

MOBIELE APPLICATIES, NATIVE OF FRAMEWORKED?

Een toegepast onderzoek naar de meest geschikte aanpak voor bedrijven

Framework Picker

Languages known

☒ C#
☒ C++
☒ Java
☒ HTML+JavaScript+CSS

API's required

☒ Accelerometer
☒ Bluetooth
☐ Camera
☐ Compass
☐ Geolocation
☐ Vibration
☐ Alerts
☐ Calendar
☐ Contacts
☐ Media
☐ File
☐ Notification
☐ Push

Extras

☐ Show free frameworks only
☐ Show open source frameworks only
☐ Show untested frameworks as well (these have been found unstable or not fit for purpose).

Name	Open source	Free	Free for	Code	Markup	Notes	Performance	Look	Download
MeSync			Everything	JavaScript C++	HTML+CSS Code				
Native Android			Everything	Java	XML				
Native Windows Phone			Everything	C#	XAML				
Xamarin			Small projects	C#	XML XAML Code	Free version is limited to 32k of compiled user code (IL), and which do not call out to native third party libraries (i.e., developers may not P/Invoke into C/C++/Objective-C/Java).			

©Jeroen Heijster 2013

Jeroen Heijster
Versie 1.2
11-7-2013

VOORWOORD

Deze scriptie is geschreven door Jeroen Heijster. Op het moment van schrijven, een 24 jarige informatica student aan de Hogeschool Rotterdam. De keuze voor het onderwerp, wanneer je voor een bepaalde framework kiest en wanneer native, is tot stand gekomen met het bedrijf Lunatech. Mobiele apps zijn ook een gebied waar ik zelf in geïnteresseerd ben.

In het onderzoek ben ik bezig geweest om de frameworks zo goed mogelijk te analyseren en om het voor de lezer duidelijk te maken welke het beste kunnen worden gebruikt. Niet elk framework werkte zoals zij zelf aangeven en anderen werkten beter dan verwacht. Uiteindelijk ben ik tot een conclusie gekomen waar ik blij mee ben. Ik denk dan ook dat anderen hier veel aan kunnen hebben.

Ik wil voor dit onderzoek de medewerkers van Lunatech bedanken. Voornamelijk Francisco José Canedo Dominguez. Mijn proeflezers, Sebastiaan Dammann, Yasmijn Kok en Wolfert de Kraker, hebben mij ook heel erg geholpen.

Verder wil ik ook mijn eerste en tweede docenten bedanken, Hossein Chamani en Stelian Paraschiv.

Ik wil ook de bedrijven Xamarin en Syncfusion bedanken voor het gratis beschikbaar stellen van hun producten Xamarin en Orubase. Zonder deze gratis licenties had ik niet hun producten kunnen onderzoeken.

Tot slot wil ik u, de lezer, uitnodigen om de scriptie te lezen.

Jeroen Heijster

Rotterdam, vrijdag 11-07-2013

<http://www.JeroenHeijster.com>

SAMENVATTING

Het onderzoek naar native apps en het gebruik van frameworks is gedaan bij Lunatech te Rotterdam. Lunatech is een bedrijf dat zich bezig houdt met het ontwikkelen van applicaties. Dit kunnen mobiele applicaties, web applicaties of desktop applicaties zijn.

Deze scriptie gaat over het ontwikkelen voor Windows Phone 8 en Android. Er is een onderzoek gedaan naar de volgende frameworks: Codename One, Cordova, Enyo, jQuery, Marmalade, MoSync, Orubase, Rhodes, Sencha Touch en Xamarin. Dit is een mix van web-frameworks en frameworks die als native apps draaien. De keuze van deze frameworks is gedaan op basis van welke frameworks zowel Android als Windows Phone ondersteunen. Naast het ontwikkelen in de frameworks is er ook gekeken naar het ontwikkelen in de native frameworks Silverlight en Android Application Framework.

De opdracht

Lunatech kiest de technologieën voor het ontwikkelen van softwareproducten op basis van de requirements, de beschikbare tijd en het budget. Daarom wil het bedrijf voor elk project kunnen bepalen wat de meest geschikte technologieën zijn. De hoofdvraag is:

Wanneer is het handiger om voor Windows Phone en Android in een multi-platform framework te ontwikkelen en wanneer kan dit beter native?

De deelvragen zijn:

- Welke frameworks komen in aanmerking?
- Hoe is de performance van de frameworks?
- Hoeveel code kan er hergebruikt worden bij het native ontwikkelen?
- Hoe goed lijken de apps die door middel van de frameworks worden gemaakt op native apps?
- Welke problemen komen er naar voren bij het renderen in Cordova tussen de platformen?
- Wat hebben de native talen overeen en waar verschilt het?
- Wat zijn de voor- en nadelen van native en de frameworks?
- Hoe kan er een keuzemodel worden gemaakt om te bepalen hoe/met wat er moet worden ontwikkeld?

Meer over de deelvragen staat ook in het hoofdstuk Methoden en materialen

Methoden en materiaal

Er is onderzoek gedaan aan de hand van experimenten, enquêtes, action research en documentonderzoek. Meer hierover in het hoofdstuk Onderzoeksmethodologie.

Bij het onderzoek zijn de volgende apparaten gebruikt: Nokia Lumia 800, Samsung Galaxy S IV en de Nokia Lumia 820.

Resultaten

Aan de hand van de resultaten is er een keuzemodel gemaakt om bedrijven te helpen welke frameworks kunnen worden gebruikt. Bij het onderzoek is rekening gehouden met de talen en features van de frameworks.

Conclusie en aanbevelingen

Het gebruik van een multi-platform framework of de native frameworks hangt af van de eisen van het project. Bij een groot project is een multi-platform framework handiger.

SUMMARY

The research about native apps and frameworks was done at Lunatech in Rotterdam. Lunatech is a company that specializes in developing applications. This can be mobile, web or desktop.

This paper is about developing for Windows Phone 8 and Android. Research has been done regarding the frameworks: Codename One, Cordova, Enyo, jQuery, Marmalade, MoSync, Orubase, Rhodes, Sencha Touch en Xamarin. These are a mix of web-frameworks and frameworks which compile to native code. The choice was done on the basis of which frameworks supported both Android and Windows Phone. The native frameworks Silverlight and Android Application Framework have also been researched.

The assignment

Lunatech chooses the technologies based on the requirements, available time and the budget. That is why the company wants to choose the right technologies for the each project. The main question is:

When is it better to develop for Windows Phone and Android in a multi-platform framework and when is it better to do this native?

The sub-questions are:

- Which frameworks are eligible?
- How good do the frameworks perform?
- How much code can be reused when developing natively?
- How well do the apps look like native apps when they are made in a framework?
- Which problems arise when rendering Cordova web-apps on different platforms?
- In what ways are the native languages the same and in what way are they different?
- What are the advantages and disadvantages of native and the frameworks?
- How can a selection model be created to choose the technologies?

Methods and materials

During the research, the following devices were used: Nokia Lumia 800, Samsung Galaxy S IV and the Nokia Lumia 820.

Research has been done with experiments, questionnaires, action research en document research.

Results

Based on the results, a selection model was made to help companies choose which frameworks can be used. This takes into account which programming languages and features are necessary.

Conclusion

The choice between a native framework and a multi-platform framework depends on the scope and size of the project. The bigger the project is, the more a multi-platform framework is recommended.

Inhoud

VOORWOORD	2
SAMENVATTING	3
DE OPDRACHT	3
METHODEN EN MATERIAAL	3
RESULTATEN.....	3
CONCLUSIE EN AANBEVELINGEN	3
SUMMARY	4
THE ASSIGNMENT	4
METHODS AND MATERIALS.....	4
RESULTS	4
CONCLUSION	4
INTRODUCTIE	7
ACHTERGROND EN AANLEIDING VOOR OPDRACHT	7
PROBLEEMSTELLING	7
SCRIPTIE	7
DOELSTELLING	8
BELANG VAN EEN NATIVE LOOK & FEEL	8
ONDERZOEKSVRAGEN.....	8
SCOPE	9
BEDRIJF.....	10
ONTWIKKELAARS.....	10
COMPETITIE	10
METHODEN EN MATERIALEN.....	11
DE APP	11
ONDERZOEKSMETHODOLOGIE	17
BESCHIKBARE FRAMEWORKS	21
PLATFORMEN	24
PRIJS.....	25
TALEN	26
CONCLUSIE.....	27
LOOK & FEEL	28

HET BELANG VAN EEN NATIVE LOOK & FEEL	28
DE HUIDIGE STAND VAN DE FRAMEWORKS	29
CONCLUSIE.....	32
PERFORMANCE.....	33
STARTUP.....	35
LOGIN AFHANDELING	37
BEREKENING.....	39
CONCLUSIE.....	41
CODE HERBRUIKBAARHEID	42
CORDOVA PORT	46
CONCLUSIE.....	46
KEUZEMODEL	47
FLOWCHART.....	47
SCREENSHOTS.....	48
CONCLUSIE EN AANBEVELINGEN	50
LOOK	50
CODE	50
CONCLUSIE.....	50
DISCUSSIE	51
DUBLIN-DESCRIPTOREN	51
VERKLARENDE WOORDENLIJST	53
VERWIJZINGEN	57
BIJLAGE A: MOBIELE REKENTOOLS.....	63
BIJLAGE B: KWALITEITSEISEN	64
BIJLAGE C: PREPROCESSOR DIRECTIVES	65
BIJLAGE D: OVEREENKOMSTEN EN VERSCHILLEN TUSSEN C# EN JAVA	66
BIJLAGE E: GEBRUIKTE TESTAPPARATEN.....	72
BIJLAGE F: WINDOWS PHONE BACK BUTTON	73
BIJLAGE G: MEETRESULTATEN	74

Introductie

Achtergrond en aanleiding voor opdracht

Lunatech is een bedrijf dat zich bezighoudt met het ontwikkelen van applicaties. Lunatech programmeert voor klanten zoals Sdu Uitgevers en UPS. Binnen het bedrijf worden vooral open standaarden en open software gebruikt zoals open frameworks en open software. Recent is er ondersteuning voor iOS bijgekomen. Dit betekent dat de klanten nu naast desktop applicaties, websites en Android ook voor iOS kunnen kiezen. In dit traject is er gewerkt met Windows Phone en Android.

Binnen Lunatech zijn er verschillende rekentools gemaakt voor de klant Sdu Uitgevers. Dit zijn applicaties om bijvoorbeeld de contante waarde te berekenen. De contante waarde is de huidige waarde van een bedrag waarover je pas na een bepaalde periode de beschikking hebt. Er wordt rekening gehouden met de tijdsperiode en de rente die misgelopen wordt. [1]

De apps zijn destijds voor iOS ontwikkeld in het web framework Cordova, ook bekend als PhoneGap, en later was dit overgezet naar Android. Op het moment dat de applicaties naar Android werden overgezet, kwamen er enkele problemen naar voren met hoe de pagina wordt weergegeven.

Voor het onderzoek is de “Contante Waarde” app¹, overgezet naar Windows Phone en Android in hun native talen. Voor Windows Phone is dit Silverlight en voor Android is dit Java. De bestaande Cordova versie is geschikt gemaakt voor Windows Phone. Verder zijn er Android en Windows Phone prototypes gemaakt in de gekozen frameworks.

De aanleiding van deze opdracht is de gemeenschappelijke vraag wanneer er beter een framework kan worden gebruikt en wanneer er beter native kan worden ontwikkeld. Beide soorten aanpakken hebben namelijk hun voor- en nadelen. Deze zijn in kaart gebracht.

Probleemstelling

Het doel bij het maken van een product is “hoe kan er zoveel mogelijk waarde creëren met zo min mogelijk kosten”. Het is belangrijk om duidelijk te hebben wanneer een bepaalde manier van werken beter is dan een andere manier.

Een foute inschatting kan tijd en dus geld kosten. Zo ook bij het ontwikkelen van de applicaties waar het onderzoek op gebaseerd is, de mobiele rekentools (zie voor meer informatie Bijlage A: Mobiele Rekentools). Als een app in een verkeerd framework wordt gemaakt, dan kan het ontwikkelen langer duren. Dit kost dan ook meer geld. Buiten dat het langer duurt, kan dit ook frustratie bij de klant en programmeur opleveren.

Scriptie

Deze scriptie is ook online beschikbaar. Door de snelle ontwikkelingen kan het zijn dat deze versie niet meer up-to-date is. De laatste versie zou altijd beschikbaar moeten zijn op:

<https://github.com/MJHeijster/Scriptie/>

<http://www.jeroenheijster.com/scriptie/>

¹ Zie voor meer informatie Bijlage A: Mobiele Rekentools

Doelstelling

Het doel van de opdracht is het onderzoeken wanneer er beter native kan worden geschreven en wanneer er beter in een framework kan worden geprogrammeerd. Het is de bedoeling om een keuzemodel te maken om dit makkelijker te bepalen.

Voor het onderzoek zijn native Android en Windows Phone versies van de Contante Waarde app geschreven. Lunatech heeft belang bij de code van de Windows Phone applicatie. De kennis hierover is nog niet in-house. Als er vragen komen over Windows Phone, dan willen zij daarmee aan de slag kunnen.

Het schrijven van code die te begrijpen is voor de Lunatech programmeurs is het belangrijkste. Op die manier verwacht Lunatech de code voor Windows Phone te begrijpen en op die manier ook aan de slag te kunnen met Windows Phone.

Belang van een native look & feel

Bij het gebruik van de framework moet het binnen dit onderzoek zo zijn dat, of de look & feel is als een native app, of dat het aan te passen is binnen het framework. Dit om een app te laten passen binnen het framework. Het is immers de bedoeling dat een gebruiker zich thuis voelt in de app. Meer hierover in het hoofdstuk Look & Feel.

Onderzoeksvragen

De hoofdvraag is:

**WANNEER IS HET HANDIGER OM VOOR WINDOWS PHONE EN ANDROID IN EEN FRAMEWORK TE
ONTWIKKELEN EN WANNEER KAN DIT BETER NATIVE.**

Bij dit onderzoek zullen verschillende aspecten worden onderzocht. De onderzoeksvragen zijn:

- Welke frameworks komen in aanmerking?
- Hoe is de performance van de frameworks?
- Hoeveel code kan er hergebruikt worden bij het native ontwikkelen?
- Hoe goed lijken de apps die door middel van de frameworks worden gemaakt op native apps?
- Welke problemen komen er naar voren bij het renderen in Cordova tussen de platformen?
- Wat hebben de native talen overeen en waar verschilt het?
- Wat zijn de voor- en nadelen van native en de frameworks?
- Hoe kan er een keuzemodel worden gemaakt om te bepalen hoe/met wat er moet worden ontwikkeld?

Scope

Bij het zoeken naar de frameworks zijn de volgende criteria opgesteld:

- De gebruikte programmeertaal is een van de volgende:
 - Java
 - C#
 - HTML + CSS + JavaScript
- Het framework is geschikt voor Android en Windows Phone.
- Het ontwikkelen moet kunnen op Windows 8
- De demo app (meestal kitchensink) moet, zonder aanpassingen, werken
- De documentatie moet aan de volgende eisen voldoen:
 - Binnen een half uur moet duidelijk zijn voor mijzelf:
 - Hoe kan er genavigeerd worden tussen pagina's/views
 - Welke API's zijn beschikbaar
 - De documentatie is beschikbaar in het Engels en/of Nederlands
 - De documentatie is up-to-date met de laatste release versie

De criteria voor de programmeertalen zijn opgesteld in overleg met Lunatech en dit zijn de talen die bij de onderzoeker bekend zijn. De afbakening van Android en Windows Phone is gebeurd voor het begin van het onderzoek. Door Windows Phone mee te nemen in dit onderzoek, is het niet mogelijk de ontwikkeling te doen in Linux of Mac. Het ontwikkelen voor Windows Phone 8 moet op een 64-bit Windows 8 computer. [2]

De reden waarom het andere grote platform, iOS, niet werd meegenomen komt door overleg met Lunatech. iOS is bij het bedrijf al bekend en hier programmeren zij al voor. Hierdoor is het platform minder relevant voor hun onderzoek. De kennis die zij op willen doen is Windows Phone. Hoewel Android bij Lunatech ook bekend is, moest er een extra platform zijn om voor te ontwikkelen en om de tests van de multi-platform framework op uit te voeren.

De eis dat de demo app zonder aanpassingen moet werken is omdat dit de volwassenheid van het framework moet aantonen en hoe het beste geprogrammeerd kan worden voor het specifieke framework. Indien dit niet in hun eigen app werkt, kan een developer er niet van op aan dat het framework dan voor hem/haar wel werkt.

De frameworks die bestaan, zijn geanalyseerd en gefilterd op programmeertaal, of het ontwikkelen kan op Windows 8 en of het framework voor Android en Windows Phone werkt.

Voor de geselecteerde frameworks zijn prototypes gemaakt om de hoofd- en deelvragen te beantwoorden. De prototypes voldoen aan de kwaliteitseisen van Bijlage B: Kwaliteitseisen.

Binnen dit onderzoek is niet gekeken naar de mogelijkheden voor games. Er is enkel gekeken naar het ontwikkelen van data-centrische apps.

Bedrijf

Lunatech is opgericht in 1993 als een IT consulting, product onderzoek en ontwikkel team. Zij staat onder leiding van Peter Hilton, Nicolas Leroux en Michael Pentowski. Alle medewerkers, op Michael Pentowski na, zijn programmeurs waarvan sommigen ook management taken zoals projectleider op zich nemen.

Lunatech is een bedrijf dat qua organisatie erg plat is. Het bedrijf verdient geld aan het ontwikkelen van applicaties voor klanten waaronder:



Ontwikkelaars

De ontwikkelaars zijn voor de klanten het rechtstreekse aanspreekpunt. Er zijn geen vaste projectleiders, dit verschilt binnen Lunatech per project. Dit betekent dat elke developer grote verantwoordelijkheid heeft over zijn/haar project. Dankzij de brede kennis zijn de medewerkers breed inzetbaar.

Competitie

Enkele concurrenten in regio Rotterdam zijn:



Lunatech onderscheidt zich door geen toegewijde verkoopmedewerkers te hebben. Hierdoor zijn er lagere kosten en is het technisch personeel vanaf het begin al betrokken om met een passend aanbod te komen.

Methoden en materialen

Het onderzoek is opgezet vanuit het perspectief van een programmeur. In dit onderzoek is ook geprobeerd om voor het management helder te krijgen welk framework binnen een project past. De auteur van dit onderzoek is van origine een C# en Java ontwikkelaar. Dit betekent dat in het onderzoek voornamelijk is gekeken naar deze twee talen.

De app

De contante waarde is de huidige waarde van een bedrag waarover je pas na een bepaalde periode de beschikking hebt. Er wordt rekening gehouden met de tijdsperiode en de rente die misgelopen word. [1]

Voor het onderzoek is de “Contante Waarde” app², overgezet naar Windows Phone en Android in hun native talen. Voor Windows Phone is dit Silverlight en voor Android is dit Java. De bestaande Cordova versie is geschikt gemaakt voor Windows Phone. Verder zijn er Android en Windows Phone prototypes gemaakt in de gekozen frameworks.

Formule

Voor het berekenen van de contante waarde kan gebruik worden gemaakt van de volgende formule:

$$HW = \frac{TW}{(1 + i)^t}$$

Hierbij staan de volgende letters voor:

- HW = huidige waarde
- TW = Toekomstige waarde
- t = periode in jaren
- i = rentevoet in percentage

Requirements

- De Android app werkt vanaf API 8 (Android 2.2) tot met als target API 17 (Android 4.2). Dit is op basis van het aantal gebruikers, maar 2.4% van de gebruikers gebruikt API 7 of lager op 26-2-2013. Op 10-7-2013 was het nog maar 1.5%. [3]
- De Windows Phone app zal worden gebouwd voor Windows Phone 8. Windows Phone 7.x wordt op het moment uit gefaseerd en het aantal Windows Phone 8 gebruikers stijgt snel ten opzichte van Windows Phone 7.x. In februari was 26% van de Windows Phone toestellen een Windows Phone 8 apparaat en in juni was dit al 58%. [4] [5]
- De applicatie berekent de Contante waarde en toont de resultaten in een apart scherm
- De applicatie heeft een specificatie scherm waarbij voor elke periode de contante waarde word getoond.
- De gebruiker moet inloggen met een gebruikersnaam en wachtwoord.
- De gebruikersnaam en wachtwoord worden geverifieerd met de SDU API.
- De gebruiker heeft een mogelijkheid zijn/haar wachtwoord te resetten via de SDU pagina.

