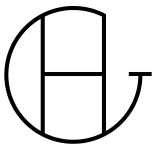


HoGent

Faculteit Bedrijf en Organisatie

Progressive webapp

Yannick Servranckx

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

Promotor:
Lotte Van Steenberghe

Instelling: —

Academiejaar: 2017-2018

Derde examenperiode

Faculteit Bedrijf en Organisatie

Progressive webapp

Yannick Servranckx

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

Promotor:
Lotte Van Steenberghe

Instelling: —

Academiejaar: 2017-2018

Derde examenperiode

Samenvatting

Tot hoe laat was de winkel weer open? Snel even mijn gsm pakken en naar de openingsuren kijken. Steeds meer en meer worden websites mobiel geopend in plaats van op desktop/laptop. Als bedrijf is het dan ook belangrijk dat men hierop inspeelt en zorgt dat hun gebruikers altijd en overal bij hen terecht kunnen. Hoe ga je als bedrijf hier mee om? Ga je voor een klassieke website die gebruikers zowel mobiel als op hun laptop kunnen bezoeken en die makkelijk kan gedeeld worden via een link? Of laat je de site achterwege en ga je voor een mobiele app?

Als bedrijf is het belangrijk te onderzoeken wat je klanten willen en op welke manier ze jouw product zoveel mogelijk gaan gebruiken. Wil je hen de app laten installeren, waarop veel mensen al zullen afhaken, zodat ze je app steeds makkelijk kunnen openen of wil je hen steeds laten terugsurfen naar je site? Door dit te onderzoeken kan je kijken op welke manier je product bij zoveel mogelijk klanten komt en kun je je hier eventueel aan aanpassen.

Native apps en web apps zijn twee verschillende types van applicaties. Native apps zijn specifiek voor elk platform. Voor zowel Apple, Android of Windows is er een andere programmeertaal nodig. Ze worden speciaal voor een bepaald platform ontwikkeld. Daardoor hebben ze verschillende functionaliteiten van een apparaat die ze kunnen gebruiken, zoals de camera, die een web app niet kan. Web apps daarentegen zijn platform onafhankelijk hierdoor hoeven ze maar eenmaal ontwikkeld te worden en kunnen direct op alle platformen bekeken worden. Web apps draaien op een browser en zijn meestal geschreven in HTML5.

Zowel het web als native apps hebben voor- en nadelen. Er was dus nood aan iets dat deze twee kon combineren en dit iets is progressive web apps. Door gebruik van service worker,

een app shell en web app manifest, lijkt en gedraagt een progressive web app zich als een native app. Aangezien progressive web apps een web app is kan deze op alle platformen draaien en hoeft het slechts eenmaal ontwikkeld te worden.

Het hoofddoel van deze thesis was het onderzoeken en implementeren van een progressive web app en te bekijken wat de belangrijkste onderdelen hiervan zijn. Hiervoor heb ik een progressive web app ontwikkeld die gaat belichten op welke manier een progressive web app een native als een web app combineert. Hiernaast heb ik ook gekeken wat de voor- en nadelen van een progressive web app zijn en deze vergeleken met een native app.

Een progressive web app heeft veel voordelen tegenover een standaard web app en is voor het web een grote stap vooruit. Het web groeit en zal altijd blijven groeien. Het is belangrijk als bedrijf hierop in te spelen. Je kan een nieuwe webapp maken en deze direct volledig als progressive web app maken. Het voordeel is dat je een huidge web app progressief kan omzetten naar een progressive web app. Je voegt enkel de functionaliteiten toe die je zelf wil of nodig hebt. Als eigenaar van een web app is het dus belangrijk te kijken welke mogelijkheden je hebt om eventueel één of meerdere functionaliteiten toe te voegen. Dit zal je web app enkel positief beïnvloeden.

Ondanks de verscheidene voordelen van een progressive web app tegenover een native app zal de web app nog niet direct de native app verdrijven van onze mobiele apparaten. Toch zullen progressive web apps steeds meer gebruikt worden en ingeburgerd raken, waardoor dit toch wel een vast gegeven zal worden in de toekomst van onze mobiele apparaten.

Voorwoord

Deze bachelorproef is tot stand gekomen in het kader van het behalen van het diploma Bachelor in Toegepaste Informatica.

Aangezien ik altijd veel interesse had in native apps heb ik voor progressive web apps als onderwerp gekozen. Native apps leek me altijd een groot voordeel te hebben tegenover web apps. Toen ik voor het eerst hoorde van progressive web apps leek dit me direct interessant aangezien dit het beste van twee werelden combineert. Misschien was dit wel de opvolger van native apps. Het was het dus zeker waard om dit eens beter te onderzoeken.

Een woord van dank aan mijn promotors Lotte Van Steenberghe en Johan Decorte. Mede dankzij hun feedback heb ik deze bachelorproef tot een goed einde kunnen brengen.

Daarnaast wil ik ook mijn vriendin, Devette Garza, bedanken om mij altijd aan te moedigen en in mij te blijven geloven.

Inhoudsopgave

1. Inleiding

"The full Safari engine is inside of iPhone. And so, you can write amazing Web 2.0 and Ajax apps that look exactly and behave exactly like apps on the iPhone. And these apps can integrate perfectly with iPhone services. They can make a call, they can send an email, they can look up a location on Google Maps. And guess what there is no SKD that you need. You got everything you need if you know how to write apps using the most modern web standards to write amazing apps for the IPhone today" (Jobs, 2007)

Steve Jobs schetste in zijn speech in 2007 al een idee omtrent wat we tegenwoordig progressive web apps ofwel PWA's noemen. Hierbij stelde hij Apple's internetbrowser Safari voor, waarop hij een idee gaf wat er allemaal mogelijk mee is. In 2008 introduceerde Apple de App Store waardoor het idee rond progressive web apps meer op de achtergrond raakte.

De voorbije jaren zijn er drie grote spelers op de mobiele markt geweest. Apple, Windows en Android. Elk van deze hebben hun eigen store waar ze apps aanbieden en waar je als ontwikkelaar je apps op kunt lanceren. Elke store werkt met een eigen programmeertaal.

