



Faculteit Bedrijf en Organisatie

Titel

Steven Stevens

Scriptie voorgedragen tot het bekomen van de graad van
professionele bachelor in de toegepaste informatica

Promotor:
Jan Janssens
Co-promotor:
Piet Pieters

Instelling: —

Academiejaar: 2018-2019

Tweede examenperiode

Faculteit Bedrijf en Organisatie

Titel

Steven Stevens

Scriptie voorgedragen tot het bekomen van de graad van
professionele bachelor in de toegepaste informatica

Promotor:
Jan Janssens
Co-promotor:
Piet Pieters

Instelling: —

Academiejaar: 2018-2019

Tweede examenperiode

Woord vooraf

Samenvatting

Inhoudsopgave

1	Inleiding	13
1.1	Probleemstelling	13
1.2	Onderzoeksvraag	14
1.3	Onderzoeksdoelstelling	14
1.4	Opzet van deze bachelorproef	14
2	Stand van zaken	15
2.1	Progressive web applicatie	15
2.1.1	Service worker	15
2.1.2	Manifest	16
2.2	Frameworks	17
2.2.1	Wat is een framework ?	17
2.2.2	Requirements	17

2.2.3	Gekozen frameworks	18
3	Methodologie	21
4	Conclusie	23
A	Onderzoeksvoorstel	25
A.1	Introductie	25
A.2	Stand van zaken	26
A.3	Methodologie	26
A.4	Verwachte resultaten	26
A.5	Verwachte conclusies	28
	Bibliografie	29

Lijst van figuren

A.1	Voorbeeld van boxplot die de snelheid vergelijkt	27
A.2	Voorbeeld van een tabel die de snelheid vergelijkt	27
A.3	Voorbeeld van histogram die de benodigde ruimte vergelijkt ...	27
A.4	Voorbeel van tabel die de benodigde ruimte vergelijkt	27

Lijst van tabellen

1. Inleiding

De inleiding moet de lezer net genoeg informatie verschaffen om het onderwerp te begrijpen en in te zien waarom de onderzoeksvraag de moeite waard is om te onderzoeken. In de inleiding ga je literatuurverwijzingen beperken, zodat de tekst vlot leesbaar blijft. Je kan de inleiding verder onderverdelen in secties als dit de tekst verduidelijkt. Zaken die aan bod kunnen komen in de inleiding (Pollefliet, 2011):

- context, achtergrond
- afbakenen van het onderwerp
- verantwoording van het onderwerp, methodologie
- probleemstelling
- onderzoeksdoelstelling
- onderzoeksvraag
- ...

1.1 Probleemstelling

Uit je probleemstelling moet duidelijk zijn dat je onderzoek een meerwaarde heeft voor een concrete doelgroep. De doelgroep moet goed gedefinieerd en afgeleid zijn. Doelgroepen als “bedrijven,” “KMO’s,” systeembeheerders, enz. zijn nog te vaag. Als je een lijstje kan maken van de personen/organisaties die een meerwaarde zullen vinden in deze bachelorproef (dit is eigenlijk je steekproefkader), dan is dat een indicatie dat de doelgroep goed gedefinieerd is. Dit kan een enkel bedrijf zijn of zelfs één persoon (je co-promotor/opdrachtgever).

1.2 Onderzoeksvraag

Wees zo concreet mogelijk bij het formuleren van je onderzoeksvraag. Een onderzoeksvraag is trouwens iets waar nog niemand op dit moment een antwoord heeft (voor zover je kan nagaan). Het opzoeken van bestaande informatie (bv. “welke tools bestaan er voor deze toepassing?”) is dus geen onderzoeksvraag. Je kan de onderzoeksvraag verder specificeren in deelvragen. Bv. als je onderzoek gaat over performantiemetingen, dan

1.3 Onderzoeksdoelstelling

Wat is het beoogde resultaat van je bachelorproef? Wat zijn de criteria voor succes? Beschrijf die zo concreet mogelijk. Gaat het bv. om een proof-of-concept, een prototype, een verslag met aanbevelingen, een vergelijkende studie, enz.