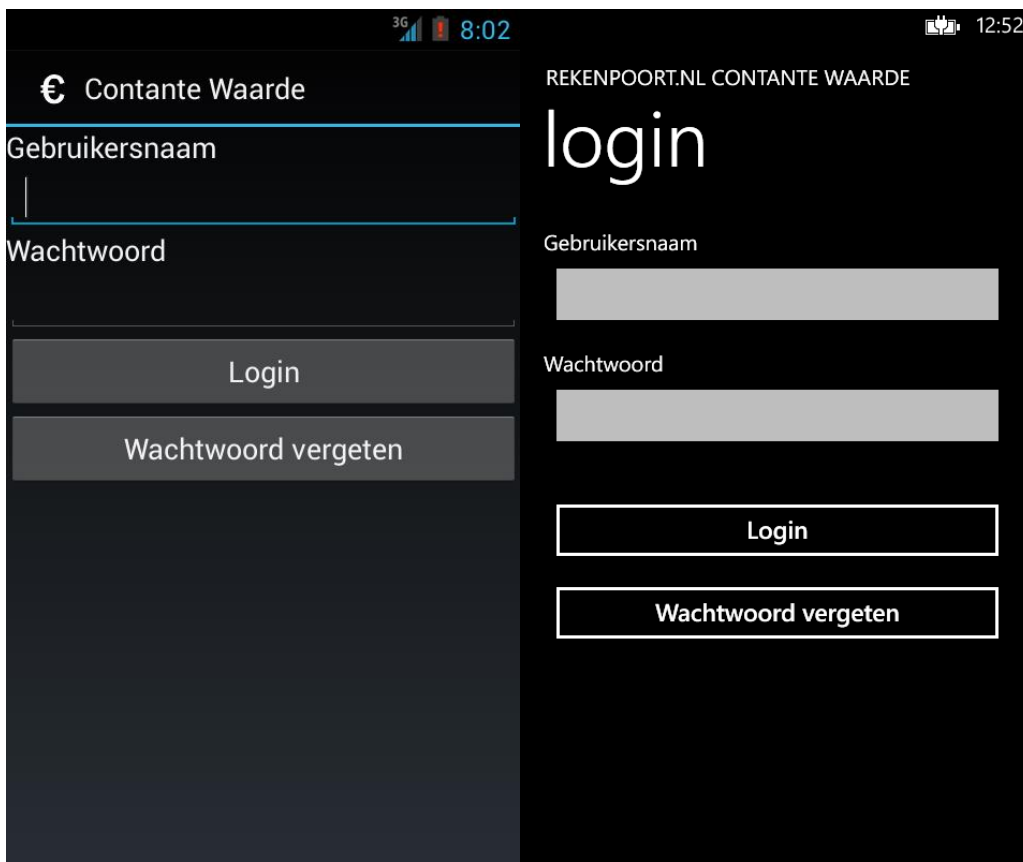
² Zie voor meer informatie Bijlage A: Mobiele Rekentools

Algemene schermen

Login

De app heeft een authenticatie mogelijkheid. Deze vraagt een gebruikersnaam en wachtwoord van de gebruiker. Aan de hand hiervan zal het zich authenticeren tegen een API van een derde partij. Indien de gebruiker zijn/haar wachtwoord niet meer weet, kan de gebruiker doorgestuurd worden naar de webpagina om het wachtwoord te resetten.

Indien de gebruiker de juiste gebruikersnaam en wachtwoord invoert, gaat hij/zij door naar het volgende scherm. Indien deze onjuist zijn krijgt de gebruiker een melding dat de gebruikersnaam en/of wachtwoord onjuist zijn.



FIGUUR 1

FIGUUR 2

Zoals in Figuur 1 en 2 te zien is, is hier het login scherm. De gebruiker voert hier zijn/haar gebruikersnaam en wachtwoord in en klikt op login. Indien de gebruiker niet weet wat zijn/haar gebruikersnaam en/of wachtwoord is, klikt de gebruiker op wachtwoord vergeten. In de webbrowser wordt de pagina geopend waar dit op te vragen is.

Invoerscherm

Voor een berekening zijn verschillende waardes nodig. Deze staan ook kort beschreven onder het kopje Formule. De originele requirements stelden dat de termijnduur in jaren, kwartalen en maanden kan worden ingevoerd. Ook moest er een mogelijkheid zijn om de valuta in guldens in te voeren.

The image shows two screenshots of a mobile application interface for calculating the present value of an annuity. The left screenshot, labeled FIGUUR 3, shows a dark-themed interface with a sidebar menu on the left. The sidebar has a header '€ Contante Waarde' and a list of options: 'Berekenen', 'Bedrag', 'Termijnen', 'Termijnduur', 'Rente', and 'Valuta'. The 'Berekenen' option is selected. The main area shows a large 'Bereken' button. The right screenshot, labeled FIGUUR 4, shows the same interface but with the input fields filled in. The 'Bereken' button is now a light gray, indicating it is disabled. The input fields are: 'Bedrag' (empty), 'Termijnen' (empty), 'Termijnduur' (set to 'Jaar'), 'Rente' (empty), and 'Valuta' (set to 'Euro'). The top status bar shows '3G' and '9:04' on the left, and '12:53' on the right.

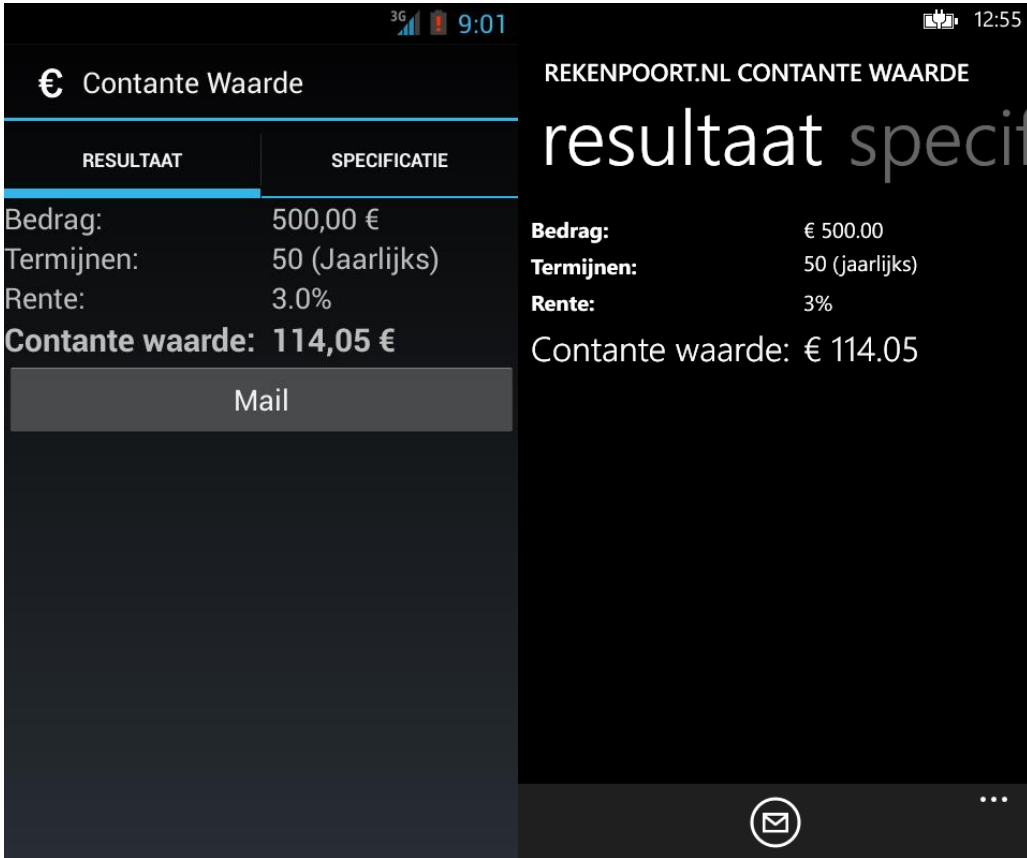
FIGUUR 3

FIGUUR 4

Zoals in figuur 3 en 4 te zien is, zijn er een aantal invoervelden voor de gebruiker. De gebruiker voert het bedrag in, de termijnen in, selecteert de termijnduur (maand, kwartaal of jaar), voert de rente in en selecteert vervolgens de valuta (euro of guldens). Vervolgens klikt de gebruiker op bereken om de invoer af te ronden en door te gaan naar het volgende scherm.

Resultaat

Zodra de gebruiker de waardes heeft ingevoerd en op bereken heeft geklikt, gaat de app door naar het resultaat scherm. Hier kan de gebruiker zien wat hij/zij heeft ingevoerd en wat het resultaat is. De gebruiker heeft ook de mogelijkheid om dit te mailen naar een, voor de gebruiker, nader te bepalen e-mail adres.



FIGUUR 5

FIGUUR 6

Zoals te zien is in Figuur 5 en 6, zijn hier de resultaten. Hierin zien wij:

- Bedrag: het ingevoerde bedrag in het vorige scherm.
- Termijnen: het ingevoerd aantal termijnen in het vorige scherm.
- Rente: het ingevoerde rentepercentage in het vorige scherm.
- Contante waarde: de daadwerkelijke contante waarde, het resultaat van de berekening.

Specificatie

Bij de specificatie ziet de gebruiker per termijn wat hoeveel het bedrag op dat moment waard is.

FIGUUR 7

RESULTAAT	SPECIFICATIE
Termijn 0: 114,05 €	Termijn 0 € 114.05
Termijn 1: 117,48 €	Termijn 1 € 117.48
Termijn 2: 121,00 €	Termijn 2 € 121.00
Termijn 3: 124,63 €	Termijn 3 € 124.63
Termijn 4: 128,37 €	Termijn 4 € 128.37
Termijn 5: 132,22 €	Termijn 5 € 132.22
Termijn 6: 136,19 €	Termijn 6 € 136.19
Termijn 7: 140,27 €	

FIGUUR 8

REKENPOORT.NL CONTANTE WAARDE

specificatie res

Termijn 0
€ 114.05

Termijn 1
€ 117.48

Termijn 2
€ 121.00

Termijn 3
€ 124.63

Termijn 4
€ 128.37

Termijn 5
€ 132.22

Termijn 6
€ 136.19

FIGUUR 7

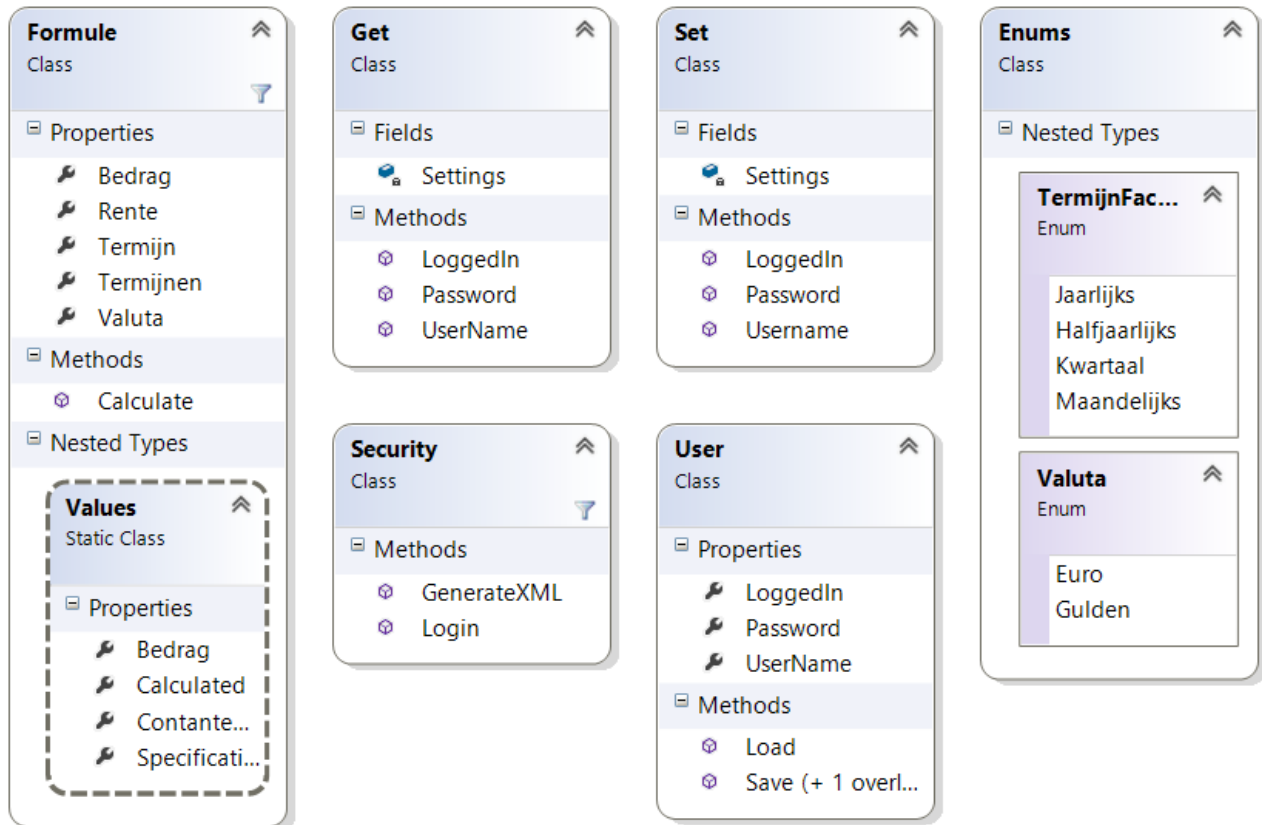
FIGUUR 8

In figuur 7 en 8 zien wij per termijn wat het geld waard is.

Klassendiagram

De opbouw van de apps gebruiken, waar mogelijk, dezelfde opbouw. Deze klassen bestaan in de Xamarin Android versie, native Android versie en de native Windows Phone versie. Indien deze opbouw niet mogelijk was, bijvoorbeeld bij Sencha, is er afgeweken van deze diagram.

In het geval van Sencha zijn er geen klassen mogelijk.



Onderzoeksmethodologie

Voor dit onderzoek is gekeken of er documenten te vinden waren die relevant waren bij mijn onderzoek. Hierbij is een internet onderzoek [6] gedaan. Om betrouwbare bronnen te vinden zijn de volgende databanken gebruikt:

- ACM Digital Library
- Company Info
- Ios Press
- HBO Kennisbank
- LexisNexis

De conclusie was dat er wel onderzoek is gedaan naar multi-platform development, maar dat Windows Phone hier niet in mee wordt genomen. Dit, waarschijnlijk, omdat het platform nog maar enkele jaren bestaat in zijn huidige vorm. Ook is er geen site gevonden waar gemakkelijk kan worden gekozen of een app in een bepaald framework moet worden gemaakt of dat het native moet worden gemaakt. [7]

De selectie van frameworks is gedaan in de week van 25 februari. Tijdens de rest van dit onderzoek is er meerdere malen gekeken of er nieuwe, geschikte, frameworks zijn. Afhankelijk van de tijd is er gekeken of deze nog mee konden worden genomen in het onderzoek.

Voordat het onderzoek begon waren de volgende onderzoeksvragen en onderzoeksmethoden opgesteld:

Welke frameworks komen in aanmerking?

Er is onderzocht welke frameworks er zijn die in zowel Android als Windows Phone werken. Ook is er gekeken naar de mogelijkheid tot het maken van hybride apps.

Methode

- Onderzoek welke frameworks bestaan
- Leg de platformen, talen en features vast
- Onderzoek of de frameworks voldoen aan de eisen
- Bouw prototypes van de geschikte frameworks
- Bepaal de sterke en zwakke punten

Een alternatieve methodologie was bijvoorbeeld het interviewen van experts. Het probleem is dat dit te veel tijd gaat kosten.

Hoe is de performance van de frameworks?

Door middel van experimenten [8] van de login snelheid, opstartsnelheid en berekensnelheid is er gemeten hoe de performance van native is tegenover het ontwikkelen in de verschillende frameworks. Er is gemeten wat de opstart tijd is, wat de inlogtijd is en hoelang het duurt om een berekening uit te voeren.

De originele hypothese stelde dat de frameworks trager zijn. Native apps zijn immers gecompileerd voor het specifieke platform en hebben rechtstreeks toegang tot de API's van het systeem.

In verband met de snelle vernieuwingen in het veld en het nog niet beschikbaar zijn van betrouwbare metingen voor Windows Phone 8 met deze frameworks is ervoor gekozen om de metingen zelf te doen.

Methode

Bij het meten van de snelheid zijn de tests 5 keer herhaald. Dit is gedaan om een betrouwbaar gemiddelde te krijgen en om de consistentie van de tests te waarborgen. Een beschrijving van de test staat in het

hoofdstuk Performance. De tests zijn elke keer op dezelfde telefoon uitgerold en getest. Zie ook Bijlage E: Gebruikte testapparaten.

De betreffende stappen waren:

- Bouw de apps in het framework
- Test de startup snelheid, login snelheid en bereken snelheid
- Noteer wat opvalt

Hierbij zijn ook tests gedaan door derden. Deze hebben alle apps getest en zijn ondervraagd over de eerste indruk, wat ze van de snelheid vonden en of er problemen waren tijdens het gebruik.

Hoeveel code kan er hergebruikt worden bij het native ontwikkelen?

Bij het onderzoek naar hoeveel code hergebruikt kan worden was eerst gekeken naar de overeenkomsten en verschillen van de twee platformen, Android en Windows Phone. Voordat dit onderzoek begon, werden er een native Windows Phone en een native Android applicatie gemaakt. Op basis van de native clients is er een nieuwe client gemaakt voor het andere platform. De native applicaties hebben dezelfde klassenstructuur.

Tijdens mijn internet onderzoek is niet naar voren gekomen hoeveel code er opnieuw kan worden gebruikt zonder compileer problemen met betrekking tot Android en Windows Phone.

Methode

De experimenten [8] zijn als volgt gedaan:

- Bouw de native applicaties
- Maak een nieuw project voor beiden
- Kopiëer de code van het ene platform naar de andere
- Analyseer de problemen
- Los de problemen op indien mogelijk

Hoe goed lijken de apps die door middel van de frameworks worden gemaakt op native apps?

Door middel van experimenten [8] en enquêtes [9] is gekeken of de apps native lijken. Ook zal worden gekeken hoe dit aangepast kan worden als het niet native lijkt. Als een app niet native lijkt, dan betekent het niet dat het framework geschikt is. Het is wellicht mogelijk om die look & feel aan te passen.

Methode

- Test de prototypes op basis van de look
- Zijn er thema's te vinden voor een native look?
- Documenteer de resultaten

De apps zijn vergeleken aan de hand van de design guidelines van de platformen. [10] [11]

Welke problemen komen er naar voren bij het renderen in Cordova tussen de platformen?

In eerste instantie werden er enkele problemen verwacht met het uniform maken van een HTML versie. Lunatech gaf aan dat zij eerder problemen hadden met het overzetten van een iOS Cordova app naar Android, omdat de browser de opmaaktaal anders rendert. Beide browsers gebruiken de WebKit rendering engine, maar toch toont WebKit het anders. Zelfs binnen de verschillende versies van Android schijnt het soms anders te werken. Windows Phone gebruikt zijn eigen engine, Trident, en het plaatsen van objecten en het interpreteren van de stylesheet werkt hierdoor anders.

Methode

- Maak een nieuw Windows Phone 8 Cordova project aan
- Kopieer de Android Cordova bestanden in het Windows Phone 8 project
- Test de applicatie op bugs
- Los de bugs op
- Documenteer de problemen en de oplossingen

Door middel van action research [12] is de Android versie van de Contante Waarde app, zie ook Bijlage A: Mobiele Rekentools, geschikt gemaakt voor Windows Phone. Hierbij werden ook de knelpunten bekend en wat ermee gedaan kon worden. De knelpunten staan beschreven in het hoofdstuk Cordova port.

Wat hebben de native talen gemeen en waar verschilt het?

Bij het onderzoeken naar de native talen voor Windows Phone en Android zal worden gekeken naar de documentatie over de talen om deze uiteindelijk te vergelijken.

Methode

Dit onderzoek is gedaan op basis van internet onderzoek. [6] De basis zijn de documenten op de sites van Microsoft, Oracle en Google vergeleken. Ander onderzoek was niet mogelijk omdat de talen doorontwikkeld worden en deze documenten constant geüpdatet worden. Boeken lopen hierdoor bijvoorbeeld snel achter.

Wat zijn de voor- en nadelen van native en de frameworks?

Bij deze deelvraag zijn bestaande onderzoeken over dit onderwerp onderzocht en gevalideerd.

Methode

In dit onderzoek was door middel van action research [12] en experimenten [8] onderzocht wat de voor- en nadelen zijn van de frameworks en van native apps.

Hierbij heb ik de prototypes en frameworks los van elkaar geanalyseerd op sterke en zwakke punten.

Hoe kan er een keuzemodel worden gemaakt om te bepalen hoe/met wat er moet worden ontwikkeld?

Rekening houdend met de voorgaande onderzoeken zal is er een keuzemodel gemaakt om binnen een paar minuten een keuze te maken tussen native en een specifiek framework. Dit keuzemodel is gebaseerd op het voorafgaande onderzoek.