- Android: Java
- Apple: Recent overgeschakeld van Objective-C naar Swift
- Windows: C#

Je hoeft je app niet op de store te plaatsen. Je kan dit ook aanbieden op je eigen website. Hiervoor moeten gebruikers je APK downloaden en installeren. Een 'Android Package Kit' of APK is de bestandsextensie die gebruikt wordt door het besturingssysteem van Android voor hun mobiele apps. Net zoals Windows .exe bestanden heeft, gebruikt Android .apk (Montegriffo, g.d.)

Als ontwikkelaar wil je geen drie verschillende programmeertalen leren. Als bedrijf wil je niet drie verschillende mensen huren voor hetzelfde werk te doen.

Op de Google I/O developer conference heeft Google als antwoord hierop de progressive web app voorgesteld. Google beschrijft de progressive web app als 'applicaties die gebruik maken van nieuwe technologieën om zo het beste van native apps en mobiele website naar de gebruiker te brengen. Ze zijn betrouwbaar en snel'. Met andere woorden: Google wil webapplicaties die de voordelen van native apps gaan gebruiken zodat de webapp zich gaat gedragen als een native app. Als ontwikkelaar zou je dan eenmaal je webapp moeten maken en kan deze op eender welk platform bekeken en gebruikt worden. Je zou geen verschillende programmeertalen moeten leren om dit voor elk platform apart te maken en toch zal het op elk platform lijken alsof het een native app is, speciaal gemaakt voor Android, Apple ...

1.1 Stand van zaken

Voordat we kunnen gaan kijken wat nu het beste alternatief is moeten we eerst begrijpen wat nu juist een PWA is en wat een native app is.

1.1.1 Wat is een native app?

Een native mobile app is een applicatie voor op de smartphone die gemaakt is voor gebruik op een specifiek platform in een specifieke programmeertaal (Rousse, g.d.). Doordat een app specifiek voor dat platform gemaakt is, kan de app gebruik maken van enkele interne technologieën van de smartphone zoals GPS, bluetooth, camera... Een goed voorbeeld hierbij is Pokemon Go. Pokemon Go maakt gebruik van vele verschillende functionaliteiten van je smartphone. Het gaat je camera gebruiken voor de Pokemons weer te geven in de echte wereld, je GPS gebruiken om te weten waar je bent, meten hoe snel je gaat ... Doordat Pokemon Go een native app is die speciaal gemaakt is voor het platform waarop het draait, dit kan zowel Apple als Android zijn, krijgt het de mogelijkheid om deze functionaliteiten te gebruiken.

Een ontwikkelaar die een native app maakt gaat afhankelijk van het platform een andere programmeertaal moeten gebruiken. Eenmaal zijn app geschreven is kan hij deze door gebruikers laten downloaden. Hij kan dit aanbieden op zijn eigen site via de APK ofwel via de store. Na installatie worden gegevens op de mobiele telefoon opgeslagen. De gebruiker kan nadien de app gebruiken zonder deze opnieuw te moeten installaren. (zie overzicht 3.1).

Voordelen

- Maximaal gebruik van alle functionaliteiten van het apparaat (camera, GPS, ...)
- Geen internetverbinding nodig
- Integratiemogelijkheden met andere apps
- Hogere snelheid op het apparaat

Nadelen

- Per platform moet apart ontwikkeld worden
- Goedkeuring voor plaatsing in de store
- Updates in software van het platform (bv. na een update van Android) is het mogelijk dat de app moet worden aangepast

1.1.2 Wat is een web app?

Steeds meer worden websites mobiel bezocht. In 2017 werden 37% van de websites bezocht via desktop en 63% mobiel.(3.3) Gebaseerd op deze gegevens lijkt het waarschijnlijk dat aan het einde van 2018 66% van alle bezoeken op websites via de mobiele telefoon zal zijn. (Enge, g.d.). Hierdoor is het dus belangrijk dat je je website optimaliseert voor mobiel gebruik. Een website die hiervoor is geoptimaliseerd is een webapp. Webapps moeten altijd bezocht worden via de browser. De gebruiker gaat naar zijn favoriete internetbrowser en kan via de link de webapp opzoeken en bezoeken. Webapps hoeven dus niet geïnstalleerd te worden. Met behulp van een bladwijzer kan een koppeling gemaakt worden op het home-scherm van je apparaat. (zie overzicht 3.1).

Voordelen

- Doordat het een webapp is, is het niet platformafhankelijk.
- Makkelijk te delen. Je deelt gewoon een link en de gebruiker moet niets installeren
- Gebruikers zien altijd de laatste versie zonder updates te hoeven downloaden van de app.

Nadelen

- Het kan niet alle functionaliteiten van je apparaat gebruiken
- Een webapp is niet vindbaar in de store

1.1.3 Web vs Native

Uit een onderzoek van comScore (2016) in 2016 is gebleken dat gebruikers 87% van hun tijd op de smarthphone of tablet doorbrengen op een mobiele app terwijl ze maar 13% van hun tijd doorbrengen op het web. Als je deze cijfers ziet, zou je je kunnen afvragen of het wel slim is je tijd te investeren in een web app. Is het niet beter je tijd te spenderen aan het ontwikkelen van native apps? Je hebt wat meer werk om het op elk platform apart te kunnen krijgen, maar als je cijfers ziet van de gebruikers lijkt het erop dat het wel zijn vruchten kan afwerpen. (zie figuur 3.1).

Ondanks dat de gebruiker gemiddeld meer tijd doorbrengt op een native app zien we in hetzelfde onderzoek van comScore (2016) dat de helft van alle gebruikers maar 1 app per maand downloadt. Dit is een zeer laag nummer als je weet dat er in het eerste kwartaal van 2018 ongeveer 7.133.500 apps beschikbaar waren over de verschillende stores. Als je

enkel kijkt naar degenen die wel apps downloaden merk je dat deze gemiddeld 3,5 apps per maand downloaden. Meer dan de helft van deze apps wordt gedownload door 13% van de smartphone eigenaars. (zie figuur 3.2).