1.4 Opzet van deze bachelorproef

De rest van deze bachelorproef is als volgt opgebouwd:

In Hoofdstuk 2 wordt een overzicht gegeven van de stand van zaken binnen het onderzoeksdomein, op basis van een literatuurstudie.

In Hoofdstuk 3 wordt de methodologie toegelicht en worden de gebruikte onderzoekstechnieken besproken om een antwoord te kunnen formuleren op de onderzoeksvragen.

In Hoofdstuk 4, tenslotte, wordt de conclusie gegeven en een antwoord geformuleerd op de onderzoeksvragen. Daarbij wordt ook een aanzet gegeven voor toekomstig onderzoek binnen dit domein.

2. Stand van zaken

2.1 Progressive web applicatie

In deze sectie wordt er beschreven wat een progressive web applicatie allemaal extra nodig heeft naast een ontwikkelde website. **PWA_EXTRA_FEATURES PWA_EXTRA_FEATURES_2**

2.1.1 Service worker

Het eerste item dat een progressive web applicatie nodig heeft is een service worker. Deze zorgt ervoor dat de website alle aanvragen voor externe data kan opslaan, beter bekend als data caching. De aanvragen worden doorgestuurd naar de service worker die deze dan op zijn beurt haalt van de externe databron online of de lokale gecachte data. De service worker maakt deze beslissing op basis van de ingestelde strategie. Er zijn momenteel vijf strategieën die gebruikt kunnen worden met workbox **WORKBOX_STRATEGIES**. Deze strategieën zijn stale while revalidate, offline first, online first, network only, cache only.

Stale-While-Revalidate

Deze strategie zorgt ervoor dat de gebruiker zo snel mogelijk een antwoord heeft op de aangevraagde data. Dit gebeurt doordat de service worker eerst de gecachte data zal weergeven als deze beschikbaar is. Indien er geen gecachte data is zal de service worker de aanvraag via het netwerk doen. Met het antwoord van het netwerk zal de cache dan weer geüpdate worden met de laatste versie van de data.

Offline first

Bij deze strategie gaat de service worker kijken of de aangevraagde data al reeds gecached is op het apparaat. Als de data al reeds opgeslagen is op het apparaat, dan gebruikt de service worker de data die al reeds gecached is. Is er reeds geen data opgeslagen dan gaat de service worker via het netwerk de externe data opvragen. Eén van de grote nadelen van deze strategie is dat de externe data niet altijd up to date is.

Online first

Bij online first zal de service worker eerst de aanvraag doen via het netwerk. Als er geen netwerkverbinding is zal de service worker de gecachte data weergeven indien deze aanwezig is. Om deze reden moet er altijd een netwerkverbinding zijn om alle data binnen te halen voordat de applicatie offline de data kan weergeven. Met deze strategie is de externe data altijd up to date indien er een netwerkverbinding is.

Network Only

Network only is een strategie die de externe data enkel maar via het netwerk zal ophalen. Als deze niet beschikbaar is zal de gecachte data ook niet weergegeven worden.

Cache Only

Cache only werkt alleen maar met de gecachte data. Deze haalt geen externe data gaan via het netwerk. Deze strategie kan gebruikt worden voor sites die geen externe data nodig hebben zoals een rekenmachine.

2.1.2 Manifest

Het tweede item dat een progressive web applicatie nodig heeft is een manifest **MANIFEST**. Dit is een bestand met alle informatie die nodig is om de applicatie draaiende te houden eens deze geïnstalleerd is op het apparaat. Deze informatie bevat maar is niet gelimiteerd tot:

Auteur

De naam van de auteur van de applicatie. Deze kan ook de naam van het bedrijf zijn.

Icoon

Het icoon waarmee de applicatie wordt weergegeven bij de applicaties op het apparaat. Er kunnen verschillende iconen meegegeven worden. Deze kunnen dienen voor de verschillende besturingssystemen, zodat er bijvoorbeeld een uniek icoon is voor apple en android. De iconen kunnen ook een verschillende grootte hebben. Het aanbod van de verschillende

groottes van iconen zorgt ervoor dat het icoon altijd goed zichtbaar is op elk scherm van het apparaat.