Methode

Op basis van het voorafgaande onderzoek zijn de API's, de look en de performance beschreven. Tevens is de prijs en of het open-source is gedocumenteerd per framework. Aan de hand hiervan is een keuzemodel gemaakt die de frameworks weergeeft.

Beschikbare Frameworks

Tijdens dit onderzoek waren tien frameworks geselecteerd voor verder onderzoek. De initiële selectie was op basis van of het framework Windows Phone ondersteunde. In eerste instantie zou ook Trigger.io worden meegenomen in het onderzoek [13], achteraf bleek dat de Windows Phone ondersteuning was gestopt.

In dit hoofdstuk worden de frameworks opgesomd met informatie over het framework en de status hiervan. Dit kan betekenen dat er in de status staat dat er in het framework is gebouwd of dat het framework niet geschikt is bevonden.

Codename One

Codename One is een framework dat Java compileert naar Android, iOS, Blackberry en Windows Phone. Dit wordt gecompileerd naar de native taal van het platform.

STATUS

Dit framework is afgefallen omdat het op het moment nog niet volwassen genoeg was. Het compileren van de tab application template app voor Windows Phone duurde twee uur en crashte bij het opstarten op mijn telefoon en in de emulator. Het compileren gaat via de build servers van Codename One. Er is geen manier om het op de ontwikkel pc te compileren voor Android of Windows Phone. In Android werkt het compileren en draaien van de app wel. Het gebrek aan Windows Phone ondersteuning zorgde ervoor dat dit framework niet geschikt is in mijn onderzoek. Bij het rapporteren van dit probleem was de reactie dat het discussieforum had moeten worden geraadpleegd, het zou een “technology preview” betreffen. [14] Hier was echter niets van te vinden op de site of in de issue tracker.

Cordova

Cordova, een framework dat ook bekend staat als PhoneGap, zorgt ervoor dat applicaties gebouwd in HTML+CSS+JavaScript als app kunnen worden gedraaid op iOS, Android, Blackberry, Windows Phone, Palm WebOS, Bada, en Symbian.

Cordova stelt allerlei native API's beschikbaar voor de developer zonder dat dit in de native taal moet worden geschreven.

STATUS

Er zijn prototypes met dit framework gemaakt en het framework is opgenomen in het onderzoek.

Enyo

Enyo is het framework dat origineel was gemaakt voor WebOS. Toen WebOS als open-source project uit werd gebracht, werd ook Enyo op GitHub uitgebracht. De views worden met DOM opgebouwd.

STATUS

Afgevallen aangezien het uitbreiden van het aantal beschikbare widgets te veel tijd kost. Verder werd het niet duidelijk in de documentatie hoe er naar een andere pagina moet worden genavigeerd.

Sencha Touch

Sencha Touch is een framework waarin met HTML+CSS+JavaScript een app kan worden gemaakt. Door middel van HTML of DOM kan de view worden opgebouwd. Het is de bedoeling om MVC te gebruiken.

STATUS

Er zijn prototypes met dit framework gemaakt en het framework is opgenomen in het onderzoek.

jQuery

jQuery is een framework waarin met HTML + CSS + JavaScript een app kan worden gemaakt. In tegenstelling tot Enyo worden de views/pagina's in HTML gemaakt.

STATUS

Er zijn prototypes met dit framework gemaakt en het framework is opgenomen in het onderzoek.

Marmalade

Marmalade is een framework waarin met C++ en/of HTML + CSS + JavaScript een app kan worden gemaakt.

STATUS

Marmalade geeft de mogelijkheid om in HTML + CSS + JavaScript te werken of in C++. De HTML versie van Marmalade (Web Marmalade) werkt op het moment alleen op Android en iOS. Voor Windows Phone was het niet te testen. Dit omdat Windows Phone ondersteuning nog in bèta is en het verkrijgen van een studentenlicentie niet lukt.³ Een licentie is namelijk nodig om de bèta versies te gebruiken. Het is op het moment alleen mogelijk om voor Windows Phone in C++ te ontwikkelen. Het probleem was echter dat de kitchensink demo app in c++ niet wil opstarten in Windows Phone. Om deze redenen is het platform afgevallen.

Mosync

Met MoSync kan in HTML5+CSS+JavaScript of C/C++ een app worden gemaakt. Door middel van MoSync Reload kunnen apps vanaf de pc herladen worden op het apparaat zonder de app steeds te compileren en te installeren. Op het mobiele apparaat installeer je de MoSync Reload client, op de ontwikkel PC MoSync reload. Vervolgens configureer je de client en server. Door middel van de "Reload" button wordt de app dan geopend. Een hindernis bij dit framework is het halen van de waardes uit bijvoorbeeld textboxes. Hiervoor moeten speciale functies worden geschreven. [15] Ook is er geen toegang tot de localStorage voor Windows Phone. [16]

Een andere hindernis is het niet goed ondersteunen van verschillende schermformaten. De lettertype groottes worden in pixels gedefinieerd waardoor de letters groter of kleiner zijn in een andere schermresolutie. [17] Dit is op te lossen door tijdens het renderen van de pagina de grootte van het scherm in pixels uit te rekenen en daarna de tekstgroottes aan te passen door middel van JavaScript.

STATUS

Er zijn prototypes met dit framework gemaakt en het framework is opgenomen in het onderzoek.

³ Op de site van Marmalade: "If you are a student or would like to use Marmalade for educational purposes, please Contact Us to find out how we could help." De aanvraag: ticket 1: Vraag het aan onder je schoolmail, ticket 2: laat het personeel op school het aanvragen. De aanvraag is vervolgens door dhr. Ter Wal gedaan, daarna bleef het stil.

Orubase

Orubase is een framework waarin met ASP.NET MVC de applicaties kunnen worden gemaakt. De applicatie wrapped de webpagina's in de app. De pagina's zijn ASP.NET pagina's. De tabs/pivots worden aangemaakt met de library van Orubase. Hoewel dit framework voornamelijk bedoeld is om met een asp.net backend te werken, is het wel mogelijk om een standalone offline applicatie te maken. [18]

STATUS

Onder andere door verschillende visuele bugs, in de versie die op het moment van schrijven uit is (1.1.0.69), is dit framework niet opgenomen in de tests. De bugs staan beschreven in het hoofdstuk Look & Feel.

Verder kwam in mijn experimenten naar voren dat het niet makkelijk mogelijk is om data van een pagina over te dragen aan een andere pagina. Het mogelijk om een login scherm te implementeren, maar vanuit daar gaat hij door naar de native pagina met tabs. In deze tabs zijn dan webbrowser elementen ingebouwd die niet onderling kunnen communiceren.

RhoMobile

RhoMobile is een framework waarin met Ruby on Rails een app kan worden gemaakt. Hoewel Ruby on Rails niet binnen de scope viel, was het wel interessant om te onderzoeken of dit framework geschikt zou zijn. Dit door persoonlijke interesse van de auteur in de Ruby programmeertaal.

STATUS

Door verschillende problemen is RhoMobile niet opgenomen in het onderzoek.

De IDE voor RhoMobile heeft bijvoorbeeld enkele problemen. De configuratie gaat uit van standaard paden, dit zorgde ervoor dat de IDE uitging van een oude JDK versie. Ook klopte het standaard Android SDK en NDK pad niet.

Toen dit was opgelost en geprobeerd werd een Android app te compileren, bleef de IDE de standaard waardes kiezen voor de keystore. Een van de standaardwaardes was: " No" bij de vraag of de ingevoerde waardes juist waren. Hierdoor bleef het proces van het genereren van een keystore maar doorgaan. Dit probleem is bekend en er is aangeraden om het handmatig te doen de eerste keer. [19]

Wat ook opmerkelijk was is dat de app ongewenst inzoomt. Tevens past de GUI niet bij de platformen.

Xamarin

Xamarin is een framework om apps in C# te bouwen. Er zijn plug-ins voor iOS en Android. Windows Phone werkt via C#, dus hier is geen extra plug-in voor nodig. Met Xamarin ontwikkel je de code in C#. De front-end zal per versie (Android, iOS, Windows Phone) opnieuw geschreven moeten worden. Het voordeel hiervan is dat elke applicatie native aanvoelt en dat de backend code, grotendeels, identiek is. Het nadeel is wel dat het front-end werk meer tijd kost dan bij andere frameworks.

STATUS

Er zijn prototypes met dit framework gemaakt en het framework is opgenomen in het onderzoek.

Platformen

Hoewel dit onderzoek zich heeft gericht op Android en Windows Phone 8, kan het zo zijn dat andere platformen ook meegenomen moeten worden. In deze tabel is te zien welke frameworks kunnen worden gebruikt voor welke platformen.

Framework	iOS	Android	Windows Phone 7	Windows Phone 8	Windows Mobile	BlackBerry
Codename One	v	v	p	p	x	v
Cordova	v	v	v	v	x	v
Enyo	v	v	p	p	x	v
jQuery	v	v	p	p	x	v
Marmalade	v	v	p	p	x	p
Mosync	v	v	v	v	v	v
Orubase	v	v	v	v	x	x
RhoMobile	v	v	v	v	v	v
Sencha Touch	v	v	p	p	x	v
Xamarin	v	v	v	v	x	x
Totaal	10	10	5	5	2	7

TABEL 1

v= volledig ondersteund

p=deels ondersteund⁴

x=niet ondersteund

⁴ Enkele features zijn nog niet af en/of het is nog voor dat platform in bèta.

Prijs

Aangezien prijs een factor is in elk project, is dit meegenomen in het onderzoek. Voor bepaalde bedrijven is het ook belangrijk dat een framework open-source is. Vandaar dat ook deze aspecten zijn meegenomen in het onderzoek.

Frame work	Prijs min	Prijs max	Extra's bij meerprijs	Betalings frequentie	Open-Source	Commercieel gebruik mogelijk met gratis licentie
Coden ame One	\$0,00	\$399,00	1. Meer builds per maand 2. Cloud diensten	Per maand	v	v
Cordova	\$0,00	\$0,00	-	-	v	v
Enyo	\$0,00	\$0,00	-	-	v	v
jQuery	\$0,00	\$0,00	-	-	v	v
Marmalade	\$149,00	\$1.499,00	1. Geen inkomstenlimiet 2. Bèta ondersteuning 3. SLA Support 4. Trainingen	Jaarlijks	x	x
MoSync	€ 0,00	€ 2.999,00	1. Support	Jaarlijks	v	v
Orubase	\$995,00	\$995,00	-	Jaarlijks	x	x
RhoMobile	\$0,00	\$0,00	-	-	v	v
Sencha Touch	\$0,00	\$0,00	-	-	v	v
Xamarin	\$0,00	\$1.899,00	1. Grotere apps 2. Visual Studio ondersteuning 3. Componenten 4. SLA	Per platform per jaar	x	v

TABEL 2

v= volledig ondersteund

x=niet ondersteund

Talen

De verschillende frameworks hebben onderling verschillen qua programmeertalen en opmaaktaal. Op basis van de voorkeuren of vereisten kan het zo zijn dat bepaalde frameworks voor een bedrijf of developer afvallen.

De tabel is in tweeën opgesplitst. De linkerkant geeft aan in welke talen er geprogrammeerd kan worden en de rechterkant geeft aan welke opmaaktalen kunnen worden gebruikt.

Framework	Java	C#	JavaScript	C++	Ruby		XML	XAML	HTML+CSS	DOM	Code
Codename One	v	x	x	x	x		x	x	x	x	v
Cordova	x	x	v	x	x		x	x	v	x	x
Enyo	x	x	v	x	x		x	x	v	v	x
jQuery	x	x	v	x	x		x	x	v	x	x
Marmalade	x	x	v	v	x		x	x	v	x	v
MoSync	x	x	v	v	x		x	x	v	x	v
Orubase	x	v	v	x	x		x	x	v	x	v
Rhodes	x	x	x	x	v		x	x	v	x	x
Sencha Touch	x	x	v	x	x		x	x	v	v	x
Xamarin	x	v	x	x	x		v	v	x	x	v
Totaal	1	2	7	2	1		1	1	8	2	5

TABEL 3

v= volledig ondersteund

x=niet ondersteund

API's

De verschillende frameworks verschillen ook in het aanbod aan hardware/OS API's. Afhankelijk van de applicatie en de behoeften kunnen bepaalde API's nodig zijn in een project.

In Enyo, jQuery en Sencha Touch zijn de API's afhankelijk van de wrapper die gebruikt wordt. De wrapper kan een webbrowser control zijn of Cordova.

API	Codename One	Cordova	Enyo	jQuery	Marmalade	MoSync	Orubas	Rhodes	Sencha Touch	Xamarin	Totaal
Accelerometer	x	v	w	w	v	v	v	v	w	v	6
Bluetooth	x	x	w	w	v	v	x	v	w	v	4
Camera	v	v	w	w	v	v	v	v	w	v	7
Compass	x	v	w	w	v	v	v	x	w	v	5
Geolocation	v	v	w	w	v	v	v	v	w	v	7
Vibration	v	v	w	w	v	v	v	v	w	v	7
Alerts	v	v	w	w	v	v	v	v	w	v	7
Calendar	v	x	w	w	v	x	x	v	w	v	4
Contacts	v	v	w	w	v	v	v	v	w	v	7
Media	v	v	w	w	v	v	x	v	w	v	6
File	v	v	w	w	v	v	x	v	w	v	6
Notification	v	v	w	w	v	v	v	v	w	v	7
Push	v	x	w	w	v	v	v	v	w	v	6

TABEL 4

v= volledig ondersteund

x=niet ondersteund

w= afhankelijk van de wrapper, bij het gebruiken van Cordova zijn er beschikbare API's en bij een web view niet.

Conclusie

Er zijn veel frameworks beschikbaar. Hoewel wellicht niet elk framework geschikt is, is er wel genoeg keuze voor bedrijven en developers. Het is echter niet zo dat alleen de features, talen en de prijs belangrijk zijn, ook de look en performance van de frameworks is belangrijk. Hiervoor kunt u verder lezen.

Look & Feel

Het belang van een native look & feel

Bij het gebruik van de framework moet het binnen dit onderzoek zo zijn dat, of de look & feel is als een native app, of dat het aan te passen is binnen het framework. Dit om een app te laten passen binnen het framework. Het is immers de bedoeling dat een gebruiker zich thuis voelt in de app.

Google

Ook Google schrijft dat het niet handig is om het design van een ander platform te gebruiken. Hiermee riskeer je, aldus Google, dat je de gebruiker zich niet prettig laat voelen. Bij het afwijken van de standaard kan het zo zijn dat de gebruiker in de war raakt. [20]

Android UI Patterns

Android UI Patterns geeft ook een aantal don'ts in app design. De auteur stelt dat hey originele Holo design de fundatie is van jouw app. Vanuit hier kan je uitbreiden naar een eigen thema.

De tabs in Android zijn belangrijk voor het platform. Het is een element dat erg opvalt in Android apps. Het is ook een van de elementen die een slechte eerste indruk kunnen geven met betrekking tot jouw eigen app.

Volgens Android UI Patterns is het vooral belangrijk om het niet te laten lijken op een app die hoort op een ander platform, zoals iOS. De eerste indruk is belangrijk, je riskeert anders dat de gebruiker niet tevreden is met de app. Een ontevreden gebruiker zal ook waarschijnlijk niet de app geïnstalleerd houden.

Voorbeelden

LinkedIn

LinkedIn bouwde eerst een app in HTML. Dit jaar koos LinkedIn ervoor om voor native clients te gaan. Enkele redenen hiervoor waren dat de app te veel geheugen vroeg en dat de animaties niet klopten. De animaties waren niet vloeiend genoeg volgens Kiran Prasad, LinkedIn's senior director for mobile engineering. Ook waren er problemen met de tools. Het is bijvoorbeeld niet goed mogelijk om te analyseren waar performance problemen vandaan komen met web apps. [21]

Facebook

Mark Zuckerberg, oprichter en CEO van Facebook, gaf in 2013 aan dat een van de grootste fouten die zij hebben gemaakt was dat ze in HTML5 hebben gebouwd. [22] [23]

"The biggest mistake we made as a company was betting too much on HTML5 instead of native. We burnt two years. Native is going to be the approach. We're betting completely on it, Native is going to be the approach that we go with for iOS and Android." -Mark Zuckerberg,

Uber

Een voorbeeld van een app waarbij gebruikers niet te spreken waren van het feit dat de app niet native was, was de app Uber. [1] Bij deze app werd er niet goed omgegaan met de look & feel. De gebruikers merkten dat het niet native was en kreeg het uiteindelijk een gemiddelde score van 2 uit de 5 sterren. [24]

Conclusie

Een native look is dus voor gebruikers belangrijk. Het is zonde om halverwege de ontwikkeling van een app te moeten wisselen naar native. Dit kost immers veel tijd en ook geld die beter besteed had kunnen worden.

De huidige stand van de frameworks

Omdat de look & feel belangrijk is voor de gebruiker, is dit opgenomen in dit onderzoek. Elk framework is onderzocht en hieronder staan de bevindingen.

Sencha

Sencha levert bij zijn framework verschillende stylesheets. De Android stylesheet lijkt echter niet te passen bij het OS. De Windows Phone stylesheet is duidelijk geïnspireerd op het Windows Phone platform, maar niet alle elementen lijken op de native elementen. De pivots, de Windows Phone versie van een Tab, scrollen bijvoorbeeld niet mee. Bij een spinner field, een veld speciaal voor numerieke invoer, geeft hij een plus en min om de invoer te veranderen. In een native app zou dit niet gebeuren, de invoer zou alleen via het schermtoetsenbord gebeuren.

In Windows Phone werken de symbolen, zoals de plus en min, niet. Dit zou moeten werken door een font-family genaamd Pictos. Dit font werd gebruikt voor bepaalde symbolen zoals een + of – karakter. De font-family wordt echter niet ingeladen wanneer deze gewrapped is in Cordova. Bij het navigeren naar een gehoste versie van de app in de browser geeft hij de symbolen wel weer. Dit probleem is bekend bij de makers van Sencha. [25] Dit probleem lijkt te maken te hebben met restricties in Windows Phone. In Windows 8 is het niet toegestaan om Data URI's voor lettertypes te gebruiken in lokale context, dit kan ook het geval zijn met Windows Phone 8. [26]

Enyo

Enyo gebruikt een eigen look & feel. Dit is voor elk platform hetzelfde.

jQuery

jQuery levert een standaard thema op wat geïnspireerd lijkt door iOS. Er zijn echter een aantal thema's beschikbaar voor jQuery waaronder een voor Windows Phone. [27] Een Android thema is er echter niet.

Onafhankelijke CSS

Het is niet nodig om de CSS te gebruiken van jQuery, Sencha of Enyo. Het is mogelijk om een eigen stylesheet te gebruiken. Dit komt omdat de bovengenoemde frameworks allemaal werken met HTML, CSS en JavaScript. Het is dan wel nodig om deze aan te passen aan het framework.

Windows Phone

Dankzij de jQM thema for Windows Phone [28] was het mogelijk om een pagina te maken die native lijkt. Het thema bevat de benodigdheden om een app te maken in de Windows Phone stijl. Het enige nadeel is dat het thema geen pivots, het Windows Phone equivalent van tabs, ondersteunt.

Android

Voor Android is het lastiger om een goed thema te vinden. Er zijn meerdere thema's die het proberen te imiteren. Het is tot op heden echter nog niet gelukt.

Een van de eerste thema's die gevonden werd was de *jqmobile-android-holo-light-theme* van *enathu* [29]. Hoewel het thema duidelijk gebaseerd is op de holo guidelines, volgt het niet de stijl van Android zoals in native apps te verwachten is.

Een ander thema is *jquery-mobile-android-theme* van *jjoe64* [30]. Ook hier is het duidelijk dat het is geïnspireerd door Android, maar het emuleert niet de look die te verwachten is van een native app.

De *Holo-Holo-Theme* van *zeickan* [31] heeft tabs, maar het mist andere belangrijke elementen, zoals ook in de tabel hieronder te zien is. De belangrijkste tekortkomingen zijn de checkboxes, radio buttons en dropdown menu's, zijn niet ingebouwd.

De *holo-css* van *vezquex* [32] heeft veel elementen die lijken op de native elementen. Het lastige van dit thema is dat het moeilijker is om te implementeren. Er zijn extra divs nodig voor dit specifieke thema.

De keuze is uiteindelijk gegaan naar de *holo-web* thema van *zmyaro* [33]. Hoewel het niet zoveel beschikbare elementen heeft, klopt de look wel.

Een vergelijking van de beschikbare elementen per stylesheet. Zoals te zien is in Tabel 5, is er geen stylesheet die alles implementeert. Bij elke stylesheet is er op zijn minst een element die ontbreekt.

	enathu	jjoe64	zeickan	vezquex	zmyaro
Donker	x	v	x	v	v
Licht	v	v	v	v	v
Button	v	v	v	v	v
Tekst field	v	v	v	v	x
Check box	v	v	x	v	v
Radio button	v	v	x	v	v
Slider	v	v	v	x	x
Toggle button	v	v	v	v	x
Tekst	v	v	v	v	v
Tabs	x	x	v	v	x
Drop down	v	v	x	v	v

TABEL 5

Orubase

Door de andere aanpak van Orubase waarbij een deel native en een deel via een webserver draait (zie ook het hoofdstuk “Beschikbare frameworks”), komen deze apps wel goed naar voren qua tabs/pivots.

