harry Beerlage. Dit verslag geeft het interview weer en behandelt de gevonden resultaten. De resultaten zullen worden gebruikt bij het selecteren van de juiste tool/framework voor Flusso.

2 Interview vragen

Vraag	Type	Mogelijke antwoorden
Wat is jouw naam?	Open vraag	
Op welke afdeling binnen Flusso ben jij werkzaam?	Open vraag	
Aan welke platform(s) moet de tool/framework ondersteuning bieden naast Apple en Android?	Selectievakjes	<div>Windows Mobile</div> <div>Windows Phone</div> <div>Blackberry OS</div> <div>Symbian</div> <div>Geen, Android en Apple is genoeg</div> <div>Anders:</div>
Er zijn (veel) gratis tools/frameworks te gebruiken. Waarin moet een tool/framework zich onderscheiden om het waard te zijn er geld voor uit te geven?	Open vraag	
Wat zijn de maximale jaarlijkse kosten per ontwikkelaar die een tool/framework met zich mee mag brengen?	Meerkeuze	<div>Geen kosten</div> <div>Minder dan 50€</div> <div>Minder dan 100€</div> <div>Minder dan 200€</div> <div>Minder dan 500€</div> <div>Meer dan 500€</div>
Welke technieken zou jij in een tool/framework willen terug zien en waarom?	Open vraag	
Welke technieken zou een klant van Flusso in een tool/framework willen terug zien?	Open vraag	
In welke mate vindt u het gewenst dat het tool gebaseerd is op open source?	Meerkeuze	<div>Open source noodzakelijk</div> <div>Liever wel, maar hoeft niet</div> <div>Geen voorkeur</div> <div>Liever niet, maar kan wel</div> <div>Absolut geen open source</div>
Welke programmeer talen moet de tool/framework ondersteunen?	Selectievakjes	<div>Java</div> <div>Progress/OpenEdge</div> <div>Java Script</div> <div>Action Script</div> <div>CSS</div> <div>Als het maar werkt</div> <div>HTML</div> <div>Anders:</div> <div>C++/C#</div> <div></div>
Welke native functies moet de tool/framework minimaal kunnen ondersteunen?	Selectievakjes	<div>Accelerometer</div> <div>Geolocation</div> <div>Camera</div> <div>Storage</div> <div>Capture</div> <div>Notification</div> <div>Compass</div> <div>Vibration</div> <div>Contacts</div> <div>Bluetooth</div> <div>File</div> <div>Anders:</div>
Welke soort type applicaties moet de tool/framework helpen realiseren?	Meerkeuze	<div>Hybride en web based applicaties</div> <div>Hybride en native applicaties</div> <div>Hybride, native en web based applicaties</div> <div>Hybride</div>
Welke user interface(ui) moet de tool/framework opleveren?	Meerkeuze	<div>Een native look & feel ui</div> <div>Een web based ui</div> <div>Anders:</div>

Tabel 1 Interview overzicht

3 Resultaten

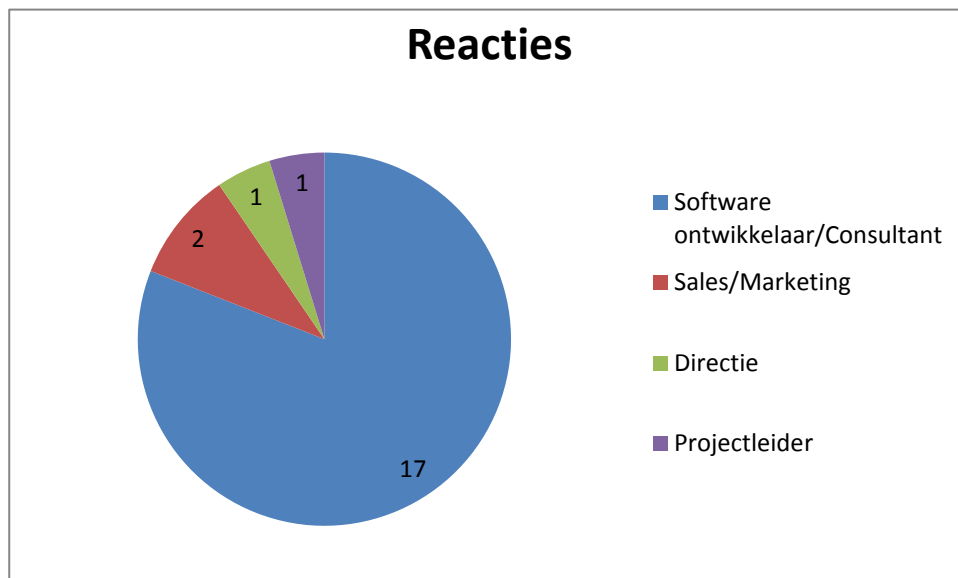
In dit hoofdstuk worden de resultaten van het interview besproken. Bij de meerkeuze vragen worden figuren met resultaten getoond die samen uit moeten komen op 100%. Bij selectievakjes vragen is de gebruiker instaat meerdere antwoorden te geven en kan het totale resultaat dus uitkomen boven de 100%.

3.1 Wat is jouw naam?

Deze vraag was optioneel en had als doel om de afnemer instaat te stellen contact op te nemen met de geïnterviewde indien er vragen waren over de gegeven antwoorden. De namen zelf zijn niet relevant voor het onderzoek en worden niet getoond bij de resultaten.

3.2 Op welke afdeling binnen Flusso ben jij werkzaam?

Om onderscheid te kunnen maken tussen de wensen van verschillende afdelingen binnen het bedrijf, moest de geïnterviewde persoon aangeven binnen welke afdeling hij/zij werkend is. Dit onderscheid zal gebruikt worden bij de vragen 3.4, 3.5 en 3.7.



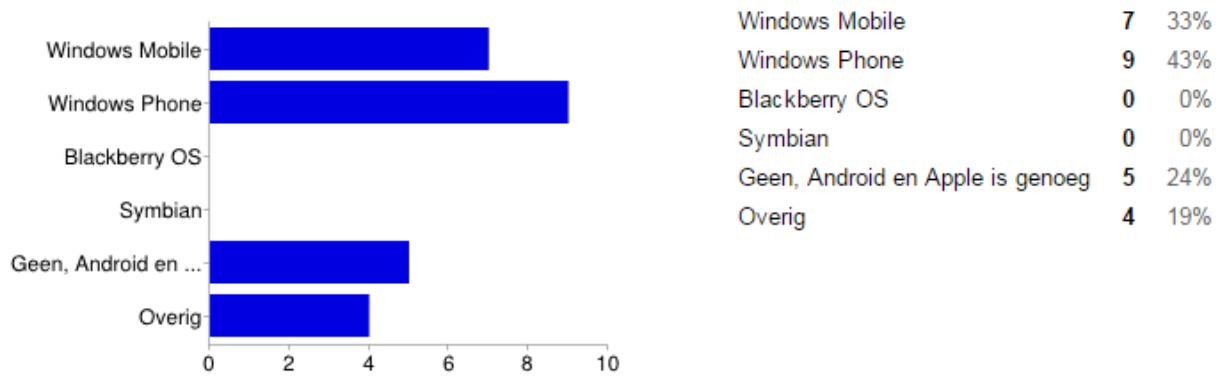
Figuur 19 Afdelingsoverzicht interview

Uit figuur 1 ontstaat een duidelijk beeld dat de meeste reacties afkomstig zijn van de software ontwikkelaars/consultants binnen Flusso. Hoewel het aantal reacties van de Sales/Marketing, Directie en de Projectleider weinig lijkt, is dit in werkelijkheid een score van 100%. Binnen Flusso is er één directeur, zijn er 2 mensen in de Sales/Marketing en loopt er één algemene projectleider rond.

3.3 Aan welke platform(s) moet de tool/framework ondersteuning bieden naast Apple en Android?

Een belangrijk aspect van een multi platform tool/framework is het aantal verschillende platforms dat deze kan bereiken. Aan de werknemers/werkgevers van Flusso werd de bovenstaande vraag gesteld en daar kwam het volgende uit:

Multiplatform front-end development met tools/frameworks



Figuur 20 Ondersteunde platforms

De resultaten van Figuur 2 geven aan dat binnen Flusso de wens is om naast Android en Apple ook de Windows Phone en zelfs de Windows Mobile platformen te ondersteunen. Een groep van vijf personen geeft aan dat alleen Android en Apple voldoende is. Een nog kleinere groep van vier personen zou graag een overig platform zien of had hier geen mening over.

3.4 Er zijn (veel) gratis tools/frameworks te gebruiken. Waarin moet een tool/framework zich onderscheiden om het waard te zijn er geld aan uit te geven?

Binnen Flusso zijn er verschillende groepen met een eigen idee over wat een tool/framework kan betekenen voor Flusso. Om dit beeld helder te krijgen zijn de groepen opgedeeld in groep 1 (Sales/Marketing en directie) en groep 2 (ontwikkelaars en projectleider).

Op de vraag wanneer een tool/framework het waard is om geld aan uit te geven, schetste groep 1 het volgende beeld:

Wanneer een tool/framework kosten met zich mee brengt, moet er achter de tool/framework een (grote) organisatie zitten. Hierdoor kan de tool/framework toekomstbestendig zijn en dat biedt Flusso de mogelijkheid zijn diensten aan klanten aan te bieden. Het uiteindelijke doel van de tool/framework moet zijn om het product zo snel mogelijk op de markt te zetten door het ontwikkelen zo gemakkelijk mogelijk te maken.

Groep 2 schetste het volgende beeld:

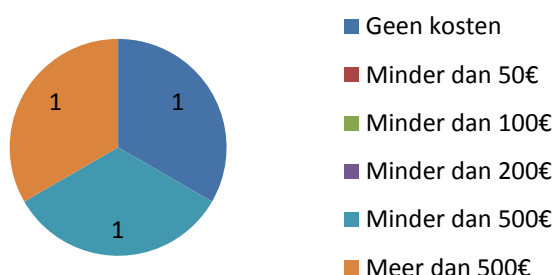
Een belangrijk en veel genoemd aspect is de return of investment. De kosten die de tool/framework met zich mee brengen moeten goedkoper zijn dan dat de kosten van de ontwikkeling zouden zijn zonder een tool/framework. Daarnaast moet de time to market sneller zijn ten opzichte van gratis open source tools/frameworks.

Een ander belangrijk punt dat wordt gemeld is het belang van een grote community, goede documentatie en dat de tool/framework in ontwikkeling blijft en mee gaat met de tijd. Dit betekent echter niet dat de tool/framework gebruik moet maken van state of the art technieken waarvan nog niet duidelijk is of deze over vijf jaar nog bestaan. Voorop staat dat de tool/framework de verwachting moet hebben om over vijf jaar nog steeds te bestaan, en dat het een volwassen tool/framework is zonder kinderziektes.