Ondanks dat gebruikers meer tijd spenderen op apps dan op mobiele websites, zijn er maandelijks meer unieke gebruikers die gebruik maken van een website dan van een app. (zie figuur 3.4) Hieruit kunnen we concluderen dat gebruikers weliswaar meer tijd doorbrengen op apps, maar dat ze wel online actiever bezig zijn. Gebruikers gaan niet zoveel nieuwe apps downloaden, maar wel veel tijd doorbrengen op hun favoriete app. Gemiddeld brengen gebruikers 45% van hun tijd door op hun favoriete app en 75% met hun top 3 apps (zie figuur 3.5).

Als je een native app hebt, heb je veel concurrenten. Het is moeilijk om veel gebruikers te krijgen aangezien maar weinigen nieuwe apps downloaden. Toch zijn de gebruikers loyaal aan hun favoriete apps en spenderen ze er veel tijd aan.

1.1.4 PWA's

Een progressive web app gaat nog een stap verder. Een progressive web app is een web app die er net als een native app uitziet en zich net zoals een native app gaat gedragen. Als gebruiker kun je de app installeren op je apparaat waardoor je een snelkoppeling krijgt. De app zal ook offline gebruikt kunnen worden. Omdat het nog steeds een webapp is kan je deze gewoon via je internetbrowser vinden. Maar vanaf wanneer is je webapp een progressive web app?

Progressive web apps worden niet in een bepaalde code geschreven. Je kan elke (bestaande) webapp gaan aanpassen zodat ze progressief is. Dit hoef je niet in een keer te doen. Je gaat de functionaliteiten gaan implementeren die je wil of nodig hebt. Indien je later andere functionaliteiten hebt, kan je deze dan gaan toevoegen. Enkele belangrijke kenmerken van een progressive webapp zijn:

- Progressief:
  Een progressive webapp moet werken voor alle gebruikers, op elke browser, op elk platform en gebruik maken van de functionaliteiten van het apparaat. Hierbij zullen sommige browsers of nieuwere versies meer functionaliteiten aanbieden dan andere. Het is de bedoeling dat alles goed werkt, ook als je browser bepaalde functionaliteiten niet ondersteunt die je in je app gebruikt.

- Responsief:
  Een progressive webapp moet geschikt zijn voor elk apparaat, welk grootte ook. Of je webapp nu bezocht wordt op een klein scherm van je telefoon of op het scherm van je desktop, overal moet de kwaliteit van je app even goed zijn.

- Installeerbaar:
  Je kan een sneltoets opslaan op je home screen. Hierdoor kan de gebruiker met één klik terug naar de app navigeren.

- Gedraagt zich als native app:
  Wanneer je de app opent via de sneltoets gaat de progressive web app eruit zien als een native app ondanks dat dit nog steeds een webapp is.

- Onafhankelijk van internet:
  Een progressive web app kan met een slecht netwerk of volledig offline werken. Door het opslaan van bestanden op het apparaat zelf, ook wel cachen genoemd, kan de app offline blijven werken. Bij een slechte internetverbinding kan hij deze bestanden ook gebruiken zodat de gebruiker geen wachttijden heeft. Het verliezen van connectiviteit is geen probleem, maar een mogelijkheid en moet daarom zeer zorgvuldig worden opgevangen. De ontwikkelaar moet daarom zien dat zodra er internetconnectie is, bestanden steeds up-to-date zijn zodat een gebruiker nooit oude data ziet.

- Ontdekbaar:
  Dankzij het W3C-manifest is het identificeerbaar als app en kan het gevonden worden door zoekmachines

- Deelbaar:
  Je kan makkelijk een link delen zodat anderen je applicatie kunnen bekijken. Installatie is niet nodig.

- Veilig:
  Progressive web apps gebruiken een beveiligde HTTPS-verbinding zodat alle data veilig kan worden uitgewisseld.

- Push notificaties
  Je kan push notificaties sturen naar de gebruikers. Dit verhoogt de kans dat gebruikers terug keren naar je app. De push notificaties voelen aan als notificaties van echte apps.

1.2 Probleemstelling en Onderzoeksvragen

Als bedrijf wil je gemakkelijk gevonden worden door iedereen. Wat voor bedrijf je ook hebt, of je nu spelletjes maakt online, of je nu een bakkerij hebt of een groentenwinkel, een groot bedrijf met honderden werknemers of een eenmanszaak. Als potentiële klanten iets zoeken wat jij aanbiedt, dan wil je snel en eenvoudig gevonden worden. Maar hoe kan je dit het best doen?

Hulpvragen:

- Investeer je als bedrijf het beste in native of in progressive web apps?

1.3 Opzet van deze bachelorproef

De rest van deze bachelorproef is als volgt opgebouwd:

In hoofdstuk 2 wordt de methodologie toegelicht en worden de gebruikte onderzoekstechnieken besproken om een antwoord te kunnen formuleren op de onderzoeksvragen.

In hoofdstuk 3, 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. Methodologie

2.1 PWA Componenten

Om een progressive web app te maken zijn er enkele belangrijke componenten. Deze gaan er voor zorgen dat je een optimale user experience krijgt net zoals je een native app zou hebben.

- Application shell
- Web App Manifest
- Service Worker

2.1.1 Web App Manifest

Je web app manifest is een simpel JSON bestand waarin je aan de browser vertelt over je web app en hoe het zich moet gedragen als het geïnstalleerd wordt op het apparaat van de gebruiker. Zonder dit manifest zal Chrome nooit de optie geven aan de gebruiker om de app te installeren op het hoofdscherm (zie figuur 3.6)

Een typisch manifest ziet er als volgt uit:

```
1 {
2 "name": "Instagram als Progressive Web App",
3 "short_name": "PWAGram",
4 "icons": [
5 {
6 "src": "/src/images/icons/app-icon-48x48.png",
```

```
"type": "image/png",
"sizes": "48x48"
},
{
"src": "/src/images/icons/app-icon-96x96.png",
"type": "image/png",
"sizes": "96x96"
},
{
"src": "/src/images/icons/app-icon-144x144.png",
"type": "image/png",
"sizes": "144x144"
},
{
"src": "/src/images/icons/app-icon-192x192.png",
"type": "image/png",
"sizes": "192x192"
},
{
"src": "/src/images/icons/app-icon-256x256.png",
"type": "image/png",
"sizes": "256x256"
},
{
"src": "/src/images/icons/app-icon-384x384.png",
"type": "image/png",
"sizes": "384x384"
},
{
"src": "/src/images/icons/app-icon-512x512.png",
"type": "image/png",
"sizes": "512x512"
}
],
"start_url": "/index.html",
"scope": ".",
"display": "standalone",
"orientation": "portrait-primary",
"background_color": "#fff",
"theme_color": "#3f51b5",
"description": "Een simpele Instagram Clone.",
"dir": "ltr",
"lang": "nl-BE"
}
```

- name en short_name:
  Eén van deze twee moet zeker aanwezig zijn. Als beide voorzien zijn zal overal de short_name gebruikt worden waar de plaats beperkt is zoals op je hoofdscherm onder het icoon.