De huidige versie (1.1.0.69) heeft echter een aantal visuele bugs. In eerste instantie zou er een nieuwe release zijn die de problemen op lost in april. Uiteindelijk is dit uitgesteld naar mei en uiteindelijk naar eind juni. De problemen waren:

De Android look klopt niet.

In de huidige versie klopt de Android look niet. De invoervelden zijn bijvoorbeeld slecht te zien, de knoppen zijn te klein en zo zijn er nog meer kleine dingen. Syncfusion, het bedrijf achter Orubase, geeft aan dat zij werken aan de Android look. Dit zou in de volgende release opgelost moeten zijn.

Wachtwoord “peek” functie werkt niet goed in Windows Phone 8

In Windows Phone is het gebruikelijk om, bij het invoeren van een wachtwoord, het laatste karakter kort te tonen. Dit gebeurt echter niet bij Orubase.

Dit schijnt volgens Syncfusion een probleem te zijn bij jQuery, dit heeft Syncfusion bij jQuery aangegeven en het probleem zou in de volgende release opgelost worden. Het hoort het laatst ingetypte karakter te tonen. Dit gebeurt goed zolang de focus niet weg gaat. Als dit wel gebeurt dan toont het een willekeurig karakter.

Witte ruimte onderaan pagina's

Bij sommige pagina's is onderaan de pagina een stuk witruimte. In de volgende release wordt dit opgelost door de achtergrond standaard zwart te maken.

Naar links/rechts swipen om van pivot te veranderen werkt niet goed

Het is niet mogelijk om in de pagina zelf naar links of rechts te swipen. Het is wel mogelijk om dit bij de pivot titel te doen of door de titel aan te klikken. Dit probleem is er omdat in de pivot control een web browser control de swipes afvangt. Een swipe is een veeg van de linker- of rechterkant van het scherm naar de tegenovergestelde richting,. [34]

Navigeren van native pagina's

Het is binnen de app niet mogelijk om de native pagina's, in Android een Activity en in Windows Phone een Page, te veranderen via de webpagina. Het is niet mogelijk om bijvoorbeeld een pagina te hebben zonder native tabs/pivots en dan te navigeren naar een pagina met native tabs/pivots.

Mosync

Mosync heeft met zijn Native UI API de native interface elementen beschikbaar gesteld voor HTML + JavaScript en C/C++. Hierdoor lijkt de app op elk platform native. Een positieve eigenschap van MoSync is dat tabs in Windows Phone ook werken zoals bij het platform hoort. Deze worden omgezet naar pivots. [35]

MoSync heeft echter wat problemen. Een voorbeeld is dat er geen dropdown menu/combo box bestaat. Op een forum is gezegd dat dit er niet in zit omdat zij het dan voor elk OS geschikt moeten maken, dit zou bij kleine apparaten en niet-touchscreen apparaten problemen opleveren. [36] Ook plaatst Mosync de elementen in het midden van het scherm in plaats van het links te plaatsen. Hierdoor zijn het wel de juiste elementen, maar op de verkeerde plaats van het scherm.

Een andere hindernis is het niet goed ondersteunen van verschillende schermformaten. De lettertype groottes worden in pixels gedefinieerd waardoor de letters groter of kleiner zijn in een andere

schermresolutie. [17] Dit zorgde ervoor dat op de Nokia Lumia 800 en Samsung Galaxy S II de letters goed leesbaar waren door de 480*800 resolutie, terwijl het op de Samsung Galaxy S IV te klein was door de 1080*1920 resolutie. Dit is op te lossen door de lettertype grootte, na het renderen van de pagina, aan te passen.

Rhomobile

RhoMobile emuleert niet de look van Android en Windows Phone. Iets wat ikzelf wel verwacht had door de platform specifieke stylesheets die in het project staan. Het is mogelijk om de stylesheets aan te passen zodat het wel native lijkt, het standaard thema doet dit niet.

Een ander probleem, wat de look & feel belemmert, is dat het ongewenst inzoomt. Dit gebeurt bijvoorbeeld bij het aantikken van een textbox.

Marmalade

Marmalade emuleert de look niet van Android. Voor Windows Phone was het niet te testen. Dit omdat Windows Phone ondersteuning nog in bèta is en het verkrijgen van een studenten licentie niet lukt. Een licentie is namelijk nodig om de bèta versies te gebruiken.

Xamarin

Xamarin pakt het cross-platform ontwikkelen anders aan. De front-end wordt gemaakt zoals je zou verwachten van de native platformen. De Windows Phone app maak je zoals gebruikelijk is voor het platform. Voor de Android versie maak je de pagina's in XML. De code van de Android app is echter C#. Zo is het bijvoorbeeld mogelijk om de backend, zoals API toegang en berekeningen, te delen tussen de platformen.

Niet alle code kan opnieuw gebruikt worden, de Windows Phone specifieke elementen, zoals de progress indicator, bestaan niet in Xamarin. Verder zijn er OS specifieke stukken code nodig. Voor Android is er een andere manier om instellingen op te slaan dan bij Windows Phone. Door middel van preprocessor directives [37] is dit af te vangen. Zie voor meer informatie over preprocessor directives Bijlage C: Preprocessor Directives.

Conclusie

De look verschilt erg tussen de frameworks. Waar Sencha een heel eigen look heeft, gebruikt Xamarin de native elementen. De andere frameworks zitten hier tussenin waarbij Mosync het meeste opvalt. Het heeft veel van de native elementen, maar het plaatst het niet goed op het scherm.

Performance

Omdat performance ook een belangrijke factor is in apps, zijn hier een aantal tests mee gedaan. De keuze voor deze tests is dat dit, in mijn ogen, de belangrijkste punten testen. Het opstarten, web performance en bereken performance. Hierbij zijn de volgende tests gedaan:

Startup test

Bij de startup test is gemeten hoe lang het duurt van het moment dat de gebruiker het icoon aanklikt tot het moment dat de app te gebruiken is.

Login afhandeling

In deze test is gemeten hoe lang het duurt van het drukken op de inlog knop tot het moment dat het volgende scherm is weergegeven. Het belang van deze test is dat kan worden gemeten hoe lang web request afhandelingen duren. Bij een app die veel met web requests doet zal dit zwaar meegerekend moeten worden.

Berekening

Bij deze test is gemeten hoe lang het duurt van het moment dat op de bereken knop wordt gedrukt tot het moment dat het volgende scherm wordt getoond. Indien er in een app veel wordt gerekend, is dit een belangrijke factor om een framework te kiezen.

Algemeen over de tests

Om de performance te testen zijn een set testcases uitgevoerd. Elke testcase zal vijf keer worden uitgevoerd. Indien één test een extreme afwijking⁵ heeft, dan zal de testcase opnieuw worden gedaan voor dat platform en die framework.

Uit de vijf tests zal het gemiddelde worden genomen en met dit gemiddelde zullen de frameworks worden vergeleken. Na elke test zal het installatiebestand, een xap of apk, opnieuw geïnstalleerd worden zodat er geen data achter blijft. De applicaties zijn gebouwd in release mode.

De tests zijn elke keer op dezelfde telefoon gebeurt. Zie ook Bijlage E: Gebruikte testapparaten.

Alle resultaten zijn in het volgende formaat: seconden : honderdsten van een seconden.

De ruwe meetdata staan in Bijlage G: Meetresultaten.

Gebruikersfeedback

Alle apps zijn ook getest door 15 gebruikers. Hierbij is niet gekeken naar de ruwe performance. Er is gekeken naar wat de gebruikers vinden van de snelheid van de apps. Om ervoor te zorgen dat de gebruikers niet bevooroordeeld zijn, kregen de apps geanonimiseerde namen. Dit waren App 1 t/m App 5. Hierbij waren vooraf opgestelde vragen en geïmproviseerde vragen. De vragen en antwoorden waren in audiovorm opgenomen.

Door het grote verschil qua vragen en antwoorden, dit was gebaseerd op de situatie en voorgaande antwoorden, is ervoor gekozen deze antwoorden niet letterlijk te verwerken, maar om hier een aantal opvallende punten uit te pakken.

⁵ Meer dan 2 keer zoveel/minder dan het gemiddelde van de andere 4 waardes voor de testcase.

Xamarin

Xamarin geeft de mogelijkheid om C# te gebruiken in Android en iOS. Dit framework is niet nodig bij Windows Phone omdat die code al in C# is. Hierdoor komt Xamarin niet in de Windows Phone tests naar voren.

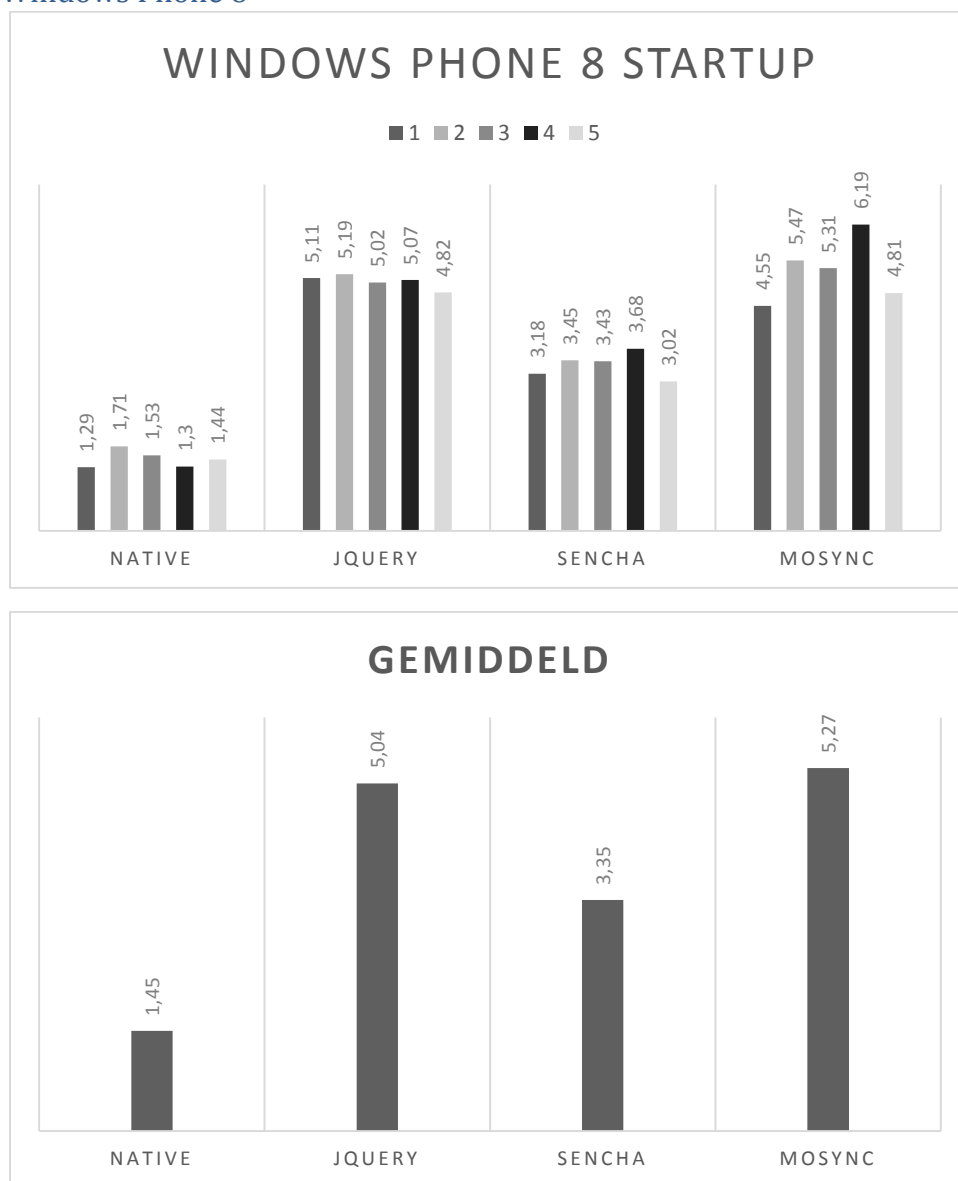
Startup

Methode

Bij deze test is de tijd gemeten van het aanklikken van het icoon tot het moment dat de applicatie gebruiksklaar⁶ was.

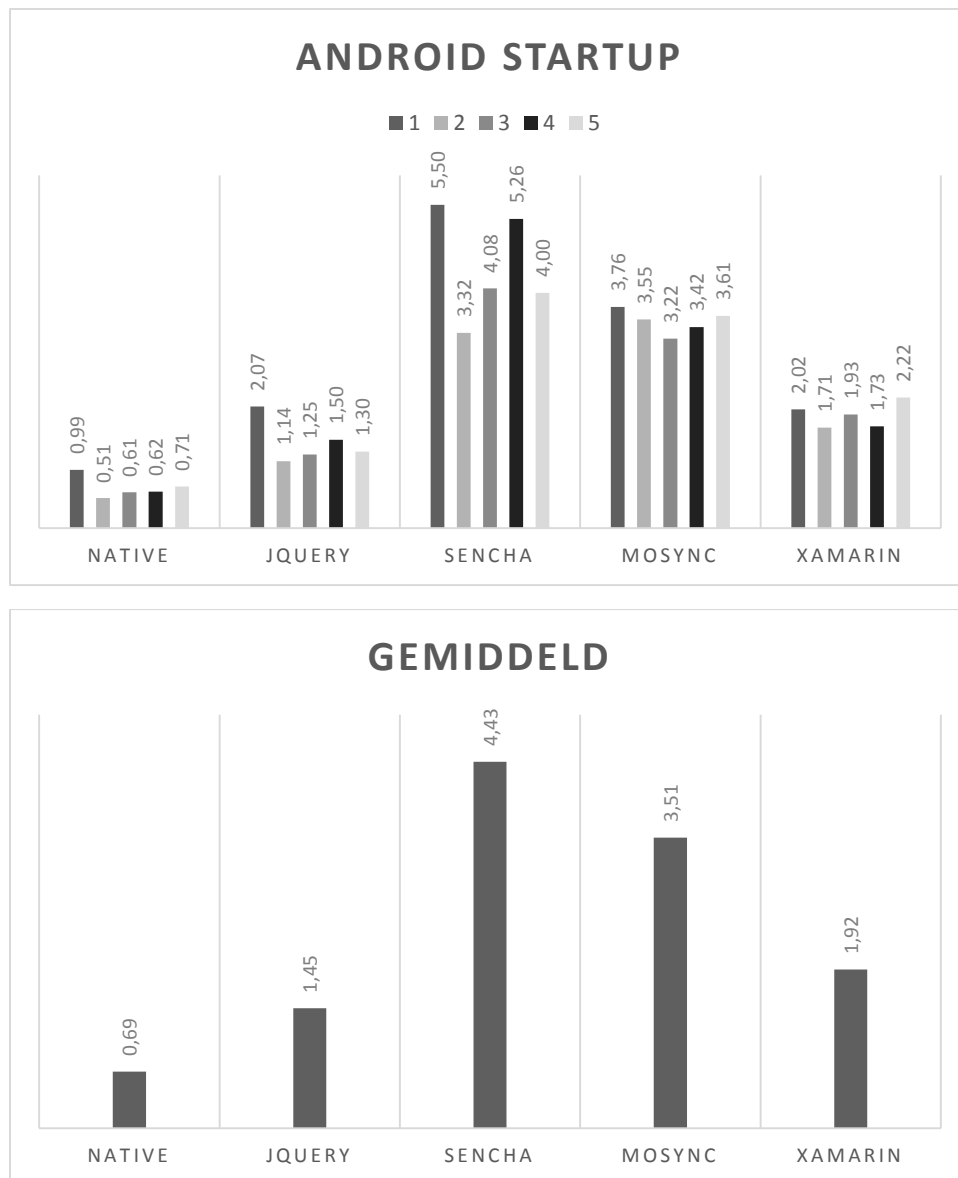
Dit is met een digitale stopwatch, handmatig, gemeten. De reactiesnelheid is een factor. De tests werden door dezelfde persoon gedaan en was, net zoals de andere tests, vijf keer gedaan. Hierdoor zouden de metingen betrouwbaar moeten zijn.

Resultaten Windows Phone 8



⁶ Gebruiksklaar is in dit geval het moment dat de gebruiker de applicatie kan gebruiken. Dit is bij de applicaties het moment dat het inlogscherm getoond word.

Resultaten Android



Resultaat

Wat opvalt bij de opstarttest is dat het opstarten bij native apps minimaal twee keer sneller opstarten dan de frameworks. In sommige gevallen is het zelfs meer dan vier keer zo snel. Het verschil tussen de native client en de andere frameworks is groot.

Gebruiker feedback

Opvallend was dat maar drie van de onderzochte mensen klaagden over de traagheid van het opstarten van de app. Dit ligt echter aan de frameworks. De andere testers hadden niets op te merken over de opstartsnelheid van de apps.

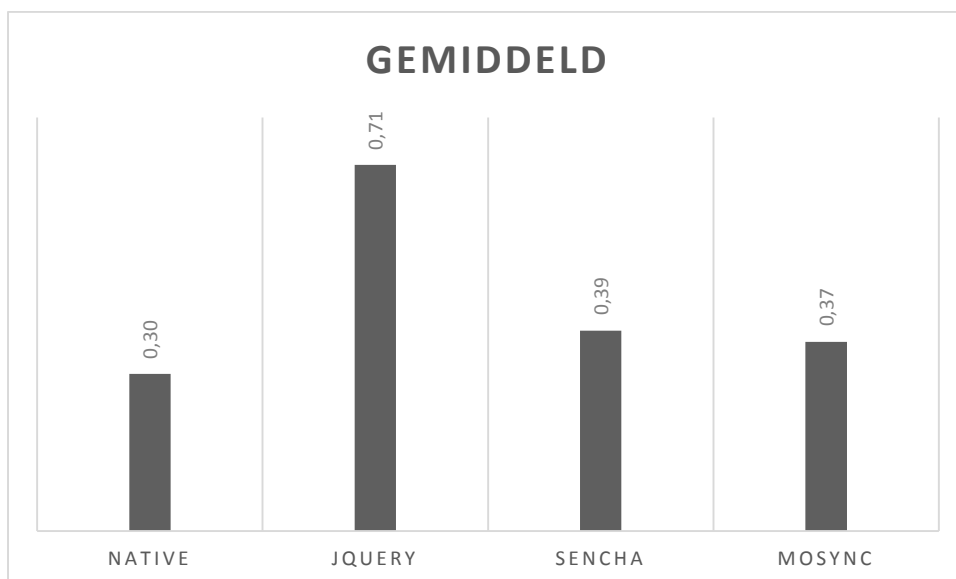
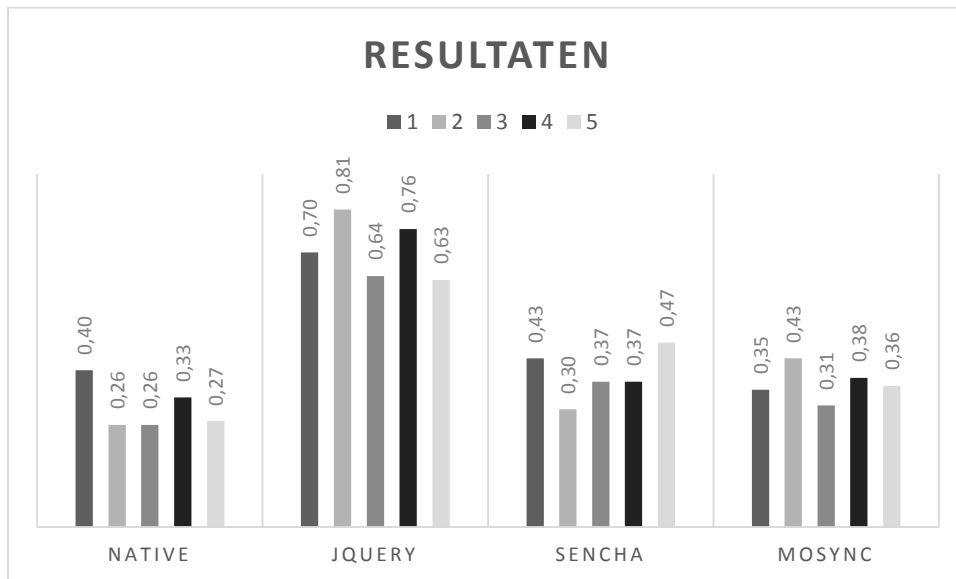
Login afhandeling