3.5 Wat zijn de maximale jaarlijkse kosten per ontwikkelaar die een tool/framework met zich mee mag brengen?

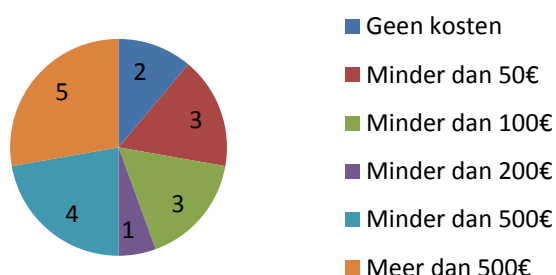
In de vorig gestelde vraag (zie 3.4) is er verantwoord wanneer een tool/framework het waard is om geld in te investeren. Een vervolg vraag is hoeveel geld er geïnvesteerd mag worden in de tool/framework. De resultaten van deze vraag zijn verdeeld over drie figuren.

Sales/Marketing & directie



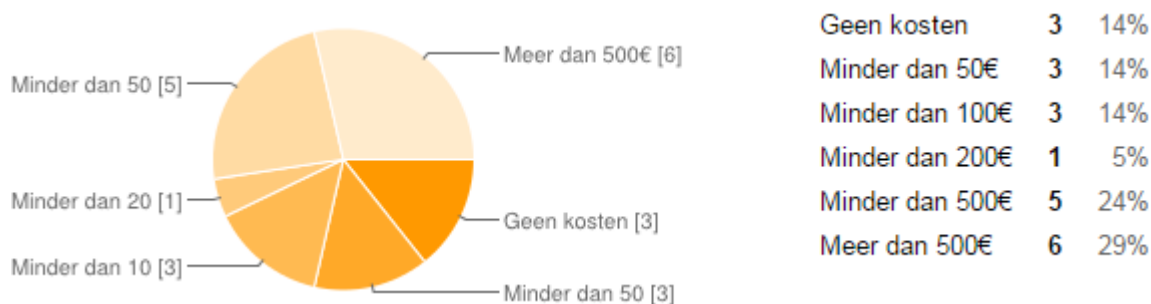
Figuur 22 Sales/Marketing & Directie kostenplaatje

Ontwikkelaars & projectleider



Figuur 21 Ontwikkelaars & projectleider kostenplaatje

De resultaten van figuur 4 tonen uiteenlopende ideeën. Er ontstaat geen duidelijk beeld over de gewenste prijs van een tool/framework. De resultaten van figuur 3 zijn nagenoeg hetzelfde. De resultaten liggen allemaal erg dicht bij elkaar en er is niet één eenduidige antwoord uit te ontwaren.



Figuur 23 Totaal kostenplaatje

Wanneer de twee groepen bij elkaar worden gezet zoals te zien in Figuur 5, blijft het beeld over de kosten hetzelfde. Figuur 5 toont dat, indien aan de eisen/wensen (zie 3.4) worden voldaan, de meerderheid het reëel vindt om meer dan €200 aan een tool/framework te besteden.

3.6 Welke technieken zou jij in een tool/framework willen terug zien en waarom?

Uit de resultaten over de gewenste technieken aan een tool/framework werden de volgende wensen inzichtelijk:

De tool/framework moet werken met een bekende programmeertaal zodat er snel/gemakkelijk met de tool/framework gewerkt kan worden zonder dat men zich moet verdiepen in een vreemde taal. Talen die genoemd worden als bekend zijn Java, Javascript, CSS en HTML(5). Een framework dat werkt met deze onderliggende technieken is Angular. Sommige hebben al eens met Angular gewerkt en zien deze graag terug.

Een andere wens is het mogelijk maken om unit-testing uit te voeren voor alle benodigde platforms. Hiermee kunnen applicaties snel en gemakkelijk getest worden, en omdat unit-testing op onderdeel niveau gaat weet de programmeur in het geval van een fout meteen waar hij/zij moet beginnen.

Daarnaast is er de wenstools/frameworks in te zetten om gedefinieerd UML's om te zetten naar code en andersom.

3.7 Welke technieken zou een klant van Flusso in een tool/framework willen terug zien?

Wanneer de medewerkers van Flusso gevraagd werd over de wensen van een tool/framework van een klant, kwamen er drie meningen naar voren.

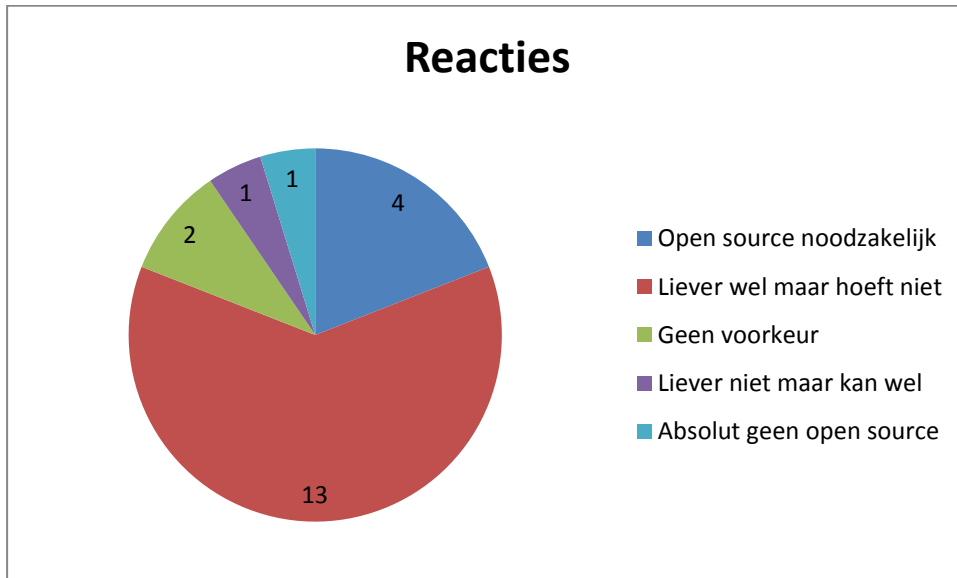
De eerste mening was dat het de klant niet zoveel zou uitmaken hoe het werkt, als het maar werkt en het de klant niet veel geld kost. De tool/framework moet dus in staat zijn om de gewenste functionaliteiten van de klant te ondersteunen.

De tweede mening is dat de klant wenst dat gebruik wordt gemaakt van een "Develop once, deploy to many" techniek. Dit houdt in dat uit één keer programmeren voor elk operating system een applicatie rolt. Daarnaast vindt de klant het belangrijk dat er geautomatiseerde testen gedaan worden om met een grotere zekerheid te kunnen garanderen dat de applicatie goed werkt.

De derde mening heeft betrekking op de front-end kant. De tool/framework moet een mooie, native look & feel presenteren maar moet qua performance net zo snel werken als een native applicatie.

3.8 In welke mate vindt u het gewenst dat het tool gebaseerd is op open source?

Op de vraag of de tool/framework gebaseerd moest worden op een open source techniek ontstond een duidelijk beeld.



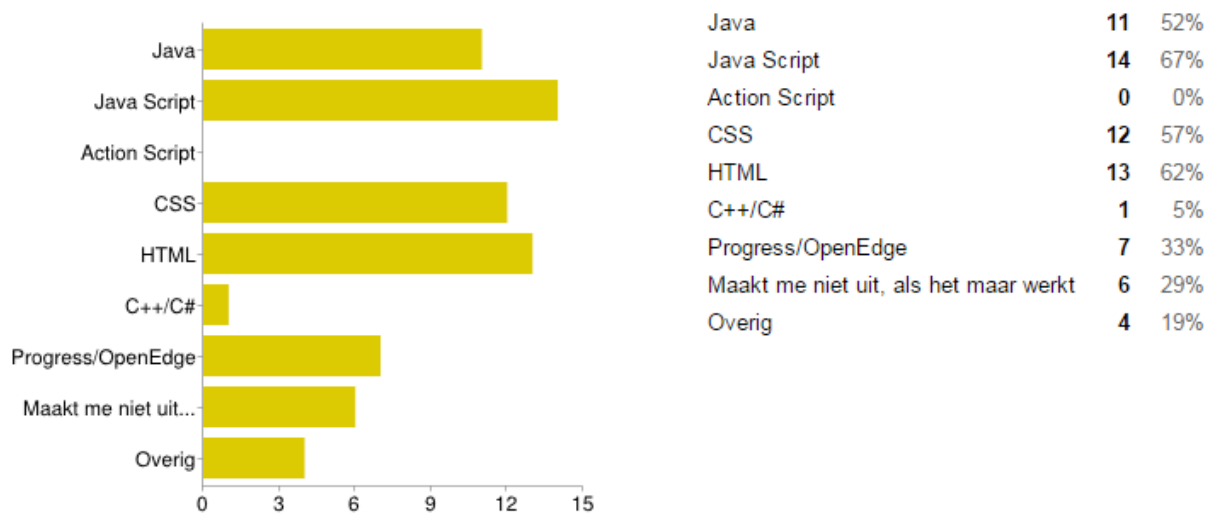
Figuur 24 Open source

Figuur 6 laat zien dat het merendeel van de geïnterviewde een voorkeur heeft voor een tool/framework dat is gebaseerd op open source. Dit betekent dat een tool/framework die open source is een groter voorkeur geniet dan een tool/framework die dat niet is. Dit houdt echter niet in dat de tool/framework perse open source moet zijn. Wanneer de tool/framework zich onderscheid maar geen open source techniek gebruikt zal de tool/framework geselecteerd worden.

3.9 Welke programmeer talen moet de tool/framework ondersteunen?

Aangezien de werknemers (ontwikkelaars/consultants) van Flusso uiteindelijk met de tool/framework zullen gaan werken, is het van belang om te achterhalen welke programmeer talen zij het liefste terug zien in de tool/framework.