- icons
  Hier kun je je iconen definiëren. Je definiëert een lijst met je iconen. De browser zal dan de beste uitpakken voor hetgeen nodig is. Als de browser het icoon nodig heeft voor te tonen op je hoofdscherm zal hij hier hetgeen met de juiste afmetingen bepalen.

- start_url:
  Dit vertelt aan de browser welke startpagina getoond moet worden aan de gebruiker.

- scope:
  Welke pagina's zijn deel van de progressive web app en kunnen dus gebruik maken van de iconen en kleuren die zijn gedefiniëerd. Het is gebruikelijk om hier alle pagina's te kiezen. Dit kan door gewoon "."te zetten.

- display:
  Hier kun je specifiëren welk deel van de browser getoond moet worden. Je kan aan de gebruiker nog steeds de adresbar tonen of je kan ervoor kiezen om alles te verbergen zodat het meer als een native app aanvoelt.

Waardes	**Beschrijving**
fullscreen	Neemt het geheel van plaats in beslag op het scherm en toont geen andere elementen.
standalone	De webapp zal eruit zien als een echte native app. Het opent op een apart venster, apart van de browser en toont geen standaard browser elementen zoals de zoekbalk.
minimal-ui	Dit wordt niet ondersteund in chrome. Hierbij heb je net hetzelfde als bij fullscreen maar je hebt een minimaal aantal aan elementen die je kan zien zoals een back-knop.
browser	Het lijkt alsof je gewoon in de browser kijkt.

- background_color:
  De achtergrondkleur van de app. Dit wordt vooral getoond in laadschermen. Hiervoor gebruik je een hexadecimale code bv: fff

- theme_color:
  Je thema kleur. Net zoals native apps een thema kleur hebben die gaat bepalen welke kleur je balk aan de top van je scherm heeft. Ook hier wordt gebruik gemaakt van

een hexadecimale code

- description:
  Hier kan je een kleine samenvatting geven. Als een internetbrowser dit nodig heeft, zal hij dit hier vandaan halen. De gebruiker zal dit dus te zien krijgen.

- dir
  Dit is de richting van je tekst. Doorgaans wordt hier ltr gebruikt, left to right. Als je wil kan je hier ook rtl kiezen.

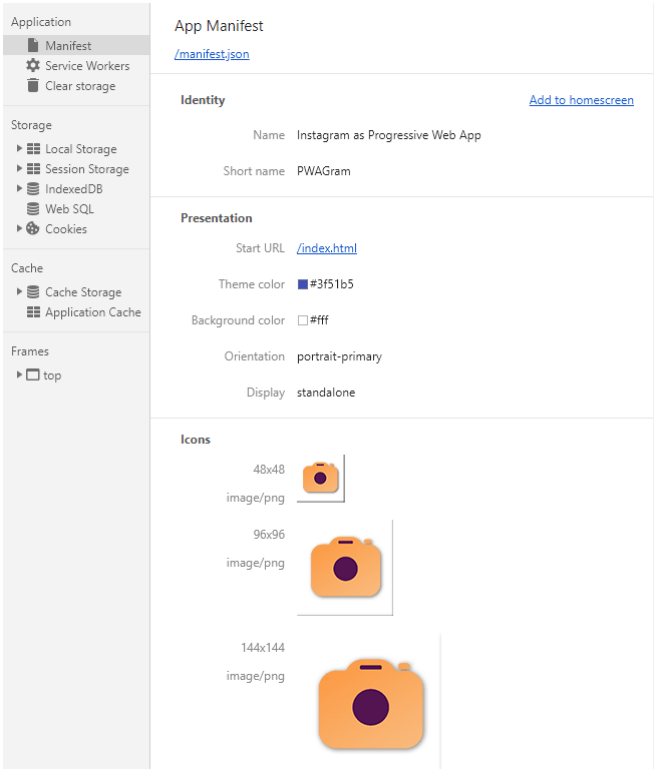
- lang
  De taal van je app: je gebruikt hier de 4 letter code van de taal die je wil.

- orientation
  Hier kan je bepalen hoe je wil dat je app getoond wordt. Wil je dit rechtop of liever gedraaid. Je kan hier aan de gebruiker opleggen dat je het scherm niet kunt draaien. vb: portrait-primary

Na het toevoegen van je manifest moet je aan al je pagina's de link naar je manifest toevoegen.

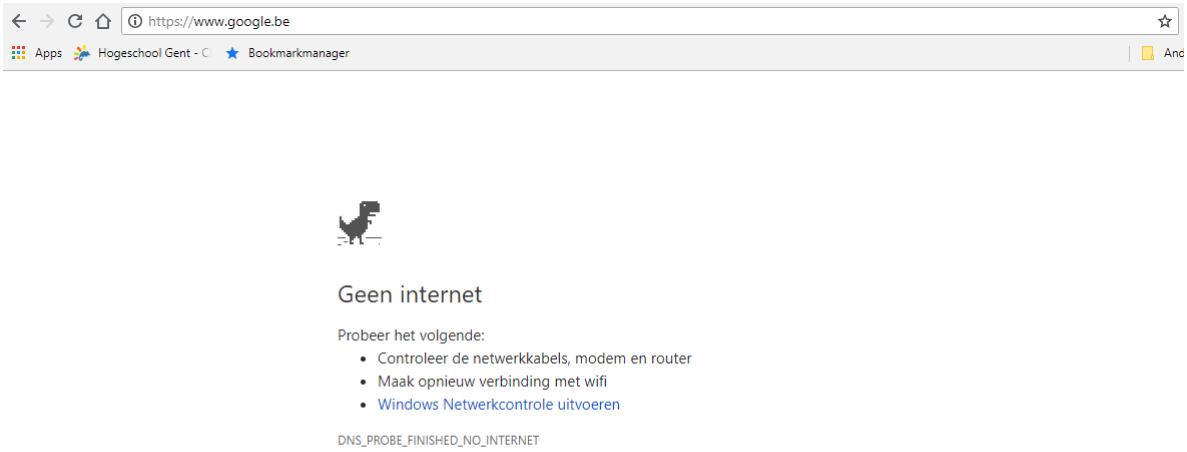
```
1 <link rel="manifest" href="/manifest.json">
```