Methode

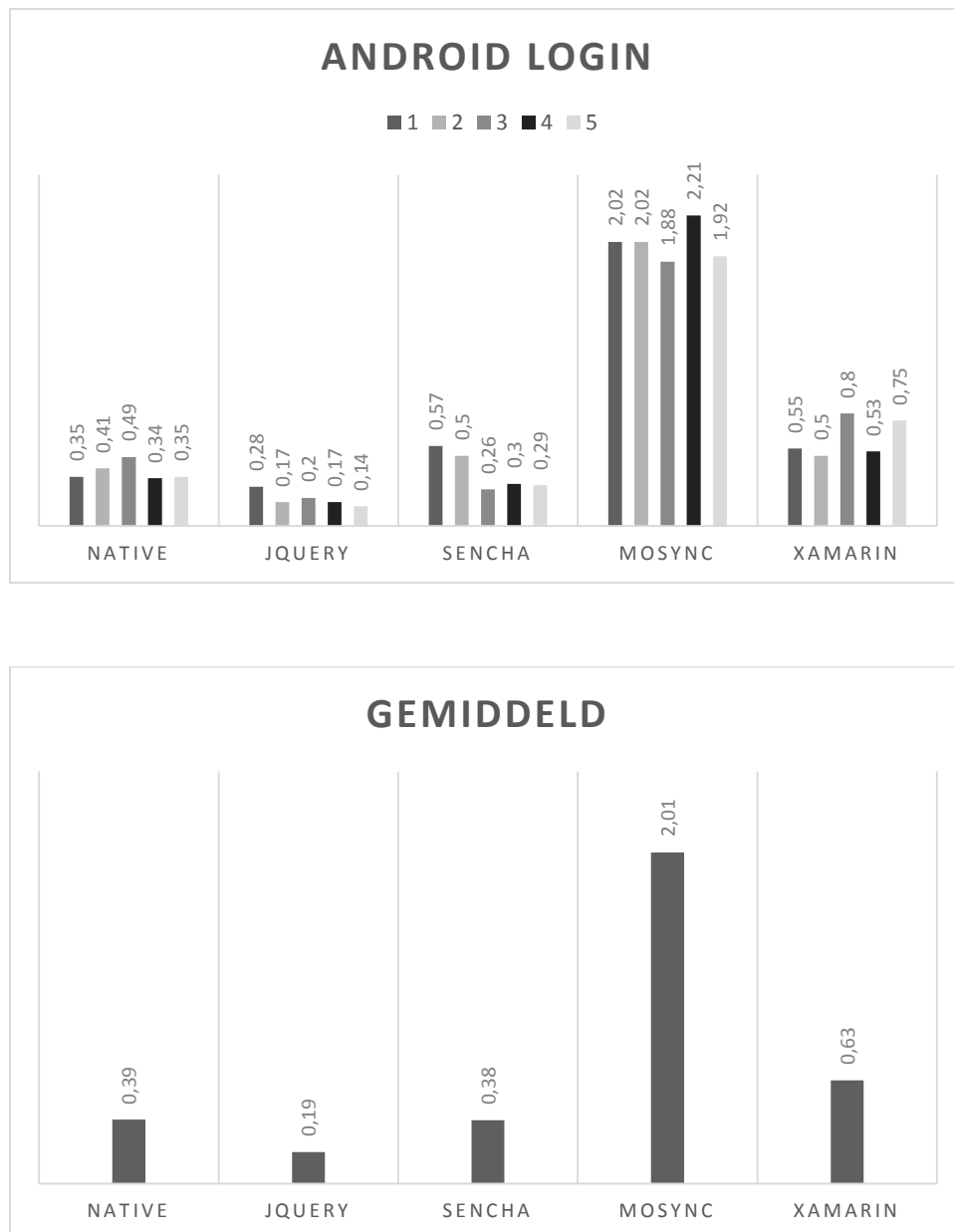
Bij deze test is gemeten hoelang het duurt van het aanklikken van de login knop tot het tonen van het volgende scherm. In verband met vertragingen op het internet is ervoor gekozen om de login server van de klant niet te gebruiken. In plaats van de login server is er een lokale computer gebruikt in hetzelfde netwerk. Deze computer heeft een statisch html bestand die altijd dezelfde tekst teruggeeft.

Het opvragen van de pagina gaat door middel van een HTTP POST request.

Resultaten Windows Phone 8



Resultaten Android



Resultaat

Op Windows Phone is het zo dat native sneller is dan de frameworks met het inloggen. Het inloggen is bij elke app sneller dan 1 seconde.

Bij Android was native niet het snelst dit keer. JQuery deed het hier twee keer zo snel. MoSync was echter vijf keer trager dan native.

Gebruiker feedback

Door het relatief lange wachten in MoSync en soms ook andere frameworks, was het zo dat vier gebruikers opnieuw op login drukten. Dit omdat men dacht dat ze niet goed de login knop hadden ingedrukt. In werkelijkheid duurde het verwerken erg lang. Eén andere gebruiker gaf achteraf aan dat hij de snelheid “bedroevend” vond in sommige apps.

Berekening

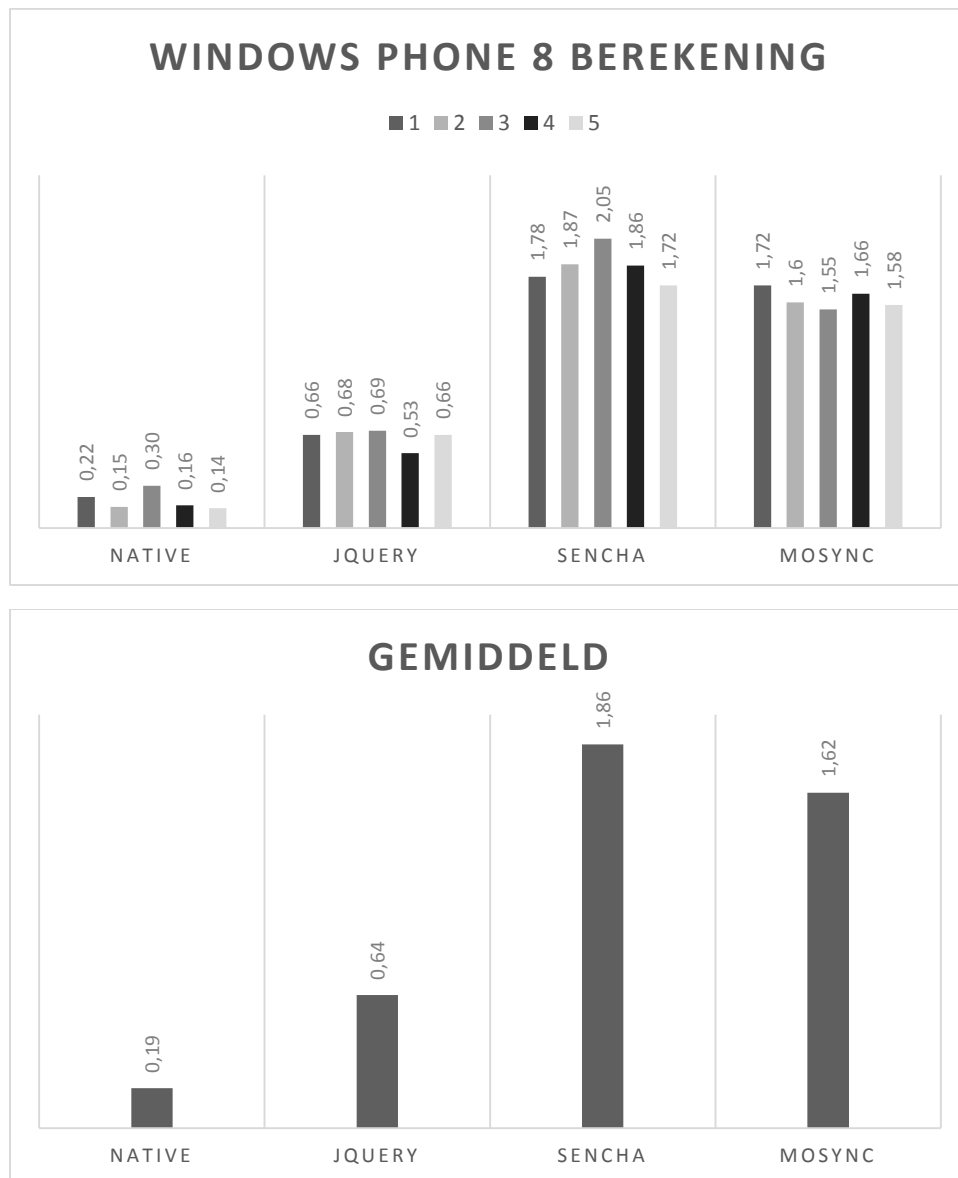
Methode

Bij deze test is gemeten hoelang het duurt van het aanklikken van de bereken knop tot het tonen van de resultaten.

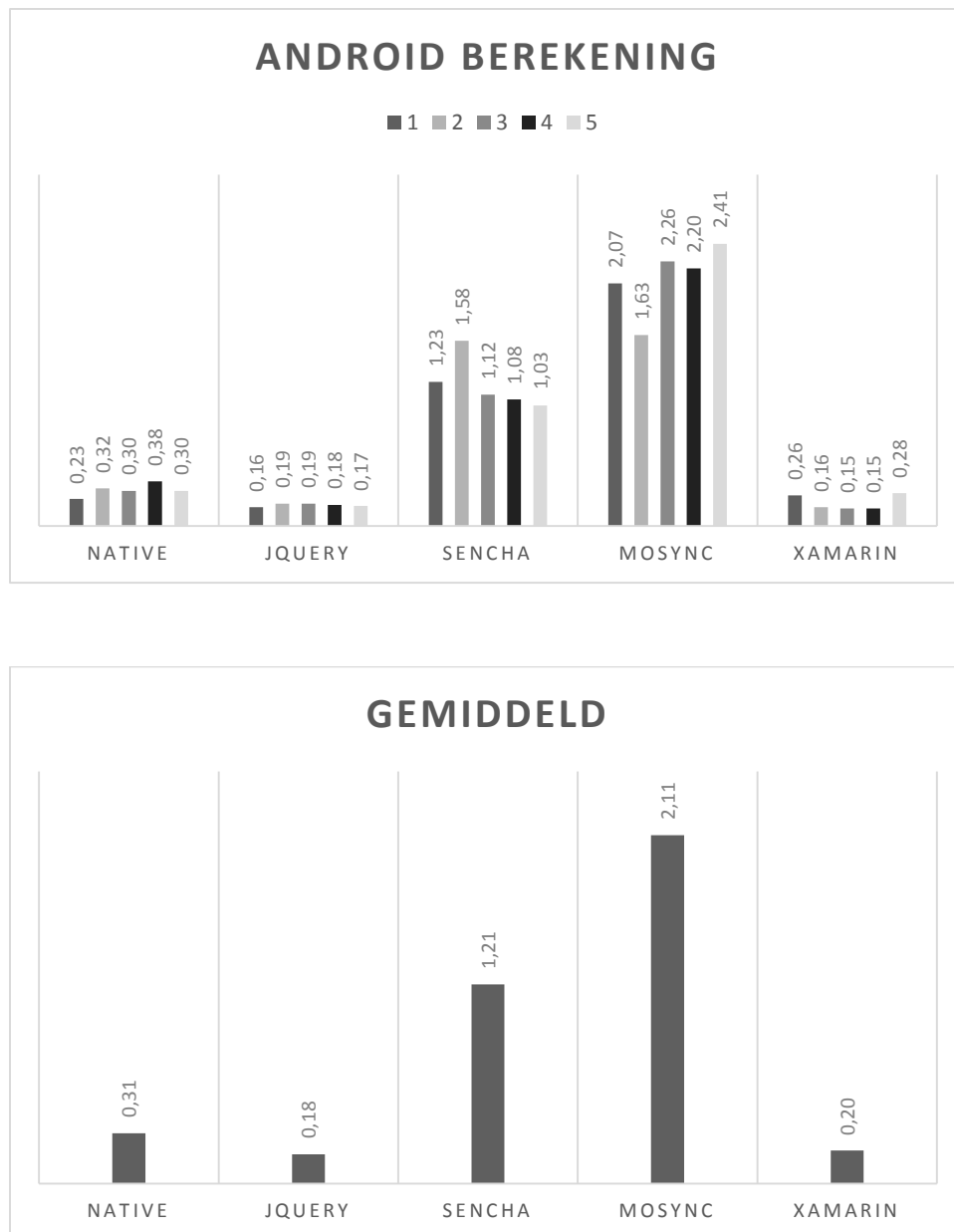
Testwaarden:

- Bedrag: 500
- Termijnen: 50
- Termijnduur: Jaar
- Rente: 3
- Valuta: Euro

Resultaten Windows Phone 8



Resultaten Android



Resultaat

Bij Android was native de winnaar qua snelheid. Deze toonde de pagina gemiddeld binnen 0:19 seconden. Sencha was hier het traagste. Die was bijna tien keer zo traag.

Bij Android was jQuery wederom sneller dan native. Dit keer was jQuery twee keer sneller. De traagste was MoSync. Die had meer dan anderhalve seconde meer nodig om de resultaten te laten zien.

Gebruiker feedback

Bij deze test was het zo dat drie gebruikers klaagden bij MoSync op Android. Eén van de gebruikers gaf aan dat bij zo'n high end toestel een berekening niet zo lang zou moeten duren.

Conclusie

De frameworks leveren allemaal in met betrekking tot de performance. In sommige gevallen, zoals jQuery in Android, is het framework sneller bij het uitvoeren van de acties dan native. Als het gaat om de opstartsnelheid, is deze echter een stuk trager. Ook is jQuery relatief traag op Windows Phone. Sencha is trager dan jQuery. Dit komt vermoedelijk door de effecten en de manier van opbouwen van de pagina's. In plaats van statische HTML pagina's, bouwt Sencha het elke keer zelf op.

MoSync is zeer traag met al zijn acties. Hierdoor kan het zijn dat het framework een slechte keuze is. Het kost immers teveel tijd voor een gebruiker om iets te doen.

Xamarin is een framework voor iOS en Android, voor Windows Phone gebruik je de native code. De performance is in Windows Phone dus native. Xamarin in Android zorgt voor een zeer goede performance met betrekking tot de andere frameworks. In het geval van de berekening is Xamarin zelfs sneller dan native.

Het gebruik van een framework hoeft dus niet per se negatief te zijn voor de performance. In sommige gevallen kan het zelfs beter zijn.

Code herbruikbaarheid

Als er een app wordt gemaakt voor meerdere platformen die hetzelfde doen, kan verwacht worden dat veel van de methodes/berekeningen hetzelfde doen. Het kan echter zo zijn dat bepaalde API's anders heten in de platformen. Het is echter zonde om alle code opnieuw te schrijven, dit kost veel tijd en als er iets gewijzigd wordt in een platform, moet dat ook in de andere. Vandaar dat is onderzocht hoeveel code er daadwerkelijk kan worden gedeeld zonder problemen en hoe eromheen kan worden gewerkt.

Gelukkig lijken C# en Java op elkaar. Beiden zijn object georiënteerd programmeertalen en hebben een overeenkomende syntax en stijl. [38] [39] Er zijn echter ook verschillen (zie Bijlage D: Overeenkomsten en verschillen tussen C# en Java). Omdat de talen toch op elkaar lijken heb is er onderzoek gedaan naar of de Java code in zijn geheel, of deels, kan worden hergebruikt. In dit hoofdstuk wordt er vanuit gegaan dat de lezer kennis heeft genomen van de bijlage.

Frameworkloos

In dit kopje zal worden gekeken naar hoeveel code er kan worden hergebruikt zonder framework. Hierbij zijn de apps al native gemaakt in Android en Windows Phone en werd de code in een nieuw project van het tegenovergestelde platform geplaatst. Hierbij zijn geen tools gebruikt.

Tijdens het overzetten van Windows Phone naar Android waren er enkele problemen. Uiteraard door de opmaak van de bestanden zoals in Bijlage D: Overeenkomsten en verschillen tussen C# en Java staat, maar ook door het ontbreken of anders heten van methodes. In Tabel 6 staan de bevindingen van de overzetting.

#	Probleem	Generiek te maken	Oplossing
1	Internal is een access modifier die niet bekend is binnen Java, maar wel in C#.	Ja	Geen gebruik maken van de internal modifier.
2	Properties bestaan niet	Ja	Gebruik maken van variables of eigen gemaakte get en set methodes
3	Enums kunnen niet een eigen waarde hebben in Java	Nee	Een methode in de enum, dit kan echter niet in c#. Dit betekent dat een aparte methode zou moeten worden geschreven die controleert op de index of naam van de enum.
4	Andere naamconventies, Math.Pow vs Math.pow	Ja	Dit kan lopen via een adapter pattern waarbij de Android en Windows Phone versie het anders af handelen.
5	Andere namen voor functies, Console.WriteLine vs System.out.println	Ja	Dit kan lopen via een proxy klassen waarbij de Android en Windows Phone versie het anders af handelen.
6	Platform specifieke opties, bijvoorbeeld de progress bar, mail	Ja	Adapter pattern

TABEL 6

Zoals te zien is in Tabel 6 kunnen de meeste punten gemakkelijk opgelost worden.

Punt 1 is gemakkelijk op te lossen.

Punt 2 kan door methodes opgelost worden of met variabelen, hiermee worden de properties omzeilt.

Punt 3 is eigenlijk een lastigere, deze is bij de twee talen niet generiek te maken. Er is hier altijd een aangepaste variant van code nodig.

Punt 4 is een probleem door de naamconventies binnen de twee talen. Hoewel beide talen voor CamelCase kiezen, heeft Java gekozen voor lowerCamelCase, het schrijven van CamelCase, maar dan met uitzondering van de eerste letter. Bij C# is er gekozen voor PascalCase. Via een proxy klasse kan dit opgelost worden. Dit zorgt echter wel voor dubbele code.

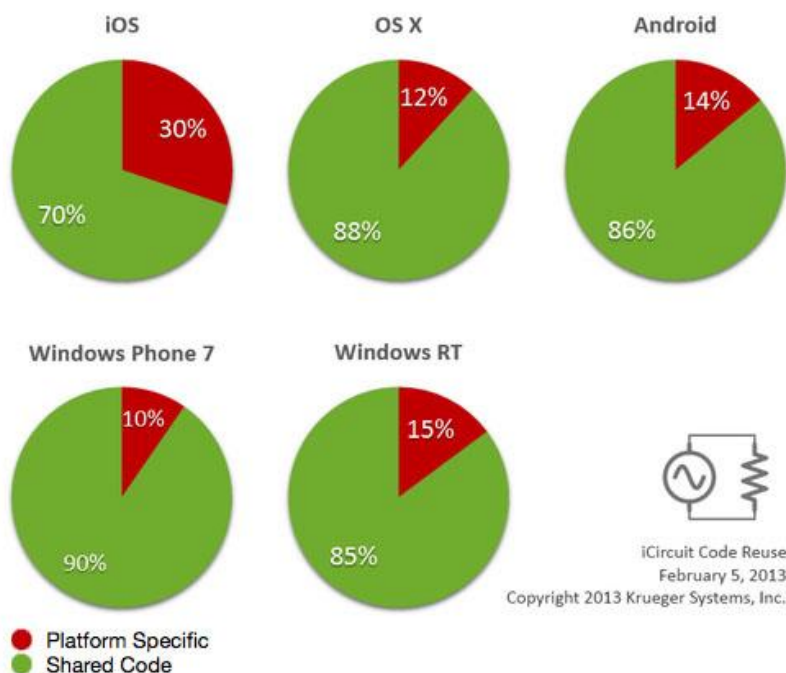
Punt 5 komt overeen met wat er in punt 4 werd vermeld. Het betreft echter niet een hoofdletter, hier gaat het om anders hetende functies. Net zoals bij punt 4 is dit op te lossen met proxy klassen. Ook hier komt dan duplicaat code.

Bij punt 6 gaat het om platform specifieke code. Een voorbeeld hiervan is het versturen van een mail bericht. In het geval van Windows Phone zal hier een EmailComposeTask moeten worden gemaakt. Bij Android zal dit via een ActionSend Intent moeten gebeuren.