Multiplatform front-end development met tools/frameworks

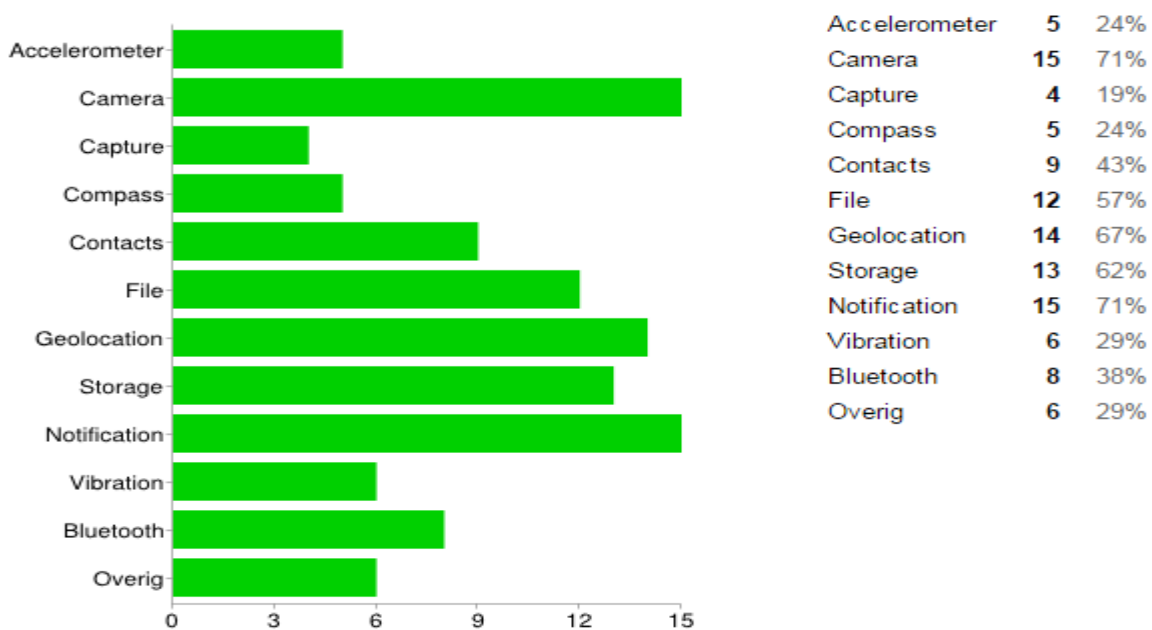


Figuur 25 Programmeer talen

In Figuur 7 worden de wensen binnen Flusso aangetoond. Een tool/framework moet kunnen werken met Java, Java Script, CSS en HTML. Deze vier talen staan boven aan het lijstje en worden door meer dan de helft van de ondervraagden aangegeven. Iets daar onder en gewenst door één derde van de ondervraagden is Progress/OpenEdge. Indien een tool/framework Progress/OpenEdge zou ondersteunen zou dit een mooi pluspunt voor de tool/framework zijn.

3.10 Welke native functies moet de tool/framework minimaal kunnen ondersteunen?

Naast de ondersteuning naar platformen en programmeertalen moet er ook gekeken worden naar het technische kunnen van een tool/framework. Aan de medewerkers van Flusso is gevraagd welke native mobiele functionaliteiten minimaal moeten worden ondersteund.



Figuur 26 Ondersteuning native functies

In Figuur 8 is af te lezen dat het gebruik van een Camera en een notificatie absolute musts zijn. Ook het kunnen bepalen van de locatie door middel van Geolocation en het gebruiken van file transport en storage staan hoog op de eisenlijst. Wanneer een tool/framework ook bij de Contact gegevens kan komen van de telefoon is dat een mooi pluspunt.

Hoewel dit een vraag is die betrekking heeft op de minimale functionaliteiten, is op elke functionaliteit minimaal 4 keer gestemd. Hieruit kan geconcludeerd worden dat hoe meer functionaliteiten er ondersteund worden door een tool/framework, hoe meer de tool/framework Flusso van dienst kan zijn bij het ontwikkelen van applicaties.

3.11 Welke soort type applicaties moet de tool/framework helpen realiseren?

In de mobiele applicaties zijn er drie onderscheiden te maken wanneer het gaat over mobiele applicaties, namelijk Hybrid, native en web based applicaties. Aan de werknemers van Flusso is gevraagd welke applicaties moeten worden ondersteund door de tool/framework.



Figuur 27 Realiseerbare applicaties

Het is binnen Flusso belangrijk dat de tool/framework in staat moet zijn om support te leveren binnen de ontwikkeling van alle drie de soorten applicaties zoals te zien in Figuur 9. Op deze manier hoeft een ontwikkelaar maar één tool/framework goed te kennen om alle applicaties ontwikkelen in plaats van drie verschillende hulpmiddelen.

3.12 Welke user interface(ui) moet de tool/framework opleveren?

Een belangrijk aspect van een applicatie is hoe deze getoond wordt aan de gebruiker. Aan de medewerkers van Flusso is gevraagd wat voor een soort user interface de tool/framework moet opleveren. Uit Figuur 10 is op te maken dat er een grote wens is voor het realiseren van native look & feel user interfaces.



Figuur 28 User interface

Bijlage 4. Functioneel & Technisch ontwerp



Lunch applicatie

Student:	E-mailadres:	Opleiding	Student nummer:
Tom Oosterwijk	tom.oosterwijk@student.hu.nl	Technische Informatica	1608726

Datum: 27-02-2015

Periode: 16-02-2015 / 27-02-2015

Versie:	Datum:	Aanpassingen:
0.1	16-02-2015	Concept Versie
0.2	20-02-2015	Eerste invulling
0.3	23-02-2015	Verwerking feedback bedrijfsbegeleider
0.4	24-02-2015	Verwerking feedback bedrijfsbegeleider
1.0	16-03-2015	Finale versie

1 Inleiding

Bij Flusso is het idee ontstaan om een zogeheten “lunch app” te maken. Bij Flusso is er elke dag een bedrijfslunch waar iedereen individueel geld inlegt en een medewerker hiermee inkopen doet. Vanwege de flexibele bezetting zijn er elke dag andere behoeften en omdat is gebleken dat veel mensen geen contant geld op zak hebben, is de lunchpot meestal niet goed genoeg gevuld om boodschappen van te doen. Hierdoor ontstaan allerlei praktische problemen en doordat het saldo niet goed wordt bijgehouden kan het zijn dat sommige medewerkers (veel) te veel en sommige (veel) te weinig geld inleggen. Door de lunchpot in een applicatie te administreren verwacht Flusso dit praktische probleem te tackelen.

Voor het realiseren van de “lunch app” wordt er een functioneel & technisch ontwerp gemaakt. Deze twee ontwerpen worden gebruikt om duidelijkheid te creëren over de applicatie zodat er achteraf geen ongewenste verrassingen zijn.

In dit document wordt er gesproken over geld en krediet van de deelnemers van de lunch app. Dit is fictief geld; er zal via de applicatie geen gebruik worden gemaakt van echte transacties. Wanneer de gebruikers geld storten via de applicatie betekent dit dat het echte geld in de algemene lunch pot is gestopt of dat dit bedrag is betaald in de supermarkt. Er wordt in dit onderzoek vanuit gegaan dat de medewerkers van Flusso hier eerlijk mee om gaan.

2 Doelgroep

De lunch app wordt gemaakt voor de werknemers binnen Flusso en zal gebruikt worden om voor ieder persoon een saldo bij te houden dat gebruikt kan worden voor de dagelijkse lunch.

3 Applicatie functionaliteit

Om af te bakenen wat de lunch applicatie allemaal moet kunnen, is er een MoSCoW analyse gemaakt op de applicatie functionaliteiten.

Must Have

- Gebruiker identificeren.
- Overzicht van de lunchdeelnemers van de huidige dag tonen.
- Gebruikers zichzelf en andere werknemers van Flusso laten aanmelden (€2.50 krediet afschrijven)/afmelden (€2.50 krediet bijschrijven) voor de lunch.
- Gebruikers toestaan krediet te laten toevoegen.

Should Have

- Overzicht tonen van de lunch deelnemers van andere dagen in de week.
- Overzicht tonen van de standaard deelnemers aan de lunch. Deze groep wordt bepaald aan de hand van lunch deelnamen in het verleden. De deelnemers staan al wel in de lunchlijst, maar staan nog niet geregistreerd als mee eters.
- Indien de schermgrootte het toe laat een weekoverzicht tonen.

Could Have

- Informatie bufferen wanneer er (tijdelijk) geen netwerk verbinding is.

Would Have

- Echt geld gebruiken in plaats van nep krediet.
- Alle *belangrijke* gegevens crypten.
- Gebruikers laten inloggen met een username + wachtwoord.
- Gebruikers instaat stellen hun wachtwoord op te vragen.

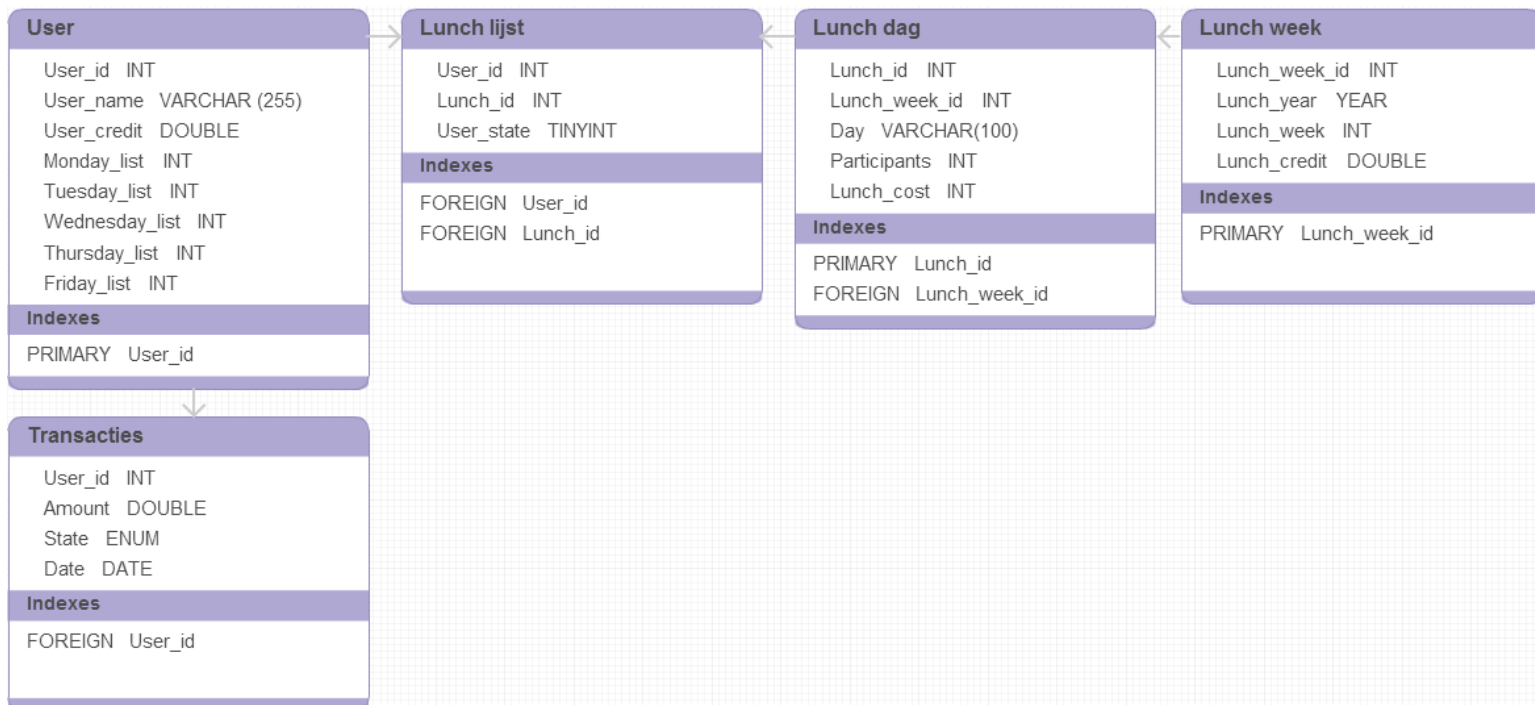
4 Applicatie Structuur

De lunch app bestaat globaal gezien uit drie onderdelen: een database, een Application Programming Interface (API) en een applicatie. Deze onderdelen worden nader behandeld.