Versie

De versie van applicatie kan ook ingesteld worden door de manifest file. Zo weet de gebruiker welke versie van de applicatie er geïnstalleerd is op zijn apparaat.

Omschrijving

Er kan ook een omschrijving van de applicatie meegegeven worden door de manifests file. Daardoor weet de gebruiker wat de applicatie inhoudt en wat hij of zij van de applicatie mag verwachten.

2.2 Frameworks

2.2.1 Wat is een framework ?

Een Framework is een geheel van softwarecomponenten dat gebruikt kan worden bij het programmeren van applicaties. Ook de afspraken over hoe die componenten gebruikt worden binnen een groep ontwikkelaars en welke code-standaarden en bibliotheken gebruikt worden kunnen onderdeel zijn van een framework. Het framework bepaalt welke software er binnen een organisatie wordt gebruikt en op welke manier. **WIKI_FRAMEWORK**

2.2.2 Requirements

De requirements voor de web applicatie zijn de volgende:

- Service worker
- Manifest
- Router
- Store
- Data binding

De requirements voor de native applicatie zijn de volgende:

- Cross-platform
- Data binding

2.2.3 Gekozen frameworks

Progressive web applicatie

Er zijn verschillende frameworks voor een progressive web applicatie te ontwikkelen, maar op basis van bovenstaande requirement is er gekozen voor Vue.js in combinatie met Nuxt.js voor het creëren van de progressive web applicatie. Andere mogelijkheden zijn **FRAMEWORKS_PROGRESSIVEWEBAPPS** :

Angular 2+

Deze heeft ook een package om de site naar een progressive web applicatie te transformeren. De reden waardoor angular niet geselecteerd werd is dat het relatief complex is en er al een kennis moet zijn van typescript voordat er een applicatie kan gebouwd worden.

React

React is geen officieel framework maar een library. Toch wordt deze opgenomen in de mogelijkheden omwille dat er een progressive web applicatie ontworpen kan worden met react. Deze werd niet geselecteerd omdat het relatief complex is in vergelijking met javascript. Daarnaast moet de programmeur een kennis van JSX hebben om react te kunnen gebruiken.

Ionic

Ionic werd niet geselecteerd omwille dat de applicatie frequent moet geupdate worden om mee te kunnen met de laatste veranderingen van het framework. Hierdoor kan de applicatie vrij snel verouderd worden. Als de site gebruik maakt van een verouderde package kan deze de site laten vastlopen.

Polymer

De reden waarom polymer niet geselecteerd werd is dat het niet search engine optimised is. Vervolgens heeft deze ook een hoge herlaadtijd voor de site.

Native applicatie

Voor de native applicatie te ontwikkelen hebben is er gekozen voor het framework kotlin. Andere frameworks die in aanmerking kwamen waren: Java en Xamarin

Java

Java is een populair framework onder de developers. Deze word nog altijd gebruikt voor allerhande applicaties, van bureaubladapplicaties tot android applicaties. Kotlin heeft vele

aspecten van java maar heeft deze beter gemaakt. Om die reden word de performantie hoger bij het gebruik van het Kotlin framework dan het Java framework. **KOTLIN_VS_JAVA**

Xamarin

Xamarin werd niet geselecteerd door dat de performantie op het android platform niet optimaal is. De performantie op ios is veel beter. Bijkomend zijn er maar een paar ondersteunende Integrated development environments (IDE's). Xamarin heeft ook geen ondersteuning voor android xml files die nodig zijn voor de opmaak. Om deze reden worden veel fouten gemaakt bij de xml bestanden tijdens de programmatie van de applicatie.

KOTIN_VS_XAMARIN

3. Methodologie

4. Conclusie

A. Onderzoeksvoorstel

Het onderwerp van deze bachelorproef is gebaseerd op een onderzoeksvoorstel dat vooraf werd beoordeeld door de promotor. Dat voorstel is opgenomen in deze bijlage.