Nadien kan je je manifest controleren in de developer tools van Google Chrome



2.1.2 Service Worker

Het web is handig. Wil je de openingsuren van een winkel weten of wil je opzoeken wat ze allemaal verkopen? Ga naar hun website en je vindt het meteen. Het is zo makkelijk. En toch heeft iedereen al eens hetzelfde meegemaakt als ze dit willen doen:

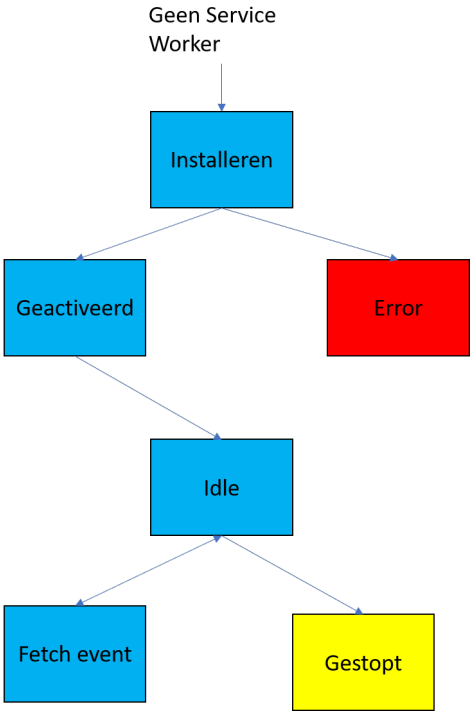


Zoals je ziet kun je niet veel zien als je geen internet hebt. Zonder internet kan je niet de informatie gaan opzoeken die je nodig hebt.

Door de introductie van de service worker is dit voor ontwikkelaars niet langer een probleem en kunnen ze dit op een juiste manier verwerken zodat de gebruikers niet merken dat ze geen internet hebben. Dit is de belangrijkste taak van de service worker maar lang niet de enige.

Een service worker is een script dat, apart van de webpagina, op de browser draait. Hierdoor zijn er verschillende functionaliteiten die gebruikt kunnen worden waarbij geen interactie van de gebruiker nodig is. Momenteel zijn dit functionaliteiten zoals push notificaties of offline gebruik van de site maar wie weet welke functionaliteiten in de toekomst nog mogelijk zullen worden gemaakt.

Aangezien een service worker apart loopt van de webpagina heeft het ook een andere levenscyclus.



Voordat de service worker kan geïnstalleerd worden moeten we deze eerst registreren.

```
1  if ('serviceWorker' in navigator) {
2  window.addEventListener('load', function() {
3  navigator.serviceWorker.register('/sw.js').then(function(
       registration) {
4  // Registration was successful
5  console.log('ServiceWorker registration successful with scope:
       ', registration.scope);
6  }, function(err) {
7  // registration failed :(
8  console.log('ServiceWorker registration failed: ', err);
9  });
10 });
11 }
```

Als je je service worker geregistreerd hebt, zal de browser deze direct installeren op de achtergrond. Je kan vanaf nu ook je service worker bekijken in de developer tools van Chrome (zie figuur 3.7).

Tijdens het installeren kan je in het script meegeven wat het moet doen. In de meeste gevallen zal hier ook het cachen van statiche bestanden gebeuren. Aangezien deze statisch zijn en weinig veranderingen nodig hebben hoeft dit eenmalig te gebeuren, tijdens de installatie. Als bestanden niet kunnen gecached worden dan zal de installatiestap falen en zal deze nooit geactiveerd worden. De volgende keer zal de browser opnieuw proberen de service worker te installeren.

```
1  var CACHE_NAME = 'my-site-cache-v1';
2  var urlsToCache = [
```

```
'/',
'/styles/main.css',
'/script/main.js'
];

self.addEventListener('install', function(event) {
// Perform install steps
event.waitUntil(
caches.open(CACHE_NAME)
.then(function(cache) {
console.log('Opened cache');
return cache.addAll(urlsToCache);
})
);
});
```

Nadat de service worker geïnstalleerd is wordt deze geactiveerd. Dit is de ideale plaats om oude caches te verwijderen en te zorgen dat de data die wel bewaard wordt up to date is.

Wanneer een service worker geactiveerd is wacht deze. Dit noemen we idle-state. De service worker wacht tot hij een taak toegewezen krijgt. Dit gebeurt als er een fetch-evenement wordt opgeroepen. In dit geval komt de service worker tot leven en zal hij deze taak uitvoeren waarna hij weer in een idle-state beland. De service worker zal in deze staat blijven tot deze wordt geüpdate. Hierdoor kan deze reageren op events zoals push-notificaties, zelfs als de web app gesloten is.

Je kan als ontwikkelaar zelf events toevoegen waarop de service worker zal reageren zoals het aanklikken van een notificatie

```
self.addEventListener('notificationclick', function (event) {
var action = event.action;

console.log(notification);

if (action === 'confirm') {
console.log('Confirm was chosen');
} else {
console.log('Another action was chosen');
}
});
```

Maar het belangrijkste event is het fetch event. Deze zal de dynamische inhoud van je webpagina laden. Door het juist implementeren van dit event kan je ervoor zorgen dat de gebruiker het niet merkt als hij geen internetverbinding zou hebben.