4.1 Database

De database heeft de functie om alle belangrijke waarden van de lunch app op te slaan. In de database zijn de gegevens van elke gebruiker bewaard en worden de gemaakte transacties opgeslagen en beheerd. Daarnaast wordt er van elke lunch bijgehouden wie er deelnam. Een overzicht van de database is te zien in afbeelding 1.

Voor de database wordt gebruik gemaakt van een (No)SQL database. Deze wordt gehost op de server(s) van Flusso en is te benaderen door een API (zie paragraaf 3.2).



Figuur 29 Database ontwerp

4.1.1 Tabellen

Binnen de database zijn er verschillende tabellen die helpen om de informatie uit elkaar te houden en koppeling tussen informatie mogelijk te maken.

User

In de tabel 'User' wordt alle informatie opgeslagen die persoonsgebonden zijn. In het geval van de lunch app is dat de volledige naam van de gebruiker (User_name) en het huidige krediet dat de gebruiker heeft.

Daarnaast wordt er voor elke werkdag bijgehouden of een user al automatisch aan de lunchlijst van een bepaalde dag kan worden toegevoegd. Dit wordt gedaan door aan het begin van een nieuwe week te kijken of de user in de afgelopen 6 weken minimaal 2 keer op een bepaalde dag

(b.v. maandag) heeft meegegeten. Indien dit het geval is, komt er (bij maandag) een 1 te staan. Indien dit niet het geval is zal er een 0 staan.

De User_id is een uniek identificatie nummer die wordt gegenereerd wanneer de gebruiker wordt toegevoegd aan de database. Er mag geen andere gebruiker bestaan met het zelfde user_id.

Transacties

De tabel 'Transacties' beschikt over alle gegevens die betrekking hebben tot het storten/opnemen (State) van geld (Amount) voor elke gebruiker (FK User_id) op elke dag (Date).

Lunch lijst

De tabel 'Lunch List' is een koppel tabel die users (FK User_id) koppelt aan een lunch (FK Lunch_id). Standaard worden users al toegevoegd aan de lunch lijst van de komende lunches wanneer zijmeerdere keren in een maand hebben deelgenomen aan een lunch (zie tabel User). Dit betekent echter nog niet dat de users met de komende lunches willen mee eten: dit moet via de applicatie worden aangegeven. De User_state houdt bij of de gebruiker wel (TRUE,1) wil mee eten of niet (FALSE,0). Standaard staat de User_state op false.

Lunch dag

De tabel 'Lunch dag' beschikt over de gegevens die te maken hebben met een lunchdag (Day) van een bepaalde week (Lunch_week_id). Er wordt per week voor elke werkdag een nieuwe lunch aangemaakt met een eigen unieke Lunch_id. Zowel de kosten van de boodschappen (lunch_cost) als het aantal deelnemers voor de lunch wordt opgeslagen. Deze gegevens worden gebruikt om een gemiddeld kostenplaatje te berekenen. Hiermee kan worden aangetoond of de standaard lunchkosten van €2.50 te hoog, te laag of precies goed is.

Lunch week

De tabel 'Lunch week' houdt alle gegevens bij van de lunches bij op week niveau. Elke week wordt er een nieuwe 'Lunch week' gemaakt met een daarbij horend unieke lunch_week_id. Van de lunch week wordt bijgehouden welke week en welk jaar het is. Deze informatie is nodig om de lunchgeschiedenis van een gebruiker te kunnen achterhalen. Tevens wordt de waarde van de lunchpot bijgehouden (Lunch_credit). Het eindbedrag van een lunchweek 1 is het begin bedrag van lunchweek 2.

4.1.2 Beveiliging

Er zit geen encryptie op de gegevens van de database omdat dit buiten de scope van de lunch applicatie valt. Daarnaast wordt er geen controle gemaakt of een inkomende transactie wel geldig is en afkomstig van een Flusso medewerker.

4.2 API

Om informatie in/uit de database te krijgen, wordt gebruik gemaakt van een API waar de applicaties mee kunnen communiceren. De API dient als tussenlaag voor de applicatie en de bewaarde data in de database.

4.2.1 API calls

In tabel 1 wordt schematische weergegeven welke API calls er te maken zijn, welke parameters deze nodig heeft en wat de return type is.

User		
Api Call	Parameters	Return
getAllUsers		JSON users Example: <pre>{ "users": [{ "firstName": "Klaas", "lastName": "A" }, { "firstName": "Jan", "lastName": "Ba" }] }</pre>
addUserToLunch	String firstName, String lastName, Date date	
toggleUserLunchStatus	String firstName, String lastName, Date date	
Lunch		
Api Call	Parameters	Return
getLunchList	Date date	JSON users Example: <pre>{ "users": [{ "firstName": "Klaas", "lastName": "A", "balance": 17.5, "joiningLunch": false }], "date": "19-02-2015", "lunchPot": "19-02-2015" }</pre>
Transacties		
Api Call	Parameters	Return
depositCash	Int User_id, Float amount	Float userBalance
payLunch	Int User_id, Float amount	Float newBalance

Tabel 1 API voor de lunchapp

4.2.2 API beveiliging

Hoewel de data op zich zelf niet beveiligd hoeven te worden, is het niet toegestaan dat derde partijen deze informatie zomaar aan de API kunnen opvragen. Dit betekent dat de afkomst van de call naar de API moet worden gevalideerd. Dit zal gedaan worden door bepaalde credentials vanuit de applicatie mee te sturen naar de API, zoals een naam en een wachtwoord.

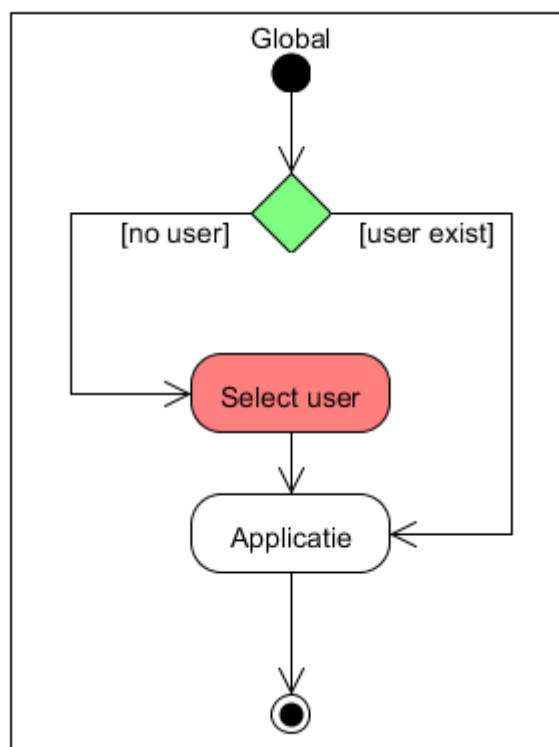
4.3 Applicatie

De applicatie is uiteindelijk het onderdeel dat de data aan de achterkant omzet tot een begrijpbare en logische user interface. Dit is het onderdeel dat zichtbaar wordt voor de gebruiker en waarmee de gebruiker gaat werken.

4.3.1 Globaal proces

In de Activity Diagram 1 is het globale proces te zien die de applicatie zal afleggen. De eerste keer dat de applicatie gedraaid wordt zal er nog geen gebruiker bekend zijn bij de applicatie. De gebruiker komt vervolgens bij een keuze scherm waar hij zijn naam kan kiezen.

Wanneer de gebruiker zijn naam heeft geselecteerd komt hij bij de applicatie terecht. Wanneer in het vervolg de applicatie wordt opgestart zal de (onthouden) gebruiker in een keer door gestuurd worden naar de applicatie.

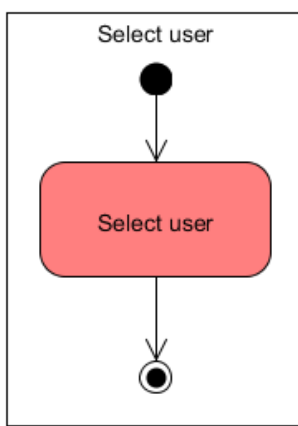


Activity Diagram 1 Het globale proces

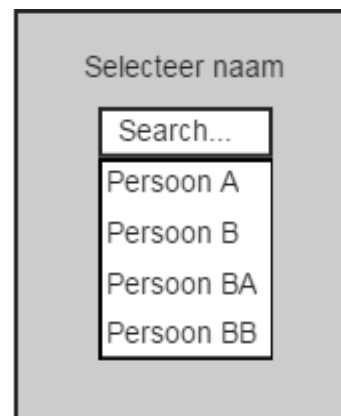
4.3.2 Aanmelden

Wanneer voor het eerst gebruik gemaakt wordt van de applicatie, moet de gebruiker aangeven welke Flusso werknemer hij is. Om de applicatie zo laagdrempelig mogelijk te houden wordt niet gewerkt met usernames en wachtwoorden. Dit betekent dat iedere gebruiker in staat is om het krediet van een ander persoon (en zichzelf) aan te passen. Er wordt uitgegaan van de eerlijkheid van de gebruiker zodat dit niet misbruikt wordt.

De namen worden alfabetisch geordend op achternaam zoals te zien in Wireframe 1. De namen zijn afkomstig van de centrale database systeem waar alle gebruikersnamen zijn opgeslagen. Wanneer de gebruiker op zijn of haar eigen naam drukt, onthoudt de applicatie de naam en wordt de gebruiker door gestuurd naar de applicatie zoals te zien in Activity Diagram 2.