Met framework/tool

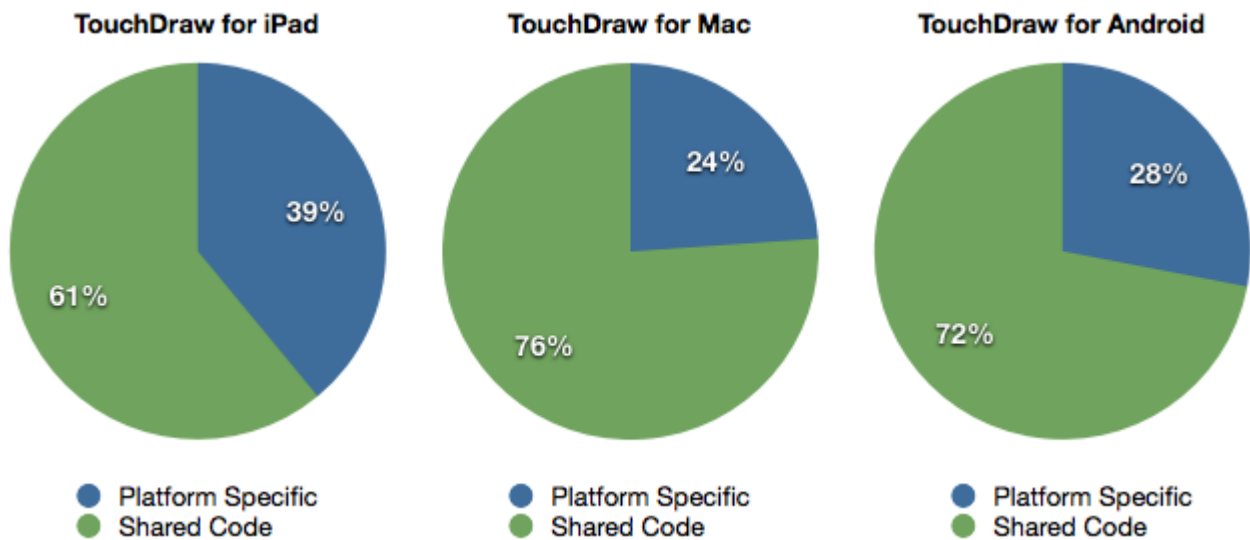
Xamarin

Xamarin zorgt ervoor dat C# code in Android en iOS kan worden gebruikt. Dit stelt ontwikkelaars in staat om code gemakkelijk te delen tussen verschillende apps op de platformen. De GUI van de applicaties wordt echter wel per platform geschreven. In de onderstaande voorbeelden is gebruik gemaakt van dezelfde code om hergebruik te berekenen. [40]



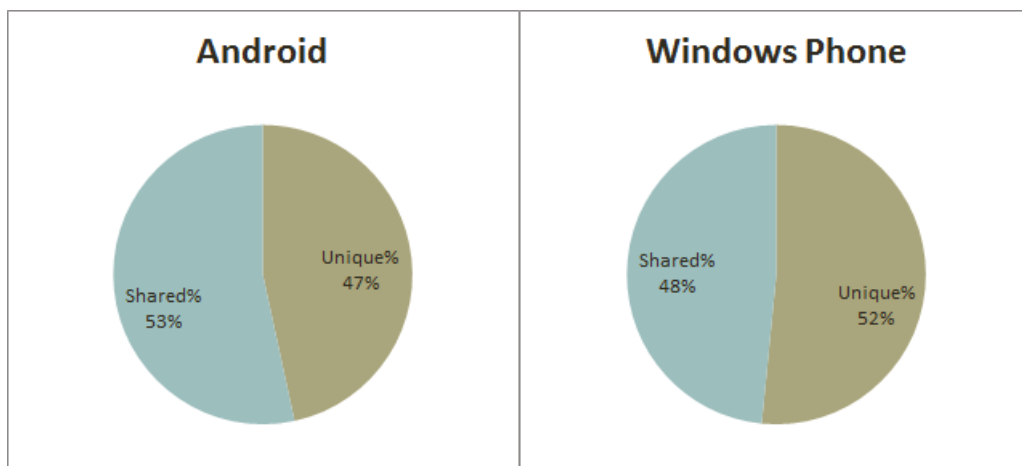
FIGUUR 9

Frank Krueger, maker van iCircuit, heeft zijn tool met behulp van Xamarin gemaakt. Hij kwam tot de conclusie dat gemiddeld 85% van de code hergebruikt kan worden tussen iOS, OS X, Android, Windows Phone en Windows RT. [41] Frank Krueger geeft ook aan dat dankzij Xamarin 39.00 regels code konden worden gedeeld.



FIGUUR 10

Jon Lipsky, maker van TouchDraw, werkt ook met Xamarin. Bij het ontwikkelen voor OS X, iPad en Android was er gemiddeld 70% code hergebruikt. [42] Door Xamarin te gebruiken is veel code opnieuw gebruikt. De projecten hebben meer dan 120.000 regels code. Door het delen van zo'n 70% code scheelt dit veel tijd.



FIGUUR 11

In de Contante Waarde app was de gedeelde code minder dan de bovenstaande voorbeelden. Dit heeft te maken met de grootte van de app en de complexiteit van de backend. Gemiddeld is er 50% gedeelde code.

FIGUUR 12

Xamarin is een framework dat de mogelijkheid biedt om code eenvoudig te hergebruiken. Dit is niet zoveel als andere frameworks, maar het geeft wel de mogelijkheid om dit te koppelen aan de native opmaaktaal. Het voordeel van een native user interface is dat de applicatie native aanvoelt.

CS2J

CS2J is een gratis C# naar Java converter. Het scant de .cs bestanden en converteert de code waar dit mogelijk is. Hier komen .class bestanden uit die in Android projecten kunnen worden gebruikt. Het betreft hier een conversie van C# source code naar Java source code, dus niet Windows Phone specifieke code naar Android. In het geval van de Contante Waarde app kon ik niet de settings klasse niet hergebruiken. De XML writer van C# wordt niet overgezet en er waren nog meer kleine problemen die ervoor zorgen dat de overzetting niet meteen werkte.

De CS2J converter stopt ook als hij een stuk code niet herkent. Bij async tasks bijvoorbeeld converteert hij het bestand niet en gaat hij vervolgens verder met het volgende bestand. Met CS2J gaat het wel makkelijker en sneller dan het handmatig porten zoals in het begin van dit hoofdstuk staat, maar het is niet zo geavanceerd als Xamarin.

Conclusie

Het hergebruiken van code kan zeer zeker. Met de native frameworks is het echter af te raden. Het kost veel tijd om alles generiek te maken. Hiervoor is Xamarin een goede oplossing. Het biedt de mogelijkheid om de code in C# te maken en te delen in Android.

CS2J was een goed idee, het overzetten van C# naar Java, maar het is niet geschikt voor Windows Phone en Android. Het probleem is hierbij vooral dat het stukken code niet herkent en dan de hele klasse overslaat.

Cordova port

Binnen Lunatech was al een Cordova versie van de rekentools, een set applicaties beschreven in Bijlage A: Mobiele Rekentools. Het was de bedoeling dat deze geschikt zou worden gemaakt voor Windows Phone 8. In eerste instantie werd er rekening gehouden met problemen door de rendering engine, Trident, met betrekking tot de look. Dit bleek echter niet het geval te zijn. Er waren echter wel andere problemen zoals hieronder vermeld staan.

No transport/NetworkError

Een probleem dat werd ontdekt was dat de AJAX verzoeken niet werden verzonden. [43]

In Windows Phone 7 zal de fout *No Transport* als *errorThrown* worden teruggestuurd. Bij Windows Phone 8 is dit de fout *NetworkError*.

Dit komt door de beveiliging, op Windows Phone is het standaard niet toegestaan om cross-site XML requests te doen. Dit in verband met de veiligheid van de gebruiker. [44] De workaround hiervoor is voor de web request het volgende te zetten: **\$.support.cors = true;**

Automatisch inloggen werkt niet

Tijdens het ontwikkelen werd opgemerkt dat als een gebruiker ingelogd was in de applicatie en hem opnieuw startte, deze niet het hoofdscherm kreeg en ook niet het inlogscherm. Normaal zou een ingelogde gebruiker wel het hoofdscherm moeten krijgen.

Dit kwam door een probleem bij de netwerk detectie in combinatie met de emulator. Bij het automatisch inloggen werd gecontroleerd of er een internet verbinding is. De Windows Phone emulator geeft bij de controle of er verbinding *Connection.UNKNOWN* terug. [45] Tijdens de tests met de Nokia Lumia 820 werd bevestigd dat het automatisch inloggen werkt zoals hoort op een normaal apparaat.

Dit heeft echter niets te maken met het voorafgaande probleem.

Back knop werkt niet

Zowel Android als Windows Phone hebben een back button op de telefoon. iOS toestellen hebben dit echter niet. Aangezien de app eerst is gemaakt voor iOS en deze later is overgezet naar Android, werkt de back button niet.

Dit is echter een probleem voor Windows Phone aangezien Microsoft elke app test voordat het in de store komt. Volgens requirement 5.2.4 [46] van de Windows Phone Store moet de applicatie terug gaan naar de vorige pagina en sluiten als het al de laatste pagina is. Er wordt geen uitzondering gemaakt, zie ook de mail in Bijlage F: Windows Phone Back Button.

In de applicatie werd de backstack, de pagina's waar de applicatie vandaan komt, al bijgehouden. Hierdoor kan door middel van de *popViewController*, een methode in de applicatie, worden teruggekeerd naar een vorige pagina. In de huidige versie van de app word dit aangeroepen door een tekstlink op de pagina.

In de applicatie is het mogelijk dit te koppelen aan de back knop door middel van Cordova, dit lost het dan ook op voor Android.

Conclusie

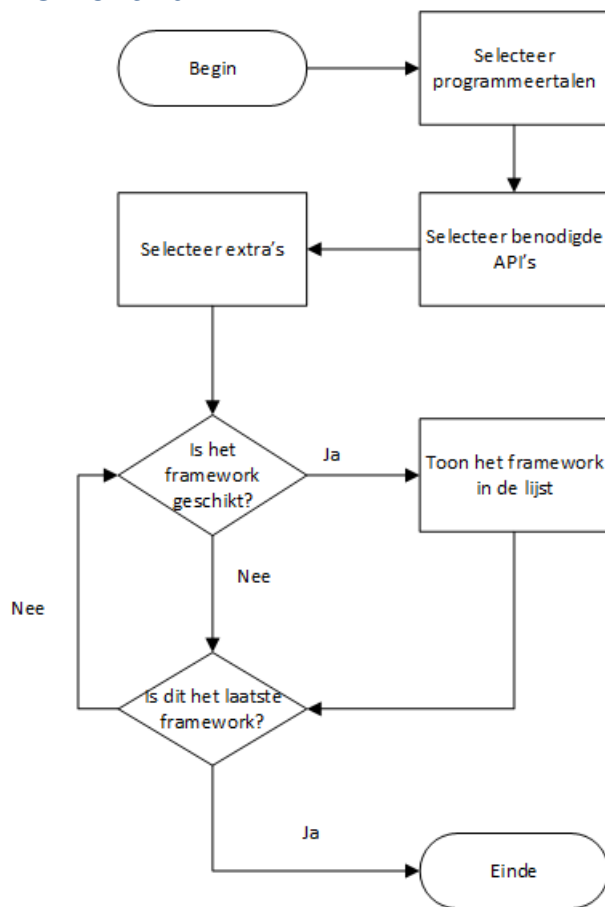
Het ontwikkelen in Cordova geeft een aantal obstakels. Cordova belooft dat de app werkt op elk platform met weinig tot geen wijzigingen. Dit betekend dus, inderdaad, niet dat het in een keer werkt.

Keuzemodel

Een van de onderzoeksvragen stelt: “Hoe kan er een keuzemodel worden gemaakt om te bepalen hoe/met wat er moet worden ontwikkeld?”.

Tijdens het onderzoek bleek dat dit het beste op te lossen is door de gebruiker een selectie te laten maken met de programmeertalen die hij/zij kent, de API's die nodig zijn, of het framework betaald mag zijn en of het framework open-source moet zijn. Hierna krijgt de gebruiker de selectie frameworks.

Flowchart



FIGUUR 13

Screenshots

Framework Picker

Languages known

☒ C#
☒ C++
☒ Java
☒ HTML+JavaScript+CSS

API's required

☒ Accelerometer
☒ Bluetooth
☐ Camera
☐ Compass
☐ Geolocation
☐ Vibration
☐ Alerts
☐ Calendar
☐ Contacts
☐ Media
☐ File
☐ Notification
☐ Push

Extras

☐ Show free frameworks only
☐ Show open source frameworks only
☐ Show untested frameworks as well (these have been found unstable or not fit for purpose).

Name	Open source	Free	Free for	Code	Markup	Notes	Performance	Look	Download
MoSync	✓	✓	Everything	JavaScript C++	HTML+CSS Code				
Native Android	✓	✓	Everything	Java	XML				
Native Windows Phone	✗	✓	Everything	C#	XAML				
Xamarin	✗	✓	Small projects	C#	XML XAML Code	Free version is limited to 32k of compiled user code (IL), and which do not call out to native third party libraries (i.e., developers may not P/Invoke into C++/Objective-C/Java).			

©Jeroen Heijster 2013

FIGUUR 14 HET HOOFDSCHERM

✕

Performance jQuery

Name	Startup	HTTP POST login	Calculation
Android	1.45	0.19	0.18
Windows Phone	5.04	0.71	0.64

FIGUUR 15 HET PERFORMANCE SCHERM VAN JQUERY

✕

Graphics jQuery

The default jQuery mobile theme is styled like an iOS app. There are various themes available. For Windows Phone you can use: [jqmobile-wp8-theme](#)
For Android you have multiple options:

Name	enathu	jjoe64	zeickan	vezquex	zmyaro
Dark	✗	✓	✗	✓	✓
Light	✓	✓	✓	✓	✓
Button	✓	✓	✓	✓	✓
Text field	✓	✓	✓	✓	✗
Checkbox	✓	✓	✗	✓	✓
Radio button	✓	✓	✗	✓	✓
Slider	✓	✓	✓	✗	✗
Toggle button	✓	✓	✓	✓	✗
Text	✓	✓	✓	✓	✓
Tabs	✗	✗	✓	✓	✗
Drop down	✓	✓	✗	✓	✓
Screenshot					
Download					

FIGUUR 16 DE OMSCHRIJVING VAN DE GRAFISCHE OPTIES VAN JQUERY.

Conclusie en aanbevelingen

Het kiezen van een framework hangt af van de kennis van de developer en wat de developer nodig heeft in het framework. Andere factoren zoals of een framework open-source moet zijn spelen ook mee.

Ook de programmeertalen die de developer beheerst zijn belangrijk. Als een developer een nieuwe taal zoals Ruby on Rails moet leren, dan kan de initiële investering zo hoog zijn dat het niet loont om dat framework te kiezen.

Hierdoor is het echter niet zo makkelijk om een keuze voor een framework te maken. Hopelijk maakt dit document het wel makkelijker voor de lezer en helpt de frameworkpicker hier ook bij.

Look

Eigen look

Indien de native look niet meetelt, bijvoorbeeld als de app er op alle platformen hetzelfde uit moet zien, dan kunnen web frameworks een goede uitkomst bieden. Deze zien er op elk platform hetzelfde uit. Xamarin en native wordt lastig. Windows Phone en Android hebben andere widgets, het aanpassen van de look gaat hierdoor (te) veel tijd kosten. Een andere factor die belangrijk is is de performance van de frameworks. Dit kan doorslaggevend zijn om toch voor native of Xamarin te gaan.

Native look

Indien er een native look nodig is, dan kunnen alle frameworks gebruikt worden. Hierbij is het wel zo dat bij bepaalde frameworks, zoals Enyo en Sencha, aangepast moeten worden om de native look te krijgen.

Als de look echter perfect moet zijn, dan is native ontwikkelen of het gebruik van Xamarin het beste. Deze leveren de native look van het platform met de snelheid zoals een gebruiker verwacht van zijn apps.

Code

Als er een grote codebase aan de back-end wordt verwacht, is het handig om een multi platform framework te gebruiken.

Als er een kleine codebase verwacht wordt en de native look belangrijk is, dan kan native de beste oplossing zijn. Hiervoor is geen investering nodig in de vorm van een licentie en de snelheid wordt behouden.

Als er een grote codebase verwacht wordt en de native look is belangrijk, dan is het de vraag hoe belangrijk de performance is. Als er wordt verwacht dat de app snel is, dan is Xamarin de beste oplossing. Indien de performance echter niet belangrijk is, dan kan elk ander framework worden gebruikt.

Conclusie

Afhankelijk van de keuze met betrekking tot de look, de code en de performance, kan er een keuze worden gemaakt tussen de frameworks. Deze keuze is echter niet makkelijk. De keuze die gemaakt wordt is tijdens het ontwikkelen doorgaans ook niet makkelijk om aan te passen. Omdat de ontwikkelingen in de IT zo snel gaan is het wel aan te raden om te kijken welke mogelijkheden er op het moment zijn.

Wel heb ik, om het de lezer makkelijk te maken, ook een web-app gemaakt waarmee het framework kan worden geselecteerd. [47] Hiermee kunnen de benodigde features worden gekozen en toont de pagina de frameworks die compatibel zijn. De gebruiker kan dan kiezen welk framework hij/zij wil kiezen.

Discussie

De gebouwde apps zijn allemaal dezelfde tool. Het betreft een app waarin een login functionaliteit, een invoer omgeving en een complexe berekening in zit. Ook is hierbij de opstartsnelheid gemeten. Hierdoor bevat het genoeg functionaliteiten om de geschiktheid van het framework te testen. Dit onderzoek test echter niet of de frameworks grafisch intensieve apps aan kunnen, zoals games. Dit heeft te maken met de beperkte tijd in het onderzoeksproject.

Ook is er niet gewerkt met iOS, wat toch een van de grote spelers van smartphones is. Dit is tijdens het opzetten van het onderzoek al bepaald, de verantwoording staat ook in het hoofdstuk Introductie onder Scope, maar het had meer inzicht kunnen bieden in de frameworks. Het tegenargument hierover is dat ik minder diep in de frameworks had kunnen duiken of dat ik tijd tekort zou hebben.

In de laatste weken van het onderzoek is er een nieuw Android thema voor jQuery uitgekomen. Aangezien het onderzoek in een te ver gevorderd stadium was, kon deze niet meer worden meegenomen. Mijn eerste indruk is positief. Het kan zijn dat dit thema beter is dan wat ik heb gebruikt. [48]

Dublin-descriptoren

Kennis en inzicht

De ontwikkelingen in de IT gaan erg snel. Zeker met een platform dat zeer snel groeit, namelijk Windows Phone 8. Zelfs in de laatste weken waren er nieuwe bronnen die ik graag had willen verwerken in de scriptie, maar het was, helaas, te laat.

Tijdens het afstuderen is wel meerdere malen gekeken of er nieuwe frameworks beschikbaar waren en ook of er nieuwe CSS templates beschikbaar waren. Niet elk framework was bruikbaar binnen het onderzoek, door technische of visuele redenen, maar ik ben ervan overtuigd dat ik de geschikte frameworks eruit heb gehaald.

Ook bij het onderzoeken heb ik meerdere malen overwogen welke onderzoeksmethode het beste was voor die onderzoeksvraag. Hier is ook gekeken naar waarom andere methodes niet geschikt waren en of ik methodieken kon combineren.

Oordeelsvorming

Voordat onderzoek begon is eerst onderzocht hoe het beste de onderzoeksvragen kon worden onderzocht. De methodieken zijn gebruikt staan beschreven in het boek *Researching Information Systems and Computing* door Briony J Oates.

Tijdens het onderzoek heb ik kritisch gekeken naar de mogelijkheden van de frameworks. De mogelijkheden en beperkingen zijn vastgesteld en de geschikte frameworks zijn gebouwd en getest.

Communicatie

Tijdens het onderzoek bleek dat ik volgens mijn begeleider te weinig communiceerde met de andere medewerkers. Na deze opmerking ben ik ermee aan de slag gegaan, ik ben betrokken geweest met gesprekken en discussies binnen het bedrijf. Ook heb ik op de chatroom mijn mening gegeven op momenten dat iemand feedback vroeg, indien iets bijdroeg aan de vraag.

Leervaardigheden

Ik ben van mening dat dit onderzoek een bijdrage levert aan Lunatech. Ik ben erop gewezen dat mijn communicatie binnen Lunatech niet voldoende was. Hier ben ik toen ook mee aan de slag gegaan.

Ben ik nu de perfecte ICT-er? Het simpele antwoord hierop is nee. Ik ben wel van mening dat ik een waardevolle aanvulling ben in IT bedrijven, maar elke IT-er moet altijd door blijven leren. De ontwikkelingen gaan erg snel en ik merkte dit ook tijdens het maken van mijn scriptie. Als ik dezelfde hoofd- en deelvragen over een jaar zou onderzoeken, dan kom ik wellicht op andere conclusies en andere frameworks uit. Afhankelijk van de tijd die ik in de toekomst heb, wil ik dit onderzoek aanvullen met nieuwe frameworks en nieuwe tests. Hiermee draag ik bij aan mijn eigen kennis en de kennis van anderen.

Verantwoording hoger cijfer

Voor Lunatech is dit een belangrijk document. Dit omdat het voor het bedrijf makkelijker is om een juiste keuze te maken voor een framework. Deze keuze kan van invloed zijn op de ontwikkeltijd, look & feel en performance van de app. Omdat de juiste keuze voor een framework resulteert tot een betere app, is dit belangrijk voor de gebruikers van de app en dus ook de klanten van Lunatech. De klanten krijgen immers meer waar voor hun geld.

Niet alleen voor Lunatech is dit een belangrijk document, ook voor andere bedrijven die een keuze willen maken kan dit document, alsmede het keuzemodel een hoop helpen. Hierdoor is de opdracht in een grotere context te plaatsen.