A.1 Introductie

De native app kende de afgelopen jaren een enorme groei in aantal gebruikers en werd de voornaamste manier om apps te maken. Er was oorspronkelijk wel een groot nadeel, namelijk dat de app apart moest worden geprogrammeerd voor android en voor ios. Hierdoor waren er verschillende code bases waardoor het onderhoud van de app moeilijker werd. Dit werd later opgelost door cross-platform development. Het nadeel hiervan is dat de categorieën zich vandaag niet meer beperken tot smartphone en tablets. Daarnaast blijft de app ook kampen met onderhoudsproblemen, vanwege bepaalde features die enkel beschikbaar zijn op ios. Aldus werd er verder gezocht naar een oplossing. Die is er nu, namelijk de progressive web app (PWA). Dit is een app die je gemakkelijk kan installeren op zowel android als op ios. Hierdoor los je het probleem op van de native apps. Namelijk dat een PWA maar 1 code base heeft voor alle platformen(Experius, 2019). Deze bachelorproef beseert zich op de volgende onderzoeksvragen.

- Wat zijn de voordelen van PWA vs Cross-Platform Native Apps?
- Welke frameworks komen hiervoor in aanmerking?
- Wat is de impact op de gebruikerservaring en toegankelijkheid ?
- Zal de PWA de native app vervangen?

A.2 Stand van zaken

Er zijn reeds onderzoeken uitgevoerd die de verschillende voor- en nadelen van een PWA en van een Cross-Platform app met elkaar vergelijken. Aldus onderzoekt men de mogelijkheid om native apps te vervangen door progressive web apps. Deze onderzoeken staan beschreven in artikels zoals die van Marjchrzak, 2018, Osmani, 2017 en Steiner, 2018

Uit onderzoek van Osmani, 2017 blijkt dat een PWA de performantie toch wel kan verhogen in vergelijking met een native app. Dit omdat de ruimte voor de app te installeren veel kleiner is dan een native app. Daarentegen vond ik terug in artikel Marjchrzak, 2018 dat er nog geen zekerheid is of een native app kan vervangen worden door een PWA. Daarnaast stelt Marjchrzak dat PWA's nog niet supported zijn binnen het apple ecosysteem. Dit omdat safari nog geen ondersteuning biedt om een service worker te draaien op het systeem. Aangezien een service worker ervoor zorgt dat een PWA alle data cached, is deze essentieel voor het bouwen van een progressive web application. Als deze data terug moet opgehaald worden als er geen internet verbinding is, haalt hij deze van de service worker die het op zijn beurt ophaalt van de cache van het systeem. Hierdoor kan de PWA offline blijven werken eens alle data geladen is.

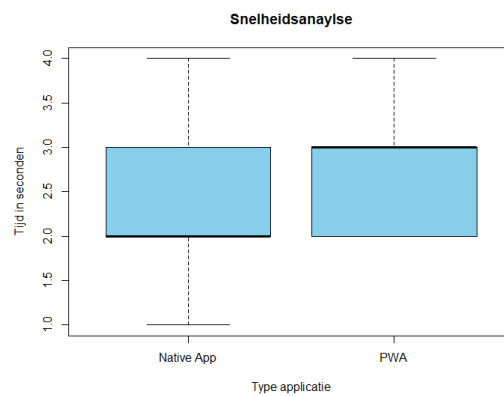
A.3 Methodologie

Om de performantie van de verschillende apps te vergelijken zal er twee maal eenzelfde applicatie gebouwd worden. Op android wordt deze gemaakt met framework kotlin, de PWA zal gemaakt worden met behulp van het framework vue.js in combinatie met nuxt. Op deze apps zullen verschillende soorten operaties uitgevoerd kunnen worden. Elk van deze operaties zal meermaals worden uitgevoerd en ondertussen zal de snelheid getest worden. Deze verkregen resultaten zullen vervolgens met elkaar vergeleken worden om te bepalen welke applicatie het meest performant is. Naast het onderzoek naar de performantie zal er ook onderzoek worden gedaan naar hoeveel ruimte deze app inneemt op een apparaat. Ook deze resultaten zullen worden vergeleken.