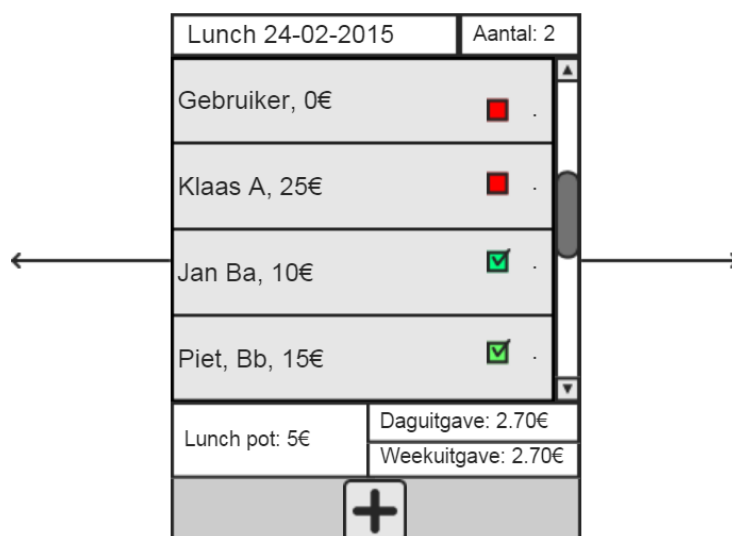
Activity Diagram 2 Select User



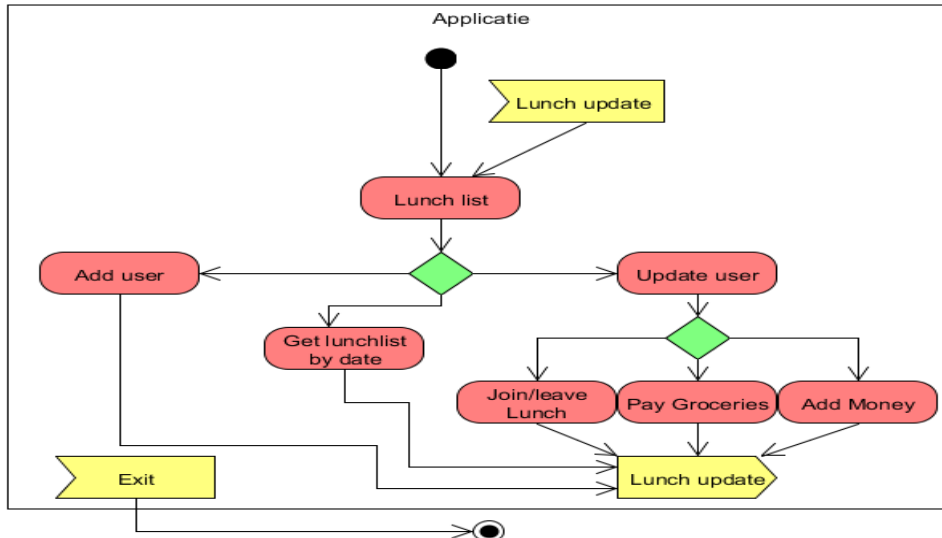
Wireframe 1 Select user

4.3.3 Applicatie proces

Wanneer de gebruiker diens naam heeft geselecteerd (of indien dit al gedaan was) komt de gebruiker in het hoofd scherm van de applicatie terecht zoals te zien in Wireframe 2. Binnen het hoofdscherm heeft de gebruiker verschillende opties die te zien zijn in Activity Diagram 3.



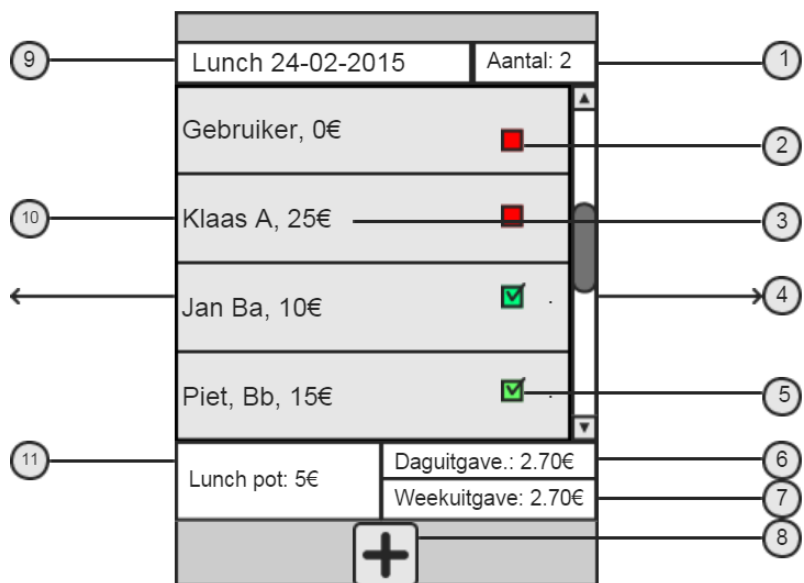
Wireframe 2 Hoofdscherm van de lunchapp



Activity Diagram 3 De lunch app applicatie

In Wireframe 3 wordt een duidelijk onderscheid gemaakt tussen de verschillende functionaliteiten die de applicatie biedt.

1. Aantal lunchdeelnemers.
2. Doet niet mee aan de lunch.
3. Krediet per persoon.
4. Naar links/rechts swipen voor lijst met datum van een dag later/eerder.
5. Doet mee aan de lunch.
6. De kosten per persoon voor de lunch van de dag (dit wordt pas berekend wanneer de boodschappen zijn gekocht).
7. De gemiddelde kosten per persoon voor de lunch per week.
8. Persoon toevoegen aan de lunch (zie Wireframe 1).
9. De datum van de lunchlijst.
10. Een overzicht van de mensen die wel/niet mee willen doen aan de lunch. Standaard wordt deze lijst gevuld met users die in het verleden regelmatig mee aten. De gebruiker ziet zichzelf altijd bovenaan het lijstje.
11. Krediet in de lunchpot, dit bedrag kan zowel plus als min zijn. Bij de initialisatie van de database zal dit bedrag op €0,-staan en zal verhoogd worden wanneer mensen krediet toevoegen.

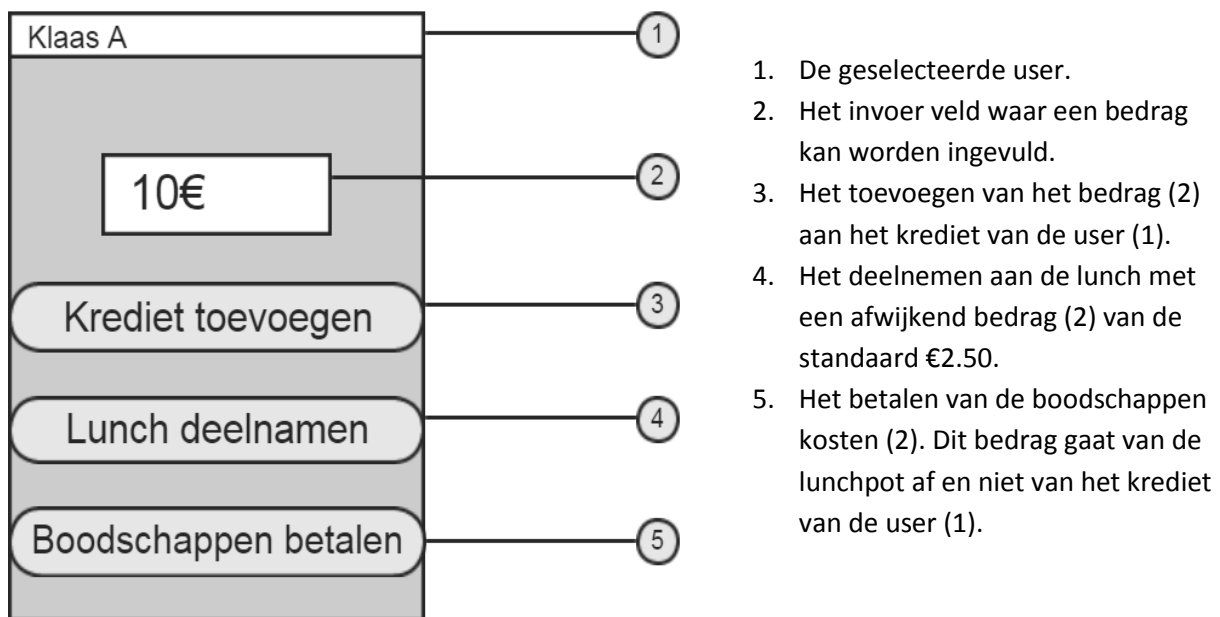


Wireframe 3 Lunchapp opgedeeld

Multiplatform front-end development met tools/frameworks

Om een persoon die zich al bevindt in de lunchlijst (10) mee te laten lunchen, moet er eenmaal geklikt worden op zijn of haar naam waarna de persoon een groen vinkboxje achter diens naam krijgt (5). De standaardkosten voor het mee lunchen bedraagt €2.50. Dit bedrag wordt van het krediet van de deelnemer afgehaald. Indien in het systeem al staat dat de persoon mee eet, werkt het als een afmelding en krijgt de persoon een rood leeg boxje achter zijn naam (2) en wordt diens ingelegde krediet vergoed.

Wanneer er langere tijd op de naam van een persoon (voorbeeld Klaas A) gedrukt wordt, krijgt de gebruiker van de applicatie een nieuwe scherm te zien (zie Wireframe 4) met de daarbij horende nieuwe functionaliteiten.



Wireframe 4 User opties

4.4 Look & feel

Voor de applicatie is het belangrijk dat er per operating system een native *look & feel* gecreëerd wordt. Met een native look & feel krijgt de gebruiker het gevoel dat er gewerkt wordt met een applicatie die ontwikkeld is voor zijn eigen OS (Wasserman, 2010). Zo moet de applicatie op een Android telefoon standaard Android eigenschappen hebben die de gebruiker gewend is. Hetzelfde geldt voor elk ander OS waarvoor de applicatie wordt gemaakt.

5 Benodigde technieken

Om de applicatie te kunnen realiseren moeten de volgende mobiele functionaliteiten worden ondersteund:

1. De gebruiker moet in staat zijn met vinger aanraking de applicatie te beheren.
2. De gebruiker moet in staat zijn met swipe de date van de lunch te verzetten.
3. De applicatie moet het OS van de telefoon kunnen detecteren om de daarbij horende look & feel te creëren.
4. De applicatie moet toegang hebben tot het netwerk voor het ophalen/versturen van informatie naar de server.
5. De netwerk status moet kunnen worden opgevraagd.
6. De applicatie moet een native look & feel hebben.