Offline gebruik

```
self.addEventListener('fetch', function (event) {
console.log('Haal de dynamische data op.');
});
```

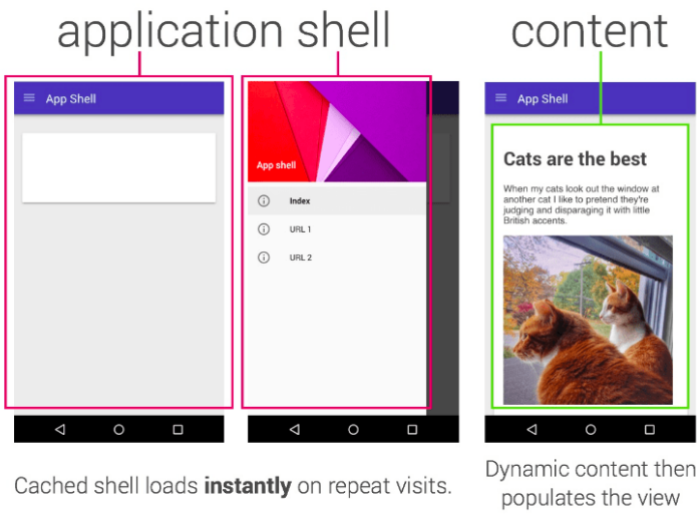
Het offline gebruiken van een progressive web app is één van de belangrijkste eigenschappen. Gebruikers zonder internetverbinding kunnen nog steeds je web app blijven gebruiken. Dit geldt ook voor gebruikers met een slechte verbinding. Ondanks hun verbinding zullen ze geen wachttijden te zien krijgen. Dit is als bedrijf heel goed aangezien gebruikers sneller zullen afhaken als ze moeten wachten tot een pagina geladen is.

Voor het ophalen van data zijn er enkele manieren waarop je te werk kan gaan. Afhankelijk van wat je wil zul je het fetch-event anders implementeren.

- Netwerk-only: Bij een fetch event zal de service worker de gegevens enkel ophalen over het netwerk. Hierdoor zal de gebruiker altijd de laatste gegevens hebben. Het offline gebruiken van de app zal niet mogelijk zijn aangezien het gegevens steeds over het internet wil ophalen.
- Cache-only: De gebruiker zal de gegevens altijd van de cache ophalen. Gegevens die op het netwerk worden aangepast zullen hier niet zichtbaar zijn aangezien ze nooit worden opgehaald. Dit is ideaal voor statische gegevens die nooit worden aangepast. Doordat het over de cache gaat en niet over het netwerk gaat dit veel sneller.
- Cache dan netwerk: De data wordt eerst opgehaald uit de cache en nadien via het netwerk. Eenmaal de gegevens van het netwerk zijn opgehaald wordt de pagina aangepast met de nieuwste data. Dit is ideaal waar data altijd up-to-date moet zijn zoals scoreborden en nieuwsartikelen.
- Cache met netwerk alternatief: De service worker zal de gegevens die hij daar terugvind ophalen uit de cache en alles waarvan hij een nieuwere versie vind op het netwerk van daar. Je zal hierdoor altijd de nieuwste versie hebben. Tijdens dit kan je ook je cache updaten waardoor de gebruiker de laatste versie in zijn cache heeft en de volgende keer hij die pagina bezoekt er minder moet worden opgehaald over het netwerk.
- Netwerk met cache alternatief: De service worker haalt de gegevens op via het netwerk. Hierdoor krijgt de gebruiker altijd de laatste gegevens te zien. Als de service worker hierbij problemen ondervind door een slechte verbinding zal hij de gegevens uit de cache ophalen. Hierdoor kan een gebruiker met een slechte verbinding lang moeten wachten voor hij data ziet.

2.1.3 Application shell

De application of app shell is het minimum aan HTML, CSS en Javascript dat je nodig hebt om je applicatie te kunnen tonen. Dit minimum wordt gecached zodat dit zelfs zonder internetverbinding altijd snel getoond kan worden aan de gebruiker. Hierdoor hoeft enkel de dynamische inhoud zoals een artikel geladen te worden via het netwerk. De app shell is heel handig om al iets te kunnen tonen aan de gebruiker terwijl andere delen nog tijd nodig hebben om te laden. Zoals je op afbeelding 2.1 ziet, heb je in de application shell het



Figuur 2.1: Application shell

minimum dat je wil tonen aan de gebruiker. Wanneer je de app ook bezoekt, zal dit altijd hetzelfde zijn. Aangezien hier weinig of geen verandering is, kan je dit lokaal opslaan (cachen). Het dynamisch gedeelte, het artikel, wordt wel via het netwerk geladen. Dit gedeelte, de inhoud, is niet altijd hetzelfde en zal vaak veranderen. Het is dus niet nodig dit lokaal op te slaan. Als er toch updates aan bestanden van de app shell worden gedaan, zal de progressive web app dit zien. De volgende keer dat de gebruiker online is, zal de app de oude bestanden vervangen door de nieuwe. Als ontwikkelaar is het belangrijk op voorhand te kijken welke bestanden je gaat cachen en welke niet. Belangrijke voorwaarden voor de app shell zijn:

- Snel laden
- Zo weinig mogelijk data gebruiken
- Statische bestanden gebruiken van de lokale cache
- Inhoud en navigatie scheiden

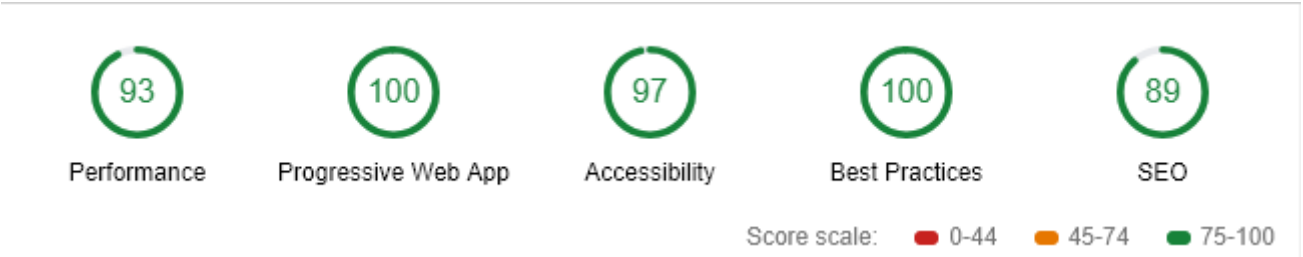
Dit is een voorbeeld van een app shell waarbij het sw.js bestand wordt gecachet.