Tevens ben ik van mening dat ik in een complexe omgeving ben afgestudeerd. Hoewel er kennis van Android binnen het bedrijf is, was er geen kennis over de meeste van de frameworks en er was geen kennis van Windows Phone. Dit betekent dat er binnen het onderzoek niet veel hulp van binnen het bedrijf mogelijk was.

Verklarende woordenlijst

Android

Een op Linux gebaseerd besturingssysteem voor smartphones en tablets. Ontwikkeld door Google en grotendeels open-source. [49]

API

API is een afkorting voor Application Program Interface. Het is een set van routines, protocollen en tools om software applicaties mee te maken. Hiermee is het makkelijker om een programma te bouwen. [50] [51]

App

Een veel voorkomende afkorting voor application. In dit document gaat het om een applicatie gemaakt voor mobiele platformen. Deze kunnen worden geïnstalleerd en zullen op zichzelf kunnen draaien. [52]

C#

C# is een programmeer taal ontwikkeld door Anders Hejlsberg om voor Windows te programmeren in het .NET framework. Qua syntax en stijl lijkt C# erg op Java. C# is gebaseerd op Java. [38] [39]

CamelCase

CamelCase, ook wel pascal notation, is het schrijven van woorden aan elkaar waarbij de woorden opgesplitst worden door het gebruik van hoofdletters bij de eerste letter van elk woord [53]. (Voorbeeld: GetSomeCookies)

Een variatie hierop is lowerCamelCase/camel notation. Dit is het gebruik van CamelCase waarbij de eerste letter geen hoofdletter is. [54] (Voorbeeld: getSomeCookies)

Contante Waarde

De contante waarde is de huidige waarde van een bedrag waarover je pas na een bepaalde periode de beschikking hebt. Er wordt rekening gehouden met de tijdsperiode en de rente die misgelopen word. [1]

Cross-platform/multi-platform

De termen cross-platform en multi-platform hebben in de loop der tijd verschillende definities gekregen.

1. De mogelijkheid om software of hardware te laten werken op meer dan een platform met identieke of bijna identieke functionaliteiten [55]
2. De mogelijkheid om in een keer software te maken voor concurrerende platformen in een bepaalde programmeertaal zonder dit opnieuw te schrijven. De software kan kan draaien op meerdere platformen waarbij slechts enkele of geen wijzigingen nodig zijn [56] [57]
3. Een term dat een taal, software applicatie of hardware apparaat beschrijft dat werkt op meer dan 1 systeem platform [58]

Bij dit document zal term 2 worden bedoeld aangezien het het meest aansluit bij mijn visie over cross-platform development.

Framework

Een vooraf gedefinieerde applicatie structuur. Hierbij kunnen op vooraf gedefinieerde plekken functionaliteiten worden toegevoegd/uitgebreid. Vaan er ook geïntegreerde libraries. [59]

GUI

Graphical User Interface. Een manier om te communiceren met een computer door icoontjes en vensters te manipuleren. Voor de GUI was het gebruikelijk om met computers om te gaan door tekst invoer. [60]

Hybride app

Hybride apps is een mix tussen een mobiele web-app en een native app. De web-app wordt in de app geladen waardoor het de web-browser rendering technologie gebruikt.

iOS

Een op MacOS-gebaseerd besturingssysteem voor de iPod, iPad, iPhone en andere producten van Apple. [61]

Java

Java is een programmeer taal die ontwikkeld is door Sun Microsystems (later overgenomen door Oracle). Java is ontwikkeld als object georiënteerde programmeertaal. Java is gebaseerd op C++. [62] [63]

JDK

Java Development Kit. Dit is de SDK voor Java.

Kernel

De basis van een besturingssysteem. De kernel kan de hardware direct aanspreken. [64]

Library

Een collectie subroutines en functies opgeslagen in een of meerdere bestanden om te gebruiken in applicaties. [65] [66]

Linux

Een, op basis van UNIX gebaseerde, kernel. De besturingssystemen die hierop worden gebouwd worden ook Linux genoemd. Door de stabiliteit is het gebruik van Linux erg populair bij internet service providers, universiteiten en kleine bedrijven. [67]

Enkele voorbeelden zijn Debian, Fedora en Ubuntu.

Look & Feel

De term Look & Feel is goed gedefinieerd [68] [69] [70]:

Het algemene uiterlijk en de werking van een user interface.

1. De user interface van een besturingssysteem, applicatie of web pagina.
2. De look is hoe de logo's, graphics, menu's en andere elementen op de pagina staan.
3. De feel gaat over de interacties: de manier waarop menu's zijn georganiseerd en de manier waarop functies worden geselecteerd en uitgevoerd.

Het betreft dus hoe de applicatie eruit ziet en hoe het werkt.

Native

Een native applicatie is een applicatie die de technieken van het specifieke platform gebruikt. Native applicaties kunnen direct de CPU en GPU aanroepen in plaats van dit bijvoorbeeld door de browser te laten lopen. Hierdoor kunnen zij sneller uitgevoerd worden en “voelen” zij sneller. Een Windows Phone applicatie zal bijvoorbeeld in C# worden geschreven en een Android applicatie in Java. Het voordeel hiervan is dat het over het algemeen sneller is en de native applicaties kunnen de API's van de apparaten aanroepen. [71]

De Dictionary of Computer and Internet Terms beschrijft native als iets wat ontworpen is voor specifieke hardware of software omgeving. [72]

In dit document wordt het woord “native” gebruikt als het gaat om een van de twee native frameworks “Silverlight” of “Android Native Framework”.

Library

Een collectie subroutines en functies opgeslagen in een of meerdere bestanden om te gebruiken in applicaties. [65] [66]

Object Georiënteerd programmeren

Object-georiënteerd programmeren is een programmeer methodologie waarin de programmeur niet alleen de data types, maar ook de methodes kan definiëren die erbij horen. Een algemeen type van een object heet een klasse. Zodra een klasse is gedefinieerd kunnen daar meerdere instanties van worden aangeroepen. [73] [74]

Opmaaktaal/Markup Language

Een taal waarin de opmaak van een pagina kan worden opgebouwd. [75] [76]

Pivot

Een pivot is te vergelijken met een tab in Android. Hiermee kunnen views of pages worden beheerd en gekozen. [77]

Preprocessor Directives

Preprocessor Directives zijn statements die voor de compilatie al worden uitgevoerd. Bij een #if statement kan bijvoorbeeld conditioneel code worden gebruikt of niet. Als wordt bepaald dat de conditie van het #if statement niet voldoet dan zal dit niet gecompileerd worden en dus ook niet uitgevoerd kunnen worden. [78]

Zie ook Bijlage C: Preprocessor Directives.

SDK

SDK is een afkorting voor Software Development Kit. Een SDK is een programmeer pakket dat programmeurs het mogelijk maakt om software te maken voor een platform. Hieronder vallen vaak een of meerdere API's, programmeer tools en documentatie. [79] [80]

Trident

Trident is een web browser engine. Het is de basis voor Internet Explorer. De Cordova versie voor Windows Phone wordt hierin gedraaid.

Web browser engine

Ook bekend als *layout engine* of *rendering engine*. Een web browser engine is een software onderdeel die ervoor zorgt dat de opmaaktaal zoals HTML en de stylesheet weergegeven worden.

WebKit

WebKit is een web browser engine. Het is de basis voor verschillende browsers zoals Safari en Chrome. De Cordova versies voor Android en iOS worden hierin gedraaid.

Windows Phone

Een mobiel besturingssysteem voor smartphones. Het is de opvolger van Windows Mobile. Windows Phone 7 gebruikt de Windows CE kernel. Windows Phone 8 gebruikt de Windows NT kernel. [81]

Verwijzingen

- [1] „Wat is Contante Waarde / Definitie van Contante Waarde / Wat betekent Contante Waarde,” [Online]. Available: <http://www.definancieelinformatiebank.nl/wat-is-de-definitie-van-contante-waarde>. [Geopend 11 6 2013].
- [2] „Windows Phone SDK 8.0,” [Online]. Available: <http://www.microsoft.com/en-us/download/details.aspx?id=35471>. [Geopend 11 6 2013].
- [3] „Dashboards | Android Developers,” [Online]. Available: <http://developer.android.com/about/dashboards/index.html>. [Geopend 10 7 2013].
- [4] „AdDuplex Blog: Windows Phone Device Stats for February 2013,” [Online]. Available: <http://blog.adduplex.com/2013/02/windows-phone-device-stats-for-february.html>. [Geopend 26 2 2013].
- [5] „AdDuplex Blog: AdDuplex Windows Phone Statistics Report for June 2013,” [Online]. Available: <http://blog.adduplex.com/2013/06/adduplex-windows-phone-statistics.html>. [Geopend 10 7 2013].
- [6] B. J. Oates, „Internet Research,” in *Researching Information Systems and Computing*, Sage, 2012.
- [7] „Scriptie Mitchel van Zuidam,” 26 11 2012. [Online]. Available: http://natschool.hro.nl/CMS/___OAI/HR%20Kennisbank/Scripties/Instituut_CMI/Informatica/2012/AV_Zuidam_van_Mitchel_0811910_INF_26-11-2012_/AV_Zuidam_van_Mitchel_0811910_INF_26-11-2012_.pdf. [Geopend 4 4 2013].
- [8] B. J. Oates, "Experiments," in *Researching Information Systems and Computing*, Sage, 2012.
- [9] B. J. Oates, „Questionnaires,” in *Researching Information Systems and Computing*, Sage, 2012.
- [10] „Design | Android Developers,” 17 6 2013. [Online]. Available: <http://developer.android.com/design>.
- [11] „Design library for Windows Phone,” 6 5 2013. [Online]. Available: [http://msdn.microsoft.com/en-US/library/windowsphone/design/hh202915\(v=vs.105\).aspx](http://msdn.microsoft.com/en-US/library/windowsphone/design/hh202915(v=vs.105).aspx). [Geopend 17 6 2013].
- [12] B. J. Oates, "Action Research," in *Researching Information Systems and Computing*, Sage, 2012.
- [13] „Windows Phone 8 Support – Cross Platform Dev Blog,” 30 10 2012. [Online]. Available: <http://trigger.io/cross-platform-application-development-blog/2012/10/30/windows-phone-8-support/>. [Geopend 14 3 2013].
- [14] S. Almog, „Issue 619 in codenameone: Windows Phone app crashes with default tab application,” Codename One, 27 3 2013. [Online]. Available: <http://code.google.com/p/codenameone/issues/detail?id=619>. [Geopend 27 3 2013].
- [15] „Native UI EditText JavaScript | MoSync Forum Topic,” 20 5 2012. [Online]. Available: <http://www.mosync.com/content/native-ui-editbox-javascript>. [Geopend 30 04 2013].

- [16] „Feature/Platform Support | MoSync Guide/Tutorial,” 18 12 2012. [Online]. Available: <http://www.mosync.com/widepage/feature-platform-support>. [Geopend 1 5 2013].
- [17] „NativeUI data-fontSize | MoSync Forum Topic,” 21 5 2013. [Online]. Available: <http://www.mosync.com/content/nativeui-data-fontsize>. [Geopend 21 5 2013].
- [18] S. S. Ponmalar, „Forums | Stand-alone apps? | Syncfusion,” Syncfusion, 25 3 2013. [Online]. Available: <http://www.orubase.com/support/forums/orubase/107917/Stand-alone-apps?>. [Geopend 15 4 2013].
- [19] „Generating private keystore issue · Issue #6 · rhomobile/rhostudio,” 26 3 2013. [Online]. Available: <https://github.com/rhomobile/rhostudio/issues/6>. [Geopend 26 3 2013].
- [20] „Pure Android | Android Developers,” [Online]. Available: <http://developer.android.com/design/patterns/pure-android.html>. [Geopend 10 6 2013].
- [21] „Why LinkedIn dumped HTML5 & went native for its mobile apps,” 17 4 2013. [Online]. Available: <http://venturebeat.com/2013/04/17/linkedin-mobile-web-breakup/>. [Geopend 11 7 2013].
- [22] „Zuckerberg admits fault in Facebook’s HTML5 focus, pledges native Android app,” 11 9 2013. [Online]. Available: <http://www.androidcentral.com/zuckerberg-admits-fault-facebook-s-html5-focus-pledges-native-android-app>. [Geopend 11 7 2013].
- [23] „Facebook to produce a native Android app,” 11 9 2013. [Online]. Available: http://www.phonearena.com/news/Facebook-to-produce-a-native-Android-app_id34339. [Geopend 11 7 2013].
- [24] „Official Uber app for Windows Phone pulled from Store,” 28 4 2013. [Online]. Available: <http://www.wpcentral.com/official-uber-app-windows-phone-pulled-store>. [Geopend 10 6 2013].
- [25] J. Avins, „[OPEN] ST2.2RC WP8 Cordova - icon font not shown,” 18 4 2013. [Online]. Available: <http://www.sencha.com/forum/showthread.php?261159>. [Geopend 4 6 2013].
- [26] 29 11 2012. [Online]. Available: <http://msdn.microsoft.com/en-us/library/windows/apps/hh465373.aspx>. [Geopend 13 6 2013].
- [27] „Resources | jQuery Mobile,” [Online]. Available: <http://jquerymobile.com/resources/>. [Geopend 15 4 2013].
- [28] S. Grebnov, „sgrebnov / jqmobile-wp8-theme,” [Online]. Available: <https://github.com/sgrebnov/jqmobile-wp8-theme>. [Geopend 14 3 2013].
- [29] „enathu/jqmobile-android-holo-light-theme,” [Online]. Available: <https://github.com/enathu/jqmobile-android-holo-light-theme>. [Geopend 9 4 2013].
- [30] „jjoe64/jquery-mobile-android-theme,” github, [Online]. Available: <https://github.com/jjoe64/jquery-mobile-android-theme>. [Geopend 9 4 2013].

- [31] „zeickan/Holo-Holo-Theme,” github, [Online]. Available: <https://github.com/zeickan/Holo-Holo-Theme>. [Geopend 4 9 2013].
- [32] „Holo CSS,” [Online]. Available: <http://vezquex.com/projects/holo-css/index.html>. [Geopend 9 4 2013].
- [33] „zmyaro/holo-web,” github, [Online]. Available: <https://github.com/zmyaro/holo-web>. [Geopend 9 4 2013].
- [34] „Cannot swipe with web browser control in pivot item,” 24 2 2011. [Online]. Available: <http://social.msdn.microsoft.com/Forums/en-US/wpdevelop/thread/87ccb29b-59ce-4d1f-83df-4f1c4b546017/>. [Geopend 10 4 2013].
- [35] „JavaScript API for Native widgets on iOS, Android | MoSync Guide/Tutorial,” MoSync, 22 2 2012. [Online]. Available: <http://www.mosync.com/documentation/manualpages/jsnativeui-library>. [Geopend 10 4 2013].
- [36] S. (. M. E. Pickard, „Layout Add Combobox | MoSync Forum Topic,” 5 11 2010. [Online]. Available: <http://www.mosync.com/content/layout-add-combobox>. [Geopend 11 4 2013].
- [37] „C# Preprocessor Directives,” Microsoft, [Online]. Available: [http://msdn.microsoft.com/en-us/library/ed8yd1ha\(v=vs.110\).aspx](http://msdn.microsoft.com/en-us/library/ed8yd1ha(v=vs.110).aspx). [Geopend 11 4 2013].
- [38] „C# from FOLDOC,” 04 10 2001. [Online]. Available: <http://foldoc.org/c%23>. [Geopend 16 4 2013].
- [39] „C#,” in *Dictionary of Computer and Internet Terms*, Barron's, 2013.
- [40] „Computes the code share stats for the iCircuit project,” 13 1 2012. [Online]. Available: <https://gist.github.com/praeclarum/1608597>. [Geopend 18 4 2013].
- [41] „præclarum - iCircuit Code Reuse, Part Cinq,” 5 2 2013. [Online]. Available: <http://praeclarum.org/post/42378027611/icircuit-code-reuse-part-cinq>. [Geopend 18 4 2013].
- [42] „TouchDraw Code Reuse - Updated — Jon Lipsky,” 11 9 2012. [Online]. Available: <http://lipsky.me/blog/2012/9/11/touchdraw-code-reuse-updated>. [Geopend 18 4 2013].
- [43] „#10660 (no transport error on Windows Phone 7),” 4 11 2011. [Online]. Available: <http://bugs.jquery.com/ticket/10660>. [Geopend 7 3 2013].
- [44] „Client-side Cross-domain Security,” 6 2008. [Online]. Available: [http://msdn.microsoft.com/en-us/library/cc709423\(v=vs.85\).aspx](http://msdn.microsoft.com/en-us/library/cc709423(v=vs.85).aspx). [Geopend 27 5 2013].
- [45] „Apache Cordova API Documentation,” 1 3 2013. [Online]. Available: http://cordova.apache.org/docs/en/2.5.0/cordova_connection_connection.md.html#Connection. [Geopend 10 4 2013].
- [46] „MSDN,” Microsoft, 6 3 2013. [Online]. Available: <http://msdn.microsoft.com/en-us/library/windowsphone/develop/hh184840%28v=vs.105%29.aspx>. [Geopend 2013 4 9].

- [47] „MJHeijster/FrameworkPicker,” 19 6 2013. [Online]. Available: <https://github.com/MJHeijster/FrameworkPicker>. [Geopend 19 6 2013].
- [48] „nativeDroid a jQuery Mobile Theme,” 2013. [Online]. Available: <http://natedroid.godesign.ch/>. [Geopend 4 6 2013].
- [49] „Android,” in *Dictionary of Computer and Internet Terms*, Barron's, 2013.
- [50] „API,” in *Dictionary of Computer and Internet Terms*, Barron's, 2013.
- [51] „What is API? - A Word Definition From Webopedia.com,” Webopedia, [Online]. Available: <http://www.webopedia.com/TERM/A/API.html>. [Geopend 8 5 2013].
- [52] „App,” in *Dictionary of Computer and Internet Terms*, Barron's, 2013.
- [53] „Pascal notation,” in *Dictionary of Computer and Internet Terms*, Barron's, 2013.
- [54] „camel notation,” in *Dictionary of Computer and Internet Terms*, Barron's, 2013.
- [55] „Cross-platform Definition,” 15 december 2005. [Online]. Available: <http://www.linfo.org/cross-platform.html>. [Geopend 26 februari 2013].
- [56] „What is cross-platform? definition and meaning,” [Online]. Available: <http://www.businessdictionary.com/definition/cross-platform.html>. [Geopend 26 februari 2013].
- [57] „cross-platform,” in *Dictionary of Computer and Internet Terms*, Barron's, 2013.
- [58] „cross-platform from FOLDOC,” 24 februari 1998. [Online]. Available: <http://foldoc.org/cross-platform>. [Geopend 26 februari 2013].
- [59] „Framework,” 12 11 2012. [Online]. Available: <http://docforge.com/wiki/Framework>. [Geopend 28 03 2013].
- [60] „graphical user interface (GUI),” in *Dictionary of Computer and Internet Terms*, Barron's, 2013.
- [61] „iOS,” in *Dictionary of Computer and Internet Terms*, Barron's, 2013.
- [62] „Java,” in *Dictionary of Computer and Internet Terms*, Barron's, 2013.
- [63] „Java from FOLDOC,” 21 8 2011. [Online]. Available: <http://foldoc.org/Java>. [Geopend 16 4 2013].
- [64] „kernel,” in *Dictionary of Computer and Internet Terms*, Barron's, 2013.
- [65] „library from FOLDOC,” 21 november 1998. [Online]. Available: <http://foldoc.org/library>. [Geopend 26 februari 2013].
- [66] „Library,” in *Dictionary of Computer and Internet Terms*, Barron's, 2013.
- [67] „Linux,” in *Dictionary of Computer and Internet Terms*, Barron's, 2013.