Bijlage 5. Inventarisatie hulpmiddelen

Tools

Phone Gap & Cordova

Phone Gap en Cordova zijn open-source, multiplatform tools die het realiseren van web/hybride applicatie's mogelijk maken door web applicaties in een native container(web view) te laten draaien (Dalmasso, Datta, Bonnet, & Nikaein, 2013). De web applicatie wordt gehost op de telefoon zelf, hierdoor is geen internet nodig om de applicatie te draaien. De web view leest de bestanden lokaal uit. Beide tools bieden een set aan mobiele APIs die de ontwikkelaar in staat stelt om met JavaScript de native functionaliteiten van een mobiel te bereiken binnen de web view (cordova.apache, z.d.).

Met Phone Gap/Cordova kan ontwikkeld worden voor elke populaire operating system(iOS, Android, Black Berry, Windows Phone) en is het beste te gebruiken voor projecten waar veel controle over de setup en ontwikkeling gewenst is (Christ, 2011).

Touchdevelop

Touchdevelop is een web applicatie die een gebruiker in staat stelt mobiele applicaties te maken met minimale code. Het programma is gemaakt om zelf codes te generen door user input die via aanraking plaats vindt. Touchdevelop wordt vooral gebruikt bij het maken van web applicaties maar het uitvoeren van native mobile functies wordt steeds meer ondersteund (Horspool & Tillmann, 2013).

Frameworks

Sencha Touch

Sencha Touch is een open source framework dat ontwikkelaars ondersteunt bij het ontwerpen, ontwikkelen en het uitbrengen van applicaties voor de desktop en mobiele apparaten op multiplatform basis. Het Sencha framework maakt gebruik van standaard web technieken zoals HTML5 en JavaScript. Door het grote user interface aanbod en het goed kunnen creëren van een native *look & feel* applicaties wordt Sencha Touch vaak in combinatie gebruikt met andere tools/frameworks(Phone Gap) voor een maximaal resultaat (Dalmasso, Datta, Bonnet, & Nikaein, 2013).

Altova-MobileTogether

Het MobileTogether framework van Altova ondersteunt de ontwikkelaar om multiplatform applicatie ontwikkeling mogelijk te maken. Met expressie talen als XPath en XQuery wordt data bereikbaar gemaakt. Daarnaast is met een *drag and drop*-interface een applicatie te maken zonder (gevorderde) programmeerkennis (Altova, z.d.).

Ionic

Ionic is een open source framework dat zich vooral richt op het maken van native/hybride uitziende applicaties. De applicaties die gemaakt worden met Ionic hebben een native *look & feel* die tot stand komt door web technologieën zoals HTML5, CSS en JavaScript (Ionicframework, z.d.). Daarnaast maakt Ionic gebruik van AngularJS, een client-side JavaScript framework dat de ontwikkelaars bij staat bij het realiseren van web/hybride applicaties (Branas, 2014).

Met de ondersteuning van HTML5 en de moderne API's voor de mobiele functionaliteiten die de mobiele van "vandaag" met zich meebrengen, past Ionic perfect bij het hier en nu. Ionic richt zich

dan ook volkomen op het heden en de toekomst en laat het verleden achter zich (Ionicframework, z.d.).

Intel XDK

Intel XDK is een gratis te gebruiken framework dat het realiseren van multiplatform hybride applicaties ondersteunt. Met Intel XDK kunnen applicaties voor Android, iOS, Windows 8, Amazon en Nook gemakkelijk worden gemaakt door middel van HTML5 en JavaScript code (Karadimce & Bogatinoska, 2014). Tevens biedt Intel XDK de mogelijkheid om met een *drag and drop*-editor een applicatie te maken zonder enige programmeer kennis.

Mobile Angular

Mobile angular is een open source HTML 5 framework dat samen met Bootstrap 3 en AngularJS interactieve mobiele applicaties ontwikkeld. Het Angular framework houdt zich bezig met de ontwikkeling van de front-end van een applicatie, de user interface. Om de applicaties hybride te maken en voor multiplatform geschikt te zijn maakt het gebruik van het Phone Gap/Cordova tool (MobileAngularui, z.d.).

Titanium- Appcelerator

Titanium is een open source framework dat de ontwikkelaar in staat stelt om hybride applicaties multiplatform te maken voor iOS, Android Black Berry. De programmeertaal die Titanium hanteert is JavaScript, die Titanium vervolgens zelf vertaalt naar native code (Dalmasso, Datta, Bonnet, & Nikaein, 2013).

Met een open source Software Developers Kit(SDK) voor meer dan 5000 mobiele toestellen en mobiele operating system APIs ondersteunt Titanium het maken van bijna elke applicatie voor elke telefoon (Appcelerator, z.d.). Om met Titanium te kunnen programmeren is een Apple ontwikkeling licentie vereist, en zijn sommige libraries niet open source waardoor deze zullen moeten worden aangeschaft (Danielsson, 2014).

MoSync

MoSync is een open source framework dat een ontwikkelaar in staat stelt met het schrijven van één code tot wel negen verschillende operating systems te bereiken, waaronder iOS, Android, Windows en Blackberry. Met de programmeertalen C/C++ of HTML5 en JavaScript is het framework in staat om web-based en hybride applicaties te ontwikkelen (Palmieri, Singh, & Cicchetti, 2012).

RhoMobile

Rhomobile is een open source multiplatform framework dat hybride applicaties kan ontwikkelen met de talen HTML5, CSS en JavaScript. Met RhoMobile krijgen applicaties een native *look & feel* waarmee de gebruiker een vertrouwd gevoel krijgt. Om met RhoMobile een applicatie te maken is een online IDE of een lokale IDE te gebruiken (Kim, Karunaratne, Regenbrecht, & Warren, 2015).

Genuitec

Genuitec biedt met het product MyEclipse een programmeeromgeving aan waar hybride multiplatform applicaties kunnen worden gemaakt voor zowel iOS als Android. Met MyEclipse is de ontwikkelaar in staat om Phone Gap applicaties te maken en gemakkelijk plug-ins toe te voegen. Om ervoor te zorgen dat de user interface er goed uit ziet kan gekozen worden om met Ionic of JQuery te werken. Daarnaast kan de ontwikkelaar snel widgets toevoegen met een *drag and drop* techniek (Genuitec: The Cloud Control Company).

Versie: 1.0

Datum: 2-6-2015

Qt

Qt is een open source multiplatform framework dat hybride applicaties kan ontwikkelen met de talen QT Quick, HTML5 en JavaScript. Met een *drag and drop* interface wordt het ontwikkelen van de applicatie makkelijker gemaakt waarmee snellere ontwikkeling tot stand komt (Qt.io, z.d.).

AppGyver-Supersonic

Supersonic is een gratis framework dat alleen gericht is op het maken van multiplatform hybride applicaties. De ondersteunde platformen zijn Apple en iOS. Om de applicaties een native *look & feel* te geven maakt Supersonic gebruik van het Ionic framework. Met de standaard web technieken JavaScript, HTML5 en een *drag and drop* interface is het tot stand komen van applicaties eenvoudig. Daarnaast is het Supersonic framework in staat content te kunnen genereren aan de hand van een op te geven web service API (Appgyver, z.d.).

Platforms

Mendix

Het Mendix App Platform richt zich op de ontwikkeling van Enterprise applicaties, met name het ontwikkelen en ondersteunen van hybride mobiele applicaties voor meerdere platformen. Met reeds gedefinieerde user interfaces voor mobiele apparaten die zorgen voor een native *look & feel*, stelt Mendix de ontwikkelaar in staat in korte tijd een applicatie te maken (Hadley, 2014).

Telerik AppBuilder

Met de Telerik AppBuilder wordt multiplatform ondersteuning voor iOS, Android en de Windows Phone aangeboden. Bij het tot stand brengen van de applicaties maakt de AppBuilder gebruik van de tool Cordova/Phone Gap. Met de webtechnieken HTML5 en JavaScript kunnen programmeurs de applicaties realiseren (Hasselknippe, 2014).

Xamarin

Xamarin is een platform waarmee hybride applicaties voor multiplatform ontwikkeld kunnen worden met de programmeertaal C# en een *drag and drop* interface. Waar de meeste tools/frameworks gebruik maken van web-based technieken, compileert Xamarin de C# source code naar verschillende native bundels, afhankelijk van het gekozen platform (Hasselknippe, 2014).

IBM MobileFirst

De IBM MobileFirst platform is een totaalpakket dat ondersteuning biedt bij de ontwikkeling van een applicatie, het leveren van de applicatie en het blijven onderhouden van de applicatie. Standaard komt het MobileFirst platform met een eigen ontwikkelomgeving, genaamd "IBM MobileFirst Studio". De ontwikkelaar is echter wel in staat een eigen IDE aan het platform te koppelen. Met de MobileFirst platform is de ontwikkelaar in staat om native, web-based en hybride applicaties te ontwikkelen.

Om hybride applicatie ontwikkeling mogelijk te maken, gebruikt IBM een combinatie van standard native en web-based code. De applicatie draait in een native container die vervolgens gebruik maakt van de browser om de HTML5, JavaScript en CSS af te beelden. De native container die gebruikt wordt is gebaseerd op Phone Gap en geeft de applicatie toegang tot de native apparaat functionaliteiten zoals de camera en de lokale opslag (IBM Corporation, 2014).

Multiplatform front-end development met tools/frameworks

Doordat de applicaties een combinatie zijn van native en web-based code, krijgt de gebruiker de snelheid en het uiterlijk die verwacht wordt van een native app en tegelijkertijd kan de applicatie op elke telefoon worden getoond door de web-based technieken.

Outsystems - Platform

Outsystems is een betaald platform dat ontwikkelaars ondersteunt met het ontwikkelen van iOS, Android en Winows Phone web/hybride applicaties. Met de keuze uit de programmeertalen Java, C#, HTML5, CSS3 en JavaScript is de ontwikkelaar in staat de applicaties vorm te geven terwijl Outsystems er voor zorgt dat de applicaties een native *look & feel* behouden (OutSystems, z.d.). Daarnaast werkt OutSystems met een *drag and drop* interface en model driven development om applicatie ontwikkeling te versnellen en vereenvoudigen.

Bijlage 6. Applicaties van Flusso

Qc4u

Device propertjes

Voor Qc4u is het nodig dat mobiel type specifieke gegevens kunnen worden opgevraagd. Uit de device informatie moet onder andere gehaald kunnen worden op welke soort type mobiel de applicatie draait, wat het model is van de mobiel en op welke *operating system* het mobiel draait.

Network information/toegang

Voor qc4u is het nodig dat de netwerk informatie kan worden opgevraagd. Dit heeft twee verschillende redenen.

De eerste reden is omdat de applicatie om de twee minuten wil synchroniseren met de back-end server. Wanneer geen netwerkverkeer mogelijk is moet de applicatie zijn veranderingen zelf bij houden.