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4  <meta charset="utf-8">
5  <title>App Shell</title>
6  <link rel="manifest" href="/manifest.json">
7  <meta http-equiv="X-UA-Compatible" content="IE=edge">
8  <meta name="viewport" content="width=device-width, initial-
       scale=1.0">
9  <link rel="stylesheet" type="text/css" href="styles/inline.css
       ">
10 </head>
11
12 <body>
```

```
13 <header class="header">
14 <h1 class="header__title">App Shell</h1>
15 </header>
16
17 <nav class="nav"></nav>
18
19 <main class="main"></main>
20
21 <div class="dialog-container"></div>
22
23 <div class="loader">
24 <!-- Show a spinner or placeholders for content -->
25 </div>
26
27 <script src="app.js" async></script>
28 <script>
29 if ('serviceWorker' in navigator) {
30 navigator.serviceWorker.register('/sw.js').then(function(
      registration) {
31 // Registration was successful
32 console.log('ServiceWorker registration successful with scope:
       ', registration.scope);
33 }).catch(function(err) {
34 // registration failed :(
35 console.log('ServiceWorker registration failed: ', err);
36 });
37 }
38 </script>
39 </body>
40 </html>
```

2.1.4 Lighthouse

Na het creëren van je progressive web app, kan je deze laten auditeren door lighthouse. Je progressive web app krijgt een rapport speciaal voor jou opgemaakt. Je krijgt hier een score (op 100) op verschillende punten). Het rapport toont ook waar je nog dingen kunt verbeteren.

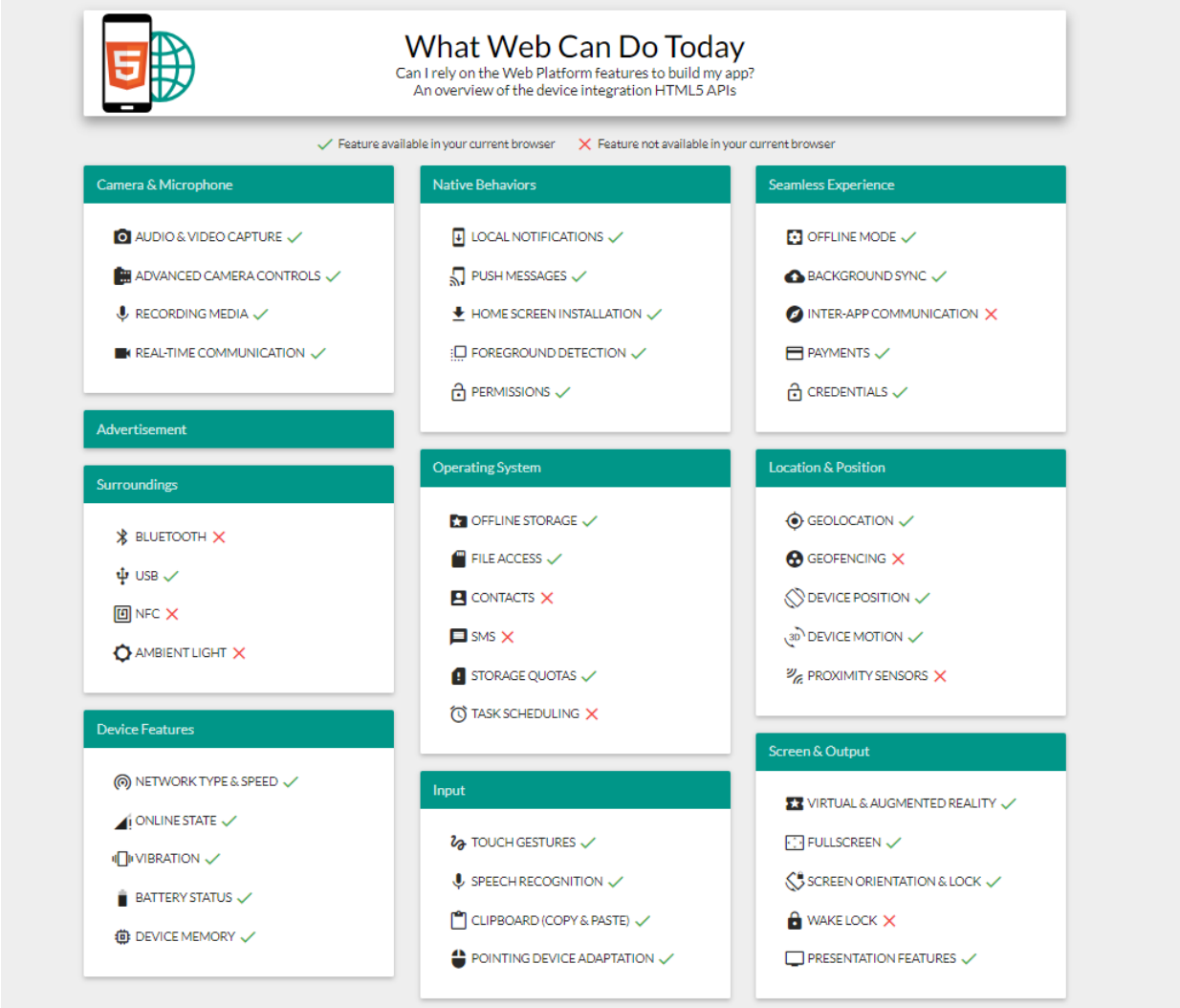


- Performance: Dit toont hoe goed je huidige app presteert. Dit zal onder andere meten hoe snel de pagina laad.

- Progressive web app: In welke mate voldoet de app aan de checklist waaraan een progressive web app moet voldoen. Dit gaat ondere andere kijken of er een pictogram is voor het hoofdscherm van een telefoon.
- Accessibility: Hoe toegankelijk is je app. Hier kan je een goede score krijgen door onder andere een goed kleurencontrast te gebruiken op je webpagina's.
- Best practices: Hou je je aan de best practices omtrent het schrijven van een web app. Hier wordt er onder andere gekeken of je geen error logs in de consoles toont.
- SEO: Is je pagina optimaal voor zoekmachines?

2.2 PWA vs Native

Met de komst van de progressive web app wordt de kloof tussen het web en native gedicht. Functionaliteiten van het apparaat die vroeger enkel door native apps konden gebruikt worden, zijn nu ook beschikbaar voor web apps.



Ondanks dat native en web dichter naar mekaar groeien zijn er nog steeds enkele verschillen tussen beiden. Zo hebben ze elk enkele voordelen tegenover de ander.

2.2.1 Voordelen native tegenover web

- Ondanks de vele functionaliteiten van het apparaat dat het web kan gebruiken zijn er toch nog enkele die enkel voor native voorbestemd zijn. Voor ontwikkelaars dat gebruik willen maken van deze specifieke functionaliteiten zullen voor een native app moeten kiezen
- App stores zijn niet altijd slecht. Gebruikers kunnen makkelijk nieuwe apps ontdekken. Ze kunnen er redelijk zeker van zijn dat de app die ze willen downloaden geen slechte bedoelingen heeft aangezien deze eerst gecontroleerd worden voor ze op de store geplaats kunnen worden. Als bedrijf kun je beoordelingen krijgen die je kan gebruiken om je app te verbeteren.
- Wake lock zorgt ervoor dat je scherm niet uitgaat na een bepaalde tijd van inactiviteit. Als je de Netflix-app gebruikt wil je niet elke vijf minuten je scherm aanraken zodat het niet uitvalt. Web apps kunnen deze wake lock niet gebruiken waardoor het zomaar kan gebeuren dat, in het midden van je artikel dat je aan het lezen bent, je telefoon uitvalt.

2.2.2 Voordelen web tegenover native