A.4 Verwachte resultaten

Om de resultaten voor te stellen wordt er gebruik gemaakt van een boxplot, zoals te zien is op figuur 1. Er zal er ook nog een tabel komen die de boxplot weergeeft. De getallen die hierin terug te vinden zijn: het gemiddelde, het maximum, het minimum, eerste kwadrant en derde kwadrant. Hierdoor krijgen we een goed overzicht om de resultaten te vergelijken tussen de PWA en native app.

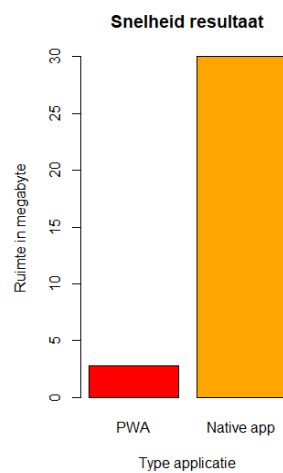
Voor de resultaten van de benodigde ruimte weer te geven zal gebruik gemaakt worden van een historiek, zoals te zien is op figuur 3. Er zal zoals bij de vergelijking van de tijd ook een tabel zijn met alle info die de historiek weergeeft.



Figuur A.1: Voorbeeld van boxplot die de snelheid vergelijkt

	min.	1ste kwa- drant	Gem.	3de kwa- drant	max.
PWA	2	2	2.714	3	4
native app	1	1	2.429	3	4

Figuur A.2: Voorbeeld van een tabel die de snelheid vergelijkt



Figuur A.3: Voorbeeld van historgram die de benodigde ruimte vergelijkt

Type applicatie	Benodigde ruimte
progressive web app	2.8 MB
native app	30

Figuur A.4: Voorbeel van tabel die de benodigde ruimte vergelijkt

A.5 Verwachte conclusies

Er wordt verwacht dat de snelheid niet significant verschillend zal zijn tussen de PWA en de native app. De native app haalt enkel de content op waardoor deze net iets sneller zal zijn. De PWA moet alle data ophalen samen met de layout, dit is de data moet weergegeven worden. Hierdoor zal de PWA een x tal milliseconden trager zijn dan de native applicatie

Verder wordt er verwacht dat de benodigde ruimte op een apparaat significant minder zal zijn dan de native app. Dit komt doordat de native app op het apparaat zelf geïnstalleerd is. Hierdoor staat er een standaard layout geïnstalleerd op het apparaat, wat redelijk veel ruimte inneemt. De PWA haalt zowel zijn inhoud als zijn layout van een server. Hierdoor kan een PWA de benodigde ruimte op het apparaat beperken.

Bibliografie

- Creeger, M. (2009). CTO Roundtable: Cloud Computing. *Communications of the ACM*, 52(8), 50–56.
- Experius. (2019). Progressive Web Apps: de toekomst voor webshops is hier. Verkregen van <https://www.experius.nl/pwa>
- Knuth, D. E. (1998). *The art of computer programming, volume 3: (2nd ed.) sorting and searching*. Redwood City, CA, USA: Addison Wesley Longman Publishing Co., Inc.
- Marjchrzak, T. A. (2018, januari 6). Progressive Web Apps: the Definite Approach to Cross-Platform Development?
- Osmani, A. (2017, december 24). A Tinder Progressive Web App Performance Case Study. Verkregen van <https://medium.com/@addyosmani/a-tinder-progressive-web-app-performance-case-study-78919d98ece0>
- Pollefliet, L. (2011). *Schrijven van verslag tot eindwerk: do's en don'ts*. Gent: Academia Press.
- Steiner, T. (2018, april 27). What is in a Web View? An Analysis of Progressive Web App Features When the Means of Web Access is not a Web Browser.