De tweede reden heeft een financieel aspect. De applicatie verstuurt foto's van inspecties die de rekening flink kunnen laten oplopen wanneer dit via de eigen 3G-verbinding loopt. Door te kijken naar de netwerk informatie laat de applicatie alleen maar het versturen van foto's toe wanneer de applicatie zich op de WI-FI verbinding bevindt.

Camera

De camera wordt gebruikt voor het maken van foto's van de producten die geleverd worden in de containers.

Local notification

De *local notification* stelt de applicatie in staat om de gebruiker te notificeren wanneer bijvoorbeeld een update gedaan kan worden van de software. Daarnaast kan zo ook de gebruiker op de hoogte gesteld worden wanneer een order-wijziging plaats vindt. Zo kan de gebruiker geïnformeerd worden zonder dat de applicatie zelf open staat. *Local notification* wordt op het huidige moment nog niet toegepast binnen qc4u maar het zit wel in de planning.

Barcode scanner

De barcode scanner wordt gebruikt om het product dat wordt geïnspecteerd te identificeren. De naam van het product zit dan in barcode die vast gemaakt zit aan een product.

File transfer

Met *file transfer* wordt het mogelijk gemaakt om door middel van http of https bestanden te versturen/downloaden van een mobiel naar de server. Qc4u maakt hier onder andere van gebruik wanneer de gemaakte afbeeldingen naar de server moeten worden gestuurd.

Geolocation

De geolocation wordt gebruikt om de huidige positie van de mobiel op te vragen. Dit is handig wanneer de tracking van de applicaties moet worden bijgehouden. Geolocation wordt op dit moment nog niet toegepast binnen qc4u maar het zit wel in de planning.

Touch screen

De applicatie moet reageren wanneer het scherm van de mobiel wordt aangeraakt.

Inspectandgo

Batterij status

Het opvragen van de batterij status is van belang om te voorkomen dat data verloren gaat bij uitval van het apparaat. Wanneer de status van de batterij “kritiek” is, worden de gegevens van de Inspectandgo applicatie opgeslagen op het filesysteem.

Camera

De inspecteurs die gebruik maken van de Inspectandgo applicatie maken afbeeldingen van geconstateerde gebreken in een woning. Om dit te doen wordt gebruik gemaakt van de device camera.

Device

Binnen de Inspectandgo applicatie worden PDF's getoond. Windows, iOS en Android gaan alle drie anders om met het openen van een PDF. Met de device data weet de applicatie met welk besturingssysteem hij te maken heeft en kan de juiste techniek geselecteerd worden.

Dialogs

De dialogs worden gebruikt voor het tonen van de “ja/nee”- of “ok/cancel”-dialogen boxen. De dialogs worden gebruikt om te bevestigen dat de gebruiker bewust een keuze maakt.

File transfer

Wanneer de inspecteur een foto heeft gemaakt moet deze naar de back-end server gestuurd worden. Om dit te kunnen doen wordt de file transfer protocol gebruikt.

File

Wanneer de batterij “kritiek” wordt en de interne gegevens opgeslagen moeten worden op het file system, wordt gebruik gemaakt van het file protocol.

Geolocation

Aan de hand van de locatie van de inspecteur kan automatisch een adres gekoppeld worden. Met de geolocation is de positie van de inspecteur te bepalen.

Globalisatie

Afhankelijk van het land waar de inspecteur zich in bevindt, zijn verschillende datumformats en geld tekens te bedenken. Met de globalisatie wordt dit zelf afgehandeld.

InAppBrowser

Met de InAppBrowser kunnen externe webpagina's worden getoond binnen de applicatie. Tevens maakt iOS gebruik van de InAppBrowser om PDF's te bekijken.

Network information/toegang

Met network information kan opgevraagd worden wat de status van het netwerk is. Wanneer geen 3G,4G of Wi-Fi beschikbaar is moet de applicatie alle wijzigingen bufferen totdat de applicatie weer gesynchroniseerd kan worden met de server.

Touch screen

De applicatie moet reageren wanneer het scherm van de mobiel wordt aangeraakt.

Bijlage 7. Afbeeldingen Lunch applicatie Ionic

Selecteer gebruiker
<input type="text" value="Search"/>
Tom
Lars
Frank

Ionic 1 Selecteer en voeg gebruiker toe scherm

Opties voor Lars op Maandag
2,5
Krediet toevoegen
Lunch deelnamen
Boodschappen betalen Opties

Ionic 3 Gebruikers opties

Lunch 23-03-2015, Aantal: 2	
<input type="text" value="Search"/>	
Maandag	
Lars , €12	<input checked="" type="checkbox"/>
Tom , €17.5	<input checked="" type="checkbox"/>
Frank , €1.5	<input type="checkbox"/>
<div> <div>Lunchpot: €5.0</div> <div> <div>Avr daguitgave: €5.0</div> <div>Avr weekuitgave: €5.0</div> </div> </div> <div>+</div>	

Ionic 2 Lunch applicatie dag scherm

Week 13,2015

Search

Maandag	Dinsdag	Woensdag	Donderdag	Vrijdag
Lars , €12 <input checked="" type="checkbox"/>	Lars , €12 <input type="checkbox"/>	Lars , €12 <input checked="" type="checkbox"/>	Lars , €12 <input type="checkbox"/>	Tom , €17.5 <input checked="" type="checkbox"/>
Tom , €17.5 <input checked="" type="checkbox"/>	Tom , €17.5 <input checked="" type="checkbox"/>	Tom , €17.5 <input type="checkbox"/>	Tom , €17.5 <input checked="" type="checkbox"/>	Lars , €12 <input checked="" type="checkbox"/>
Frank , €1.5 <input checked="" type="checkbox"/>	Frank , €1.5 <input type="checkbox"/>	Frank , €1.5 <input checked="" type="checkbox"/>	Frank , €1.5 <input checked="" type="checkbox"/>	Frank , €1.5 <input checked="" type="checkbox"/>

Lunchpot: €5.0	Avr daguitgave: €4.0 Avr weekuitgave: €5.0	Lunchpot: €5.0	Avr daguitgave: €6.0 Avr weekuitgave: €5.0	Lunchpot: €5.0	Avr daguitgave: €5.0 Avr weekuitgave: €5.0	Lunchpot: €5.0	Avr daguitgave: €8.0 Avr weekuitgave: €5.0	Lunchpot: €5.0	Avr daguitgave: €2.0 Avr weekuitgave: €5.0
+		+		+		+		+	

Ionic 4 Lunch applicatie week scherm

Bijlage 8. Afbeeldingen Lunch applicatie Intel XDK

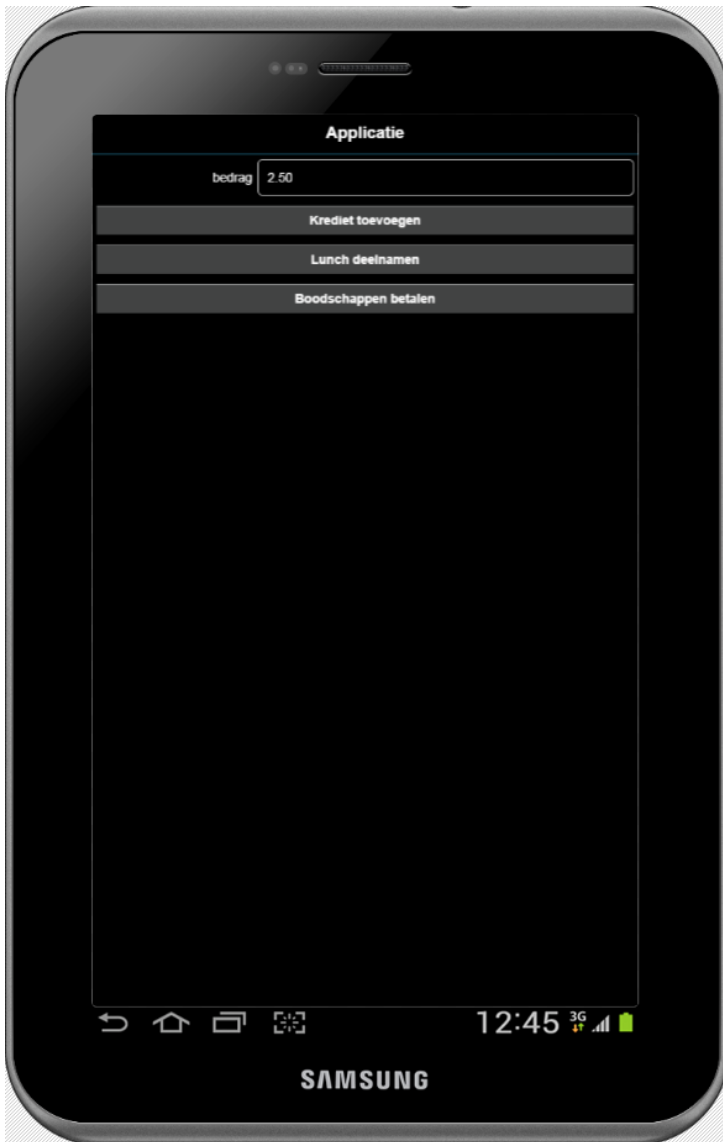


Intel XDK 1 Begin scherm



Intel XDK 2 Selecteer gebruiker

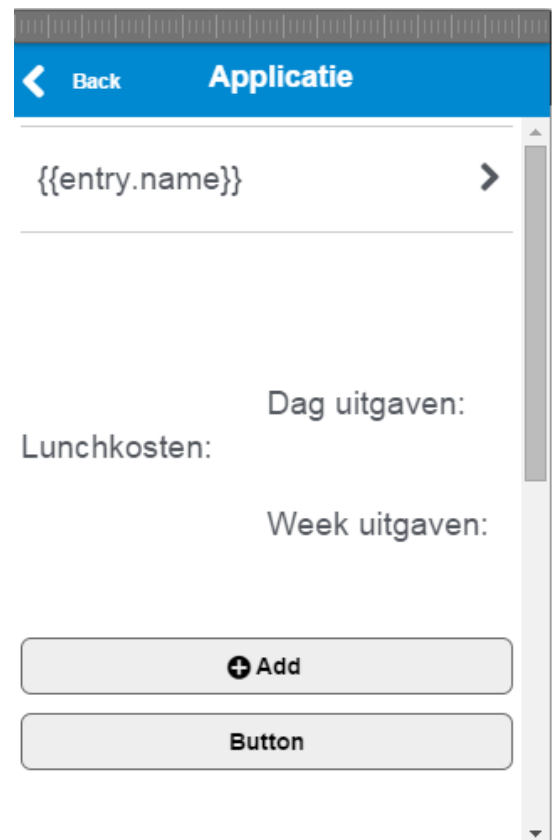
Multiplatform front-end development met tools/frameworks