- [68] „Look and feel Definition from PC Magazine Encyclopedia,” [Online]. Available: http://www.pcmag.com/encyclopedia_term/0,1237,t=look+and+feel&i=46319,00.asp. [Geopend 26 Februari 2013].
- [69] „look and feel from FOLDOC,” 3 maart 1995. [Online]. Available: <http://foldoc.org/look+and+feel>. [Geopend 2 februari 2013].
- [70] „What is look-and-feel? - A Word Definition From the Webopedia Computer Dictionary,” [Online]. Available: http://www.webopedia.com/TERM/L/look_and_feel.html. [Geopend 26 februari 2013].
- [71] E. W. John Daintith, „Native Software,” Oxford, 2004. [Online]. Available: <http://www.oxfordreference.com/view/10.1093/acref/9780199234004.001.0001/acref-9780199234004-e-3435?rskey=S1Z0aa&result=3012&q=>. [Geopend 16 4 2013].
- [72] „Native,” in *Dictionary of Computer and Internet Terms*, Barron's, 2013.
- [73] „object-oriented programming,” 11 10 2001. [Online]. Available: <http://foldoc.org/object-oriented+programming>. [Geopend 16 4 2013].
- [74] „Object-oriented programming,” in *Dictionary of Computer and Internet Terms*, Barron's, 2013.
- [75] „markup language,” in *Dictionary of Computer and Internet Terms*, Barron's, 2013.
- [76] „markup from FOLDOC,” 14 12 2012. [Online]. Available: <http://foldoc.org/markup>. [Geopend 18 4 2013].
- [77] „Pivot control for Windows Phone,” Microsoft, [Online]. Available: [http://msdn.microsoft.com/en-us/library/windowsphone/develop/ff941098\(v=vs.105\).aspx](http://msdn.microsoft.com/en-us/library/windowsphone/develop/ff941098(v=vs.105).aspx). [Geopend 9 4 2013].
- [78] S. Allen, „C# Directives,” Dot Net Perls, [Online]. Available: <http://www.dotnetperls.com/directive>. [Geopend 11 4 2013].
- [79] „SDK,” in *Dictionary of Computer and Internet Terms*, Barron's, 2013.
- [80] „What is SDK? - A Word Definition From the Webopedia Computer Dictionary,” [Online]. Available: <http://www.webopedia.com/TERM/S/SDK.html>. [Geopend 8 5 2013].
- [81] „What is Windows Phone? A Definition from the Webopedia Computer Dictionary,” [Online]. Available: http://www.webopedia.com/TERM/W/windows_phone.html. [Geopend 22 4 2013].
- [82] „Source File Conventions (C# vs. Java),” 2005. [Online]. Available: [http://msdn.microsoft.com/en-US/library/ms228500\(v=vs.80\).aspx](http://msdn.microsoft.com/en-US/library/ms228500(v=vs.80).aspx). [Geopend 16 4 2013].
- [83] „Main () and Other Methods,” 2008. [Online]. Available: [http://msdn.microsoft.com/en-US/library/ms228506\(v=vs.80\).aspx](http://msdn.microsoft.com/en-US/library/ms228506(v=vs.80).aspx). [Geopend 16 4 2013].
- [84] „Data Types (C# vs. Java),” 2005. [Online]. Available: [http://msdn.microsoft.com/en-US/library/ms228360\(v=vs.80\).aspx](http://msdn.microsoft.com/en-US/library/ms228360(v=vs.80).aspx). [Geopend 16 4 2013].

- [85] F. McCown, „Java and C# Comparison,” 2 1 2011. [Online]. Available: http://www.harding.edu/fmccown/java_csharp_comparison.html. [Geopend 16 4 2013].
- [86] „Java Practices -> Immutable objects,” [Online]. Available: <http://www.javapractices.com/topic/TopicAction.do?Id=29>. [Geopend 16 4 2013].
- [87] Z. Paracha, „Java String comparison. The difference between == and equals().,” 23 2 2010. [Online]. Available: <http://www.zparacha.com/java-string-comparison/>. [Geopend 16 4 2013].
- [88] „Controlling Access to Members of a Class,” Oracle, [Online]. Available: <http://docs.oracle.com/javase/tutorial/java/javaOO/accesscontrol.html>. [Geopend 17 4 2013].
- [89] „Accessibility Levels (C# Reference),” MSDN, [Online]. Available: [http://msdn.microsoft.com/en-us/library/ba0a1yw2\(v=vs.110\).aspx](http://msdn.microsoft.com/en-us/library/ba0a1yw2(v=vs.110).aspx). [Geopend 17 4 2013].
- [90] B. J. Oates, „Documents,” in *Researching Information Systems and Computing*, Sages, 2012.
- [91] „NativeUI data-fontSize | MoSync Forum Topic,” 21 5 2013. [Online]. Available: <http://www.mosync.com/content/nativeui-data-fontsize>. [Geopend 21 5 2013].
- [92] „Android UI Patterns: Dont's of Android design,” 17 6 2013. [Online]. Available: <http://www.androiduipatterns.com/2013/06/donts-of-android-design.html>. [Geopend 20 6 2013].
- [93] „Dos and Don'ts of Android Design - xda-developers,” 20 6 2013. [Online]. Available: <http://www.xda-developers.com/android/dos-and-donts-of-android-design/>. [Geopend 20 6 2013].

Bijlage A: Mobiele Rekentools

Dit is de beschrijving van de originele ontwikkeling van de Mobiele Rekentools.

Wat is het

De mobiele rekentools zijn een set aan rekentools. Hieronder vallen:

- Rente (6:119; 6:119a, contractuele rente, invorderingsrente, heffingsrente (betaling), heffingsrente (vordering) en basisherfinancieringsrente)
- Liquidatietarief
- Contante Waarde
- Vastrecht
- Kantonrechttersformule
- Erven & schenken
- Termijnberekening
- Prijsindexcijfer

De gebruiker logt in met zijn gebruikersnaam en wachtwoord van Sdu Uitgevers en de API wordt gebruikt om de gebruiker te authenticeren en autoriseren.

Technologieën

Component	Nut
CoffeeScript	JavaScript vervanging (het compileert naar JavaScript)
Backbone.js	JavaScript MVC-framework
Cordova	Omhult de HTML5/JavaScript applicaties naar native mobiele applicaties
Play/Scala	Een web-applicatie framework
LESS	Een dynamische Stylesheet taal (het compileert naar CSS)
HTML	
Rhino	JVM JavaScript engine
JSON	Een op tekst gebaseerde standard om gegevens uit te wisselen
Globalize	JavaScript localisatie library

Wat ging goed

- Door de technologieën waren snelle en korte iteraties makkelijk
- De architectuur is makkelijk te begrijpen
- CoffeeScript is makkelijk te leren
- Cordova werkt zoals beschreven staat
- Play's zip archief deploiment is handig

Wat kon beter

- Het was niet duidelijk hoe Backbone correct kon worden gebruikt
- Backbone is moeilijk om consistent te gebruiken
- Het initiële design werkte niet op Android
- Het kost veel tijd om JavaScript te laten werken in Rhino
- De StyleSheet structuur is gebaseerd op een stylesheet voor alle platformen
- Er waren datum input/format problemen
- Input verwerking en validatie werd niet consistent afgehandeld

Bijlage B: Kwaliteitseisen

Applicaties

- De broncode wordt ingecheckt via GIT op de Lunatech repository
- Geteste en werkende software
- Alle code die op de GIT repository komt zal comments bevatten in de code die te begrijpen zijn door de programmeurs binnen Lunatech
- Volgens kwaliteitseisen code van Lunatech
- Volgens code conventies
- Wekelijkse updates

De code zal worden opgebouwd met logische namen voor methodes, variabelen en klassen. Alles functies die logischerwijs hergebruikt kunnen worden zullen in aparte methodes worden geschreven.

Documenten

- De documentatie zal in het Nederlands worden geschreven
- De documentatie zal worden goedgekeurd door Francisco Canedo van Lunatech, zowel taaltechnisch als inhoudelijk

Bijlage C: Preprocessor Directives

Uitleg Preprocessor Directives

Preprocessor Directives zijn statements die voor de compilatie al worden uitgevoerd. Bij een `#if` statement kan bijvoorbeeld conditioneel code worden gebruikt of niet. Als wordt bepaald dat de conditie van het `#if` statement niet voldoet dan zal dit niet gecompileerd worden en dus ook niet uitgevoerd kunnen worden. [78]

Voorbeeld om settings op te halen voor Windows Phone en Android

```
using System;
using System.IO.IsolatedStorage;
#if __ANDROID__
using Android.Content;
using Android.Preferences;
using Android.App;
#endif
namespace Contante_Waarde.Common
{
    class Get
    {
        #if WINDOWS_PHONE
            static readonly IsolatedStorageSettings Settings = IsolatedStorageSettings.Ap
plicationSettings;
        #elif __ANDROID__

            private static Context context = Application.Context.ApplicationContext;
            static ISharedPreferences prefs = PreferenceManager.GetDefaultSharedPreferenc
es (context);

        #endif

        /// <summary>
        /// Gets the string.
        /// </summary>
        /// <param name="value">The value.</param>
        /// <returns>The value from the settingsstorage.</returns>
        private static String GetString(String value)
        {
            String returnValue;
            #if WINDOWS_PHONE
                Settings.TryGetValue(value, out returnValue);
            #elif __ANDROID__

                returnValue = prefs.GetString (value, "");
                Console.Out.WriteLine("Get: " + value + " + " + returnValue);
            #endif

            return returnValue;
        }
    }
}
```

In dit voorbeeld zijn `__ANDROID__` en `WINDOWS_PHONE` definities die op true worden gezet als de app voor dat platform word gecompileerd.

Bijlage D: Overeenkomsten en verschillen tussen C# en Java

Java en C# lijken erg op elkaar qua syntax en stijl. Beiden zijn ook gebaseerd op C++. [38] [39] Er zijn echter wel verschillen. Daardoor kunnen de twee talen niet makkelijk code kunnen delen. In deze bijlagen staan enkele verschillen relevant worden geacht door de auteur voor deze scriptie.

Declaraties [82]

Bij Java wordt er gebruik gemaakt van packages. Elke klasse in een package ziet de andere packages. In C# wordt gebruik gemaakt van namespaces. Deze werken bijna hetzelfde als de packages.

Als er classes uit andere packages/namespaces moeten worden gebruikt kan dat. Met Java zal een package door middel van “import” aanroepbaar worden gemaakt. Bij C# gaat dit door middel van “using”.

Java	C#
<pre>package Acme; import java.io.*; class Customer { }</pre>	<pre>using System.IO; namespace Acme { class Customer { } }</pre>

In C# is het mogelijk meerdere namespaces te gebruiken in een enkele .cs bestand. Bij Java is elke klasse in een .java bestand altijd in dezelfde package.

<pre>namespace AcmeAccounting { public class GetDetails { } } namespace AcmeFinance { public class ShowDetails { } }</pre>

Methods [83]

Bij zowel Java als C# begint het programma bij de Main Method. Een verschil is dat de conventie van C# is dat methodes beginnen met een hoofdletter. In Java is dit met een kleine letter. Dit zorgt voor enkele dingen waar op gelet moet worden. Het begin van de app begint bijvoorbeeld als volgt:

Java	C#
<pre>public static void main(String[] args) { System.out.println(GetLanguage()); } public static String GetLanguage() { return "Java"; }</pre>	<pre>static void Main(string[] args) { Console.WriteLine(GetLanguage()); } public static String GetLanguage() { return "C#"; }</pre>

Datatypes [84] [85]

C# bevat alle primitives van Java. Het voegt er echter unsigned numerals en 128-bit high-precision floating-point types aan toe.

Het omzetten van datatypes gaat ook anders. Bijvoorbeeld:

Java	C#
<pre>// int to String int x = 123; String y = Integer.toString(x); // y is "123" // String to int y = "456"; x = Integer.parseInt(y); // x is 456 // double to int double z = 3.5; x = (int) z; // x is 3 (truncates decimal)</pre>	<pre>// int to string int x = 123; String y = x.ToString(); // y is "123" // string to int y = "456"; x = int.Parse(y); // or x = Convert.ToInt32 (y); // double to int double z = 3.5; x = (int) z; // x is 3 (truncates decimal)</pre>

In C# kunnen primitives ook nullable worden gemaakt. Dit kan bijvoorbeeld handig zijn bij booleans die in een database null kunnen zijn. Dit gaat door na de datatype een "?" te plaatsen. Bijvoorbeeld:

<pre>bool? x; if (x.HasValue) { System.Console.WriteLine(x.Value); } else { System.Console.WriteLine("Undefined"); }</pre>
--

Enums

Enums, of enumeraties, zijn gegroepeerde constanten. Bij C# is het mogelijk om eigen waarden toe te wijzen aan de enumeraties. Bij Java moet een extra methode worden geschreven in de enum. Een alternatief is het gebruiken van “dummy” enums.

Java	C#
<pre> public enum TermijnFactor { Jaarlijks(1), Halfjaarlijks(2), Kwartaal(4), Maandelijks(12); private int value; TermijnFactor(int value) { this.value = value; } public int getValue() { return value; } } </pre>	<pre> public enum TermijnFactor { Jaarlijks = 1, Halfjaarlijks = 2, Kwartaal = 4, Maandelijks = 12 } </pre>

Dit kan bijvoorbeeld handig zijn bij berekeningen. In het geval van de termijnfactor kan makkelijker worden gerekend.

Vergelijkingen

Bij Java is het niet mogelijk om 2 string waarden te vergelijken met de operators. In Java vergelijken de operators de interne referenties, niet de inhoud ervan. Omdat Strings in Java immutable zijn (ze kunnen niet veranderen na de constructie) [86] kunnen 2 strings naar dezelfde waarde refereren in het geheugen. Als er in het programma wordt geforceerd dat er een nieuwe String moet worden gemaakt met dezelfde waarde, dan zal deze een andere plek in het geheugen hebben. Dit terwijl het wel dezelfde waarde heeft. [87]

In C# worden de waarden gecontroleerd.

Java	C#
<pre> String x = "Java"; String y = "Java"; if (x.equals(y)) System.out.println("Succes!"); </pre>	<pre> String x = "C#"; String y = "C#"; if (x == y) Console.WriteLine("Succes!"); </pre>

Ref [83]

Een ref is in C# een variabele die als referentie word meegestuurd. Hierbij wordt deze geheugen locatie meegestuurd naar de methode die hem aan kan passen. Hier is geen equivalent van in Java.

```
class TestRef
{
    private static void Add(int i, ref int result)
    {
        result += i;
        return;
    }

    static void Main()
    {
        int total = 20;
        System.Console.WriteLine("Original value of 'total': {0}", total);

        Add(10, ref total);
        System.Console.WriteLine("Value after calling Add(): {0}", total);
    }
}
```

Output:

Original value of 'total': 20

Value after calling Add(): 30

Out [83]

Out lijkt erg op ref. Het verschil is dat bij out niet de originele waarde mee word gestuurd. Het geeft alleen aan waar het naartoe moet worden geschreven. Hier is geen equivalent van in Java.

```
class TestOut
{
    private static void Add(int i, int j, out int result)
    {
        // The following line would cause a compile error:
        // System.Console.WriteLine("Initial value inside method: {0}",
result);

        result = i + j;
        return;
    }

    static void Main()
    {
        int total = 20;
        System.Console.WriteLine("Original value of 'total': {0}", total);

        Add(33, 77, out total);
        System.Console.WriteLine("Value after calling Add(): {0}", total);
    }
}
```

Output:

Original value of 'total': 20

Value after calling Add(): 110

Properties

Bij C# zijn er properties. Bij de properties kunnen een get en set worden gebruikt. Het voordeel bij de C# get en set is dat bijvoorbeeld de set als private kan worden ingesteld. Bij Java moeten deze methodes apart worden gemaakt voor elke variabele door middel van getX en setX.

Verder kan eventueel de waarde worden veranderd bij een get/set. Bij Java zal er een aparte get methode en een aparte set methode moeten worden gemaakt.

Java	C#
<pre> public class Animal { private String name; public String getSpecies() throws Exception { return "The species is: " + name; } public void setSpecies(String value) throws Exception { name = value; } } public class TestAnimal { Animal animal = new Animal(); animal.setSpecies("Lion"); // set accessor System.out.println(animal.getSpecies ()); // get accessor } </pre>	<pre> public class Animal { private string name; public string Species { get { return "The species is: " + name; } set { name = value; } } } class TestAnimal { static void Main() { Animal animal=new Animal(); animal.Species="Lion";//set accessor System.Console.WriteLine(animal.Speci es); // get accessor } } </pre>

Access levels

Java heeft voor klassen drie access levels. [88] Dit zijn:

Public	Het is mogelijk om de klasse vanuit elke plek in de code te benaderen.
Protected	Hier kunnen alleen de klasse, de package en subclasses de klasse benaderen.
Package-private	Op deze manier kunnen alleen de klasse en de package de klasse benaderen.
Private	Hierbij kan alleen de klasse de code benaderen.

Hoewel C# package-private in naam mist, bestaat deze als protected internal. Deze biedt hetzelfde als package-private. C# heeft echter wel een extra access levels. [89] Dit betreft Internal.

Internal	De toegang is alleen voor de huidige assembly.
Protected internal	Een combinatie van protected en internal. De toegang is alleen voor de huidige assembly en subclasses.

Bijlage E: Gebruikte testapparaten

Voor de verschillende tests zijn altijd dezelfde apparaten gebruikt. Dit zijn:

Android

Voor de **Android** tests is het volgende toestel gebruikt:

Samsung Galaxy S IV

- Modelnummer: I9505
- Chipset: Qualcomm APQ8064T Snapdragon 600
- CPU: Quadcore 1.9 GHz Krait 300
- GPU: Adreno 320
- OS: Android 4.2.2 (JDQ39.I9505XXUAMDF)
- RAM: 2GB

Windows Phone

Voor de **Windows Phone 7** tests is het volgende toestel gebruikt:

Nokia Lumia 800

- Chipset: Qualcomm MSM8255 Snapdragon
- CPU: Single core 1.4GHz Scorpion
- GPU: Adreno 205
- OS: Windows Phone 7.8 (7.10.8862.114/1750.0805.8773.12220)
- RAM: 512MB

Voor de **Windows Phone 8** tests is de volgende telefoon gebruikt:

Nokia Lumia 820

- Chipset: Qualcomm MSM8960 Snapdragon
- CPU: Dual-core 1.5GHz Krait
- GPU: Adreno 225
- OS: Windows Phone 8.0 (8.010211.204/1232.5957.1308.0002)
- RAM: 1GB

Bijlage F: Windows Phone Back Button

Hello Jeroen,

Thank you for contacting Microsoft Windows Phone Developer Support. I see that you have questions regarding the use of the back button feature. You will need to design your app to work with the back-button navigation to pass through certification on both the Windows Phone 7 and 8 devices.

To assist you with this I have included some documentation links on the matter. Please let me know if you have any further questions.

[http://msdn.microsoft.com/en-us/library/windowsphone/develop/ff626521\(v=vs.105\).aspx](http://msdn.microsoft.com/en-us/library/windowsphone/develop/ff626521(v=vs.105).aspx)

[http://msdn.microsoft.com/en-us/library/windowsphone/develop/hh394012\(v=vs.105\).aspx](http://msdn.microsoft.com/en-us/library/windowsphone/develop/hh394012(v=vs.105).aspx)

Thank you,

Joe

Your Partners at Microsoft

Use what you know. Do what you've always imagined.

<removed>@microsoft.com

Hours: 8-5PM Central Time

Bijlage G: Meetresultaten

Windows Phone startup

Startup	1	2	3	4	5	Gemiddeld
Native	1,29	1,71	1,53	1,30	1,44	1,45
jQuery+Cordova	5,11	5,19	5,02	5,07	4,82	5,04
Sencha+Cordova	3,18	3,45	3,43	3,68	3,02	3,35
MoSync	4,55	5,47	5,31	6,19	4,81	5,27

Android startup

Startup	1	2	3	4	5	Gemiddeld
Native	0,99	0,51	0,61	0,62	0,71	0,69
jQuery+Cordova	2,07	1,14	1,25	1,50	1,30	1,45
Sencha+Cordova	5,50	3,32	4,08	5,26	4,00	4,43
MoSync	3,76	3,55	3,22	3,42	3,61	3,51
Xamarin	2,02	1,71	1,93	1,73	2,22	1,92

Windows Phone login

Login	1	2	3	4	5	Gemiddeld
Native	0,40	0,26	0,26	0,33	0,27	0,30
jQuery+Cordova	0,70	0,81	0,64	0,76	0,63	0,71
Sencha+Cordova	0,43	0,30	0,37	0,37	0,47	0,39
MoSync	0,35	0,43	0,31	0,38	0,36	0,37

Android login

Login	1	2	3	4	5	Gemiddeld
Native	0,35	0,41	0,49	0,34	0,35	0,39
jQuery+Cordova	0,28	0,17	0,2	0,17	0,14	0,19
Sencha+Cordova	0,57	0,5	0,26	0,3	0,29	0,38
MoSync	2,02	2,02	1,88	2,21	1,92	2,01
Xamarin	0,55	0,5	0,8	0,53	0,75	0,63

Windows Phone berekening

Bereken	1	2	3	4	5	Gemiddeld
Native	0,22	0,15	0,30	0,16	0,14	0,19
jQuery+Cordova	0,66	0,68	0,69	0,53	0,66	0,64
Sencha+Cordova	1,78	1,87	2,05	1,86	1,72	1,86
MoSync	1,72	1,6	1,55	1,66	1,58	1,62

Android berekening

Bereken	1	2	3	4	5	Gemiddeld
Native	0,23	0,32	0,30	0,38	0,30	0,31
jQuery+Cordova	0,16	0,19	0,19	0,18	0,17	0,18
Sencha+Cordova	1,23	1,58	1,12	1,08	1,03	1,21
MoSync	2,07	1,63	2,26	2,20	2,41	2,11
Xamarin	0,26	0,16	0,15	0,15	0,28	0,20