Intel XDK 4 Gebruikers opties

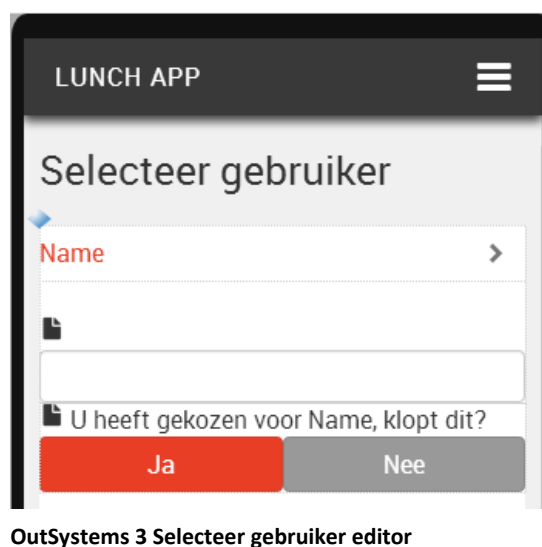
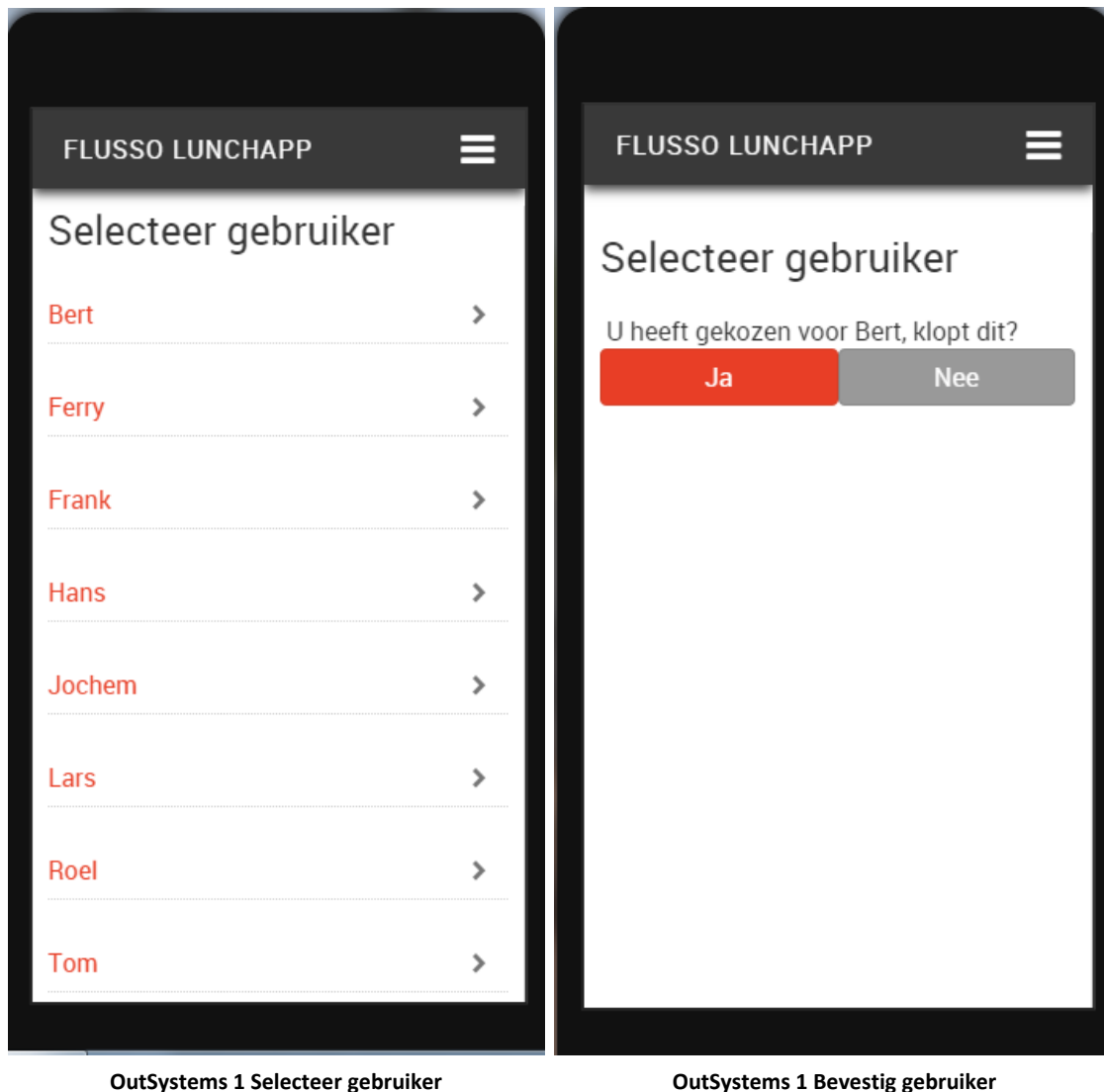


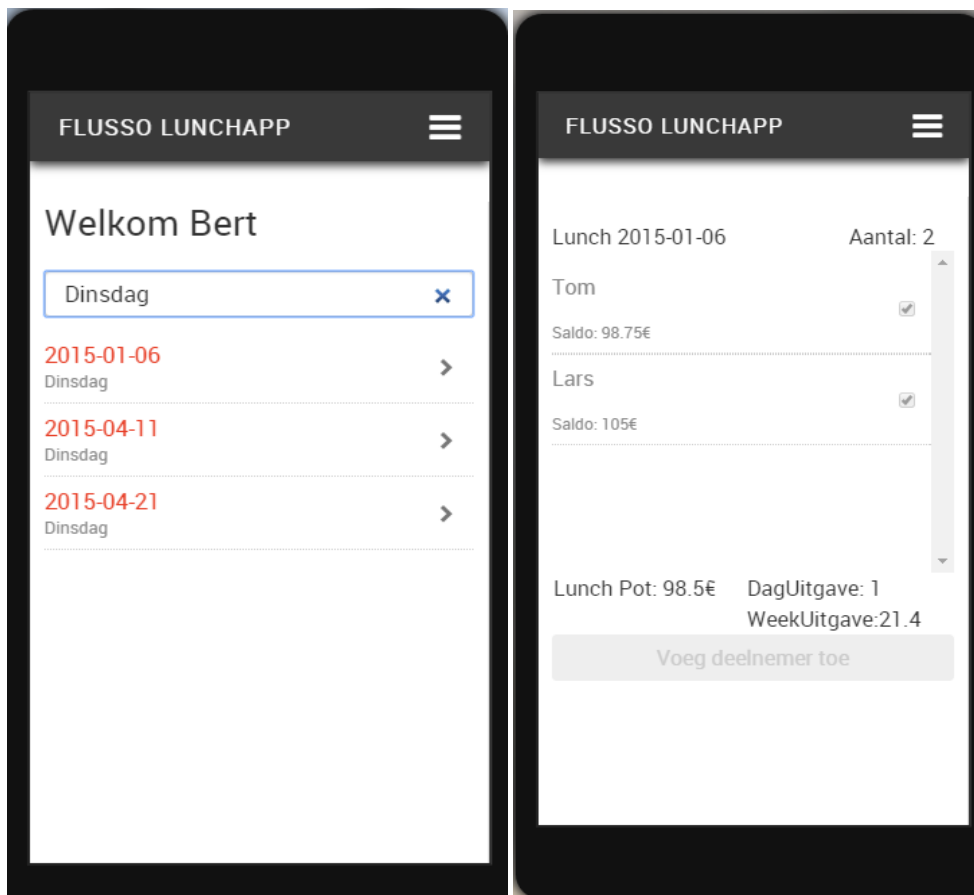
Intel XDK 3 Lunch applicatie



Intel XDK 5 Lunch applicatie editor

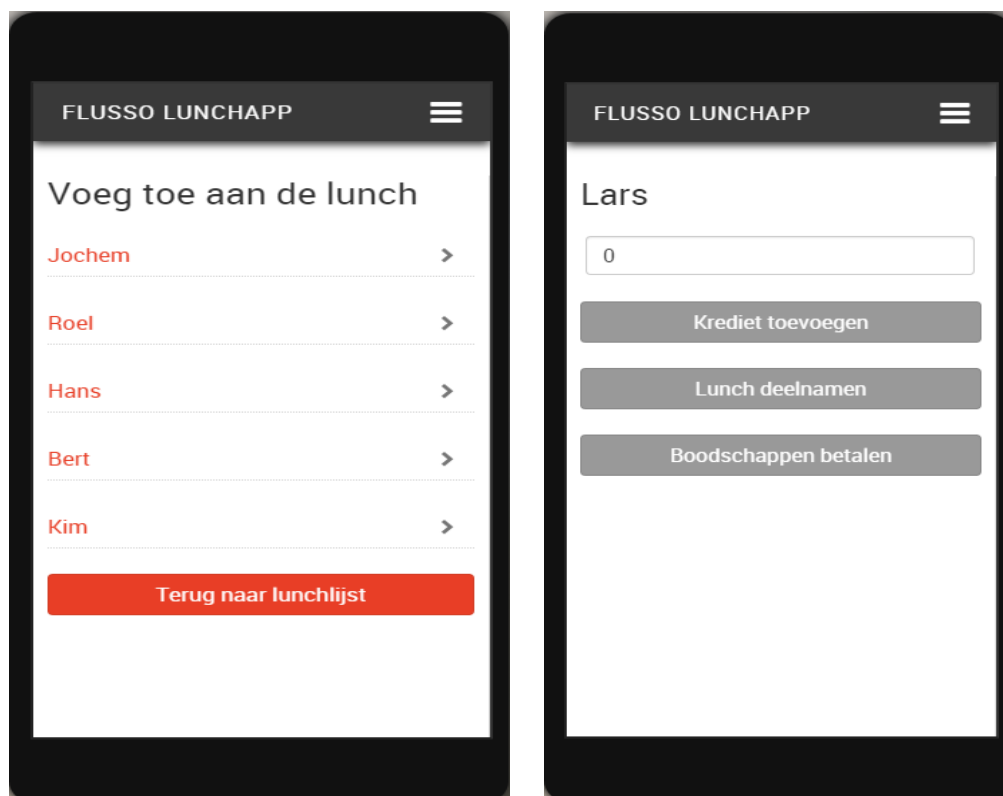
Bijlage 9. Afbeeldingen Lunch applicatie OutSystems





OutSystems 2 Dag/week/maand overzicht

OutSystems 3 Lunch applicatie



OutSystems 6 Voeg gebruiker toe scherm

OutSystems 7 Gebruikers opties