- Inhoud in progressive web apps kan makkelijk gevonden worden door zoekmachines. Voor native apps wordt er niet gekeken naar inhoud. Native apps zoals Reddit zul je enkel terugvinden als je zoekt naar Reddit, maar niet als je zoekt naar één van de vele subreddits die Reddit rijk is.
- Wil je een artikel delen die je hebt gezien in een online winkel? Bij een web app kan je makkelijke elke pagina linken en deze delen met anderen. Gebruikers hebben maar één link nodig om direct op een bepaalde pagina te komen. Bij native app moeten ze eerst via de app store om daar de app te downloaden. Nadien moeten ze nog navigeren naar de juiste pagina.
- Als ontwikkelaar moet je niet elke update laten goedkeuren door de app store maar kan je je web app updaten wanneer je wil.
- Vanaf dat je web app online is, is het bereikbaar voor iedereen. Voordat je een native app kunt gebruiken moet je enkele stappen doorlopen. Bij elk van deze stappen verlies je ongeveer 20% gebruikers. Als je bij 1000 mensen interesse hebt opgewekt voor je app, heb na alle stappen te doorlopen nog ongeveer 262 gebruikers over. (zie figuur 3.8)

3. Conclusie

Met de progressive web app is Google erin geslaagd het beste van native apps en web apps met mekaar te combineren. Hierdoor gaan ze de concurrentie aan met native apps.

De voordelen die progressive web apps ten opzichte van native apps hebben, zijn ondertussen duidelijk :

- Je beschikt altijd over de nieuwste versie bij het openen van de app.
- Je kan makkelijk een link delen met anderen
- Je hoeft niets te downloaden
- Platform onafhankelijk
- ...

Dankzij de voordelen en de extra functionaliteiten tegeover een gewone web app worden ze steeds populairder en meer gebruikt. Toch kampen ook progressive web apps met problemen. Ondanks dat ze platform onafhankelijk zijn en ze maar eenmalig moeten ontwikkeld worden, moet er rekening gehouden worden met verschillende browsers en browserversies. Momenteel worden progressive web apps niet op alle ondersteund.

Progressive web apps is iets nieuws. Hierdoor is het nog niet gekend. Hier zal verandering in komen. De progressive web app zal steeds meer ingeburgerd geraken bij de mensen waardoor ze na verloop van tijd zullen verwachten dat ze een website kunnen opslaan op hun hoofdscherm. Dit hoeft niet allemaal in één keer. Als bedrijf kan je bepalen welke aspecten je nodig hebt van een progressive web app en enkel deze implementeren in je web app. Zo kan je je web app stap voor stap progressiever maken.

Als winkel of bedrijf kan je er niet omheen. Klanten moeten je online kunnen vinden. Je hebt een plaats online nodig waar mensen de nodige informatie kunnen vinden over

jou. Momenteel is het zeker waard om te investeren in een progressive web app. Wil je toch in de store gevonden worden of gebruik maken van specifieke eigenschappen van een apparaat, dan heb je een native app nodig.

Toen ik begon aan deze thesis zag ik een progressive web app als een positieve evolutie van een web app. Tijdens dit onderzoek ben ik gaan inzien dat dit een gelijkwaardige rivaal van een native app kan zijn. Volgens mij is dit momenteel nog niet het geval aangezien het nog iets nieuw is. Toch denk ik dat, met meer browser-support, progressive web apps de toekomst kunnen zijn van het mobiele web.

Bibliografie

comScore. (2016). The 2016 U.S. Mobile App Report. Verkregen van https://www.comscore.com/Insights/Presentations-and-Whitepapers/2016/The-2016-US-Mobile-App-Report

Cselle, G. (g.d.). Every Step Costs You 20% of Users. Verkregen van https://medium.com/gabor/every-step-costs-you-20-of-users-b613a804c329

Enge, E. (g.d.). Mobile vs Desktop Usage in 2018: Mobile takes the lead. Verkregen van https://www.stonetemple.com/mobile-vs-desktop-usage-study/

Gaunt, M. (g.d.-a). Service Workers: an introduction. Verkregen van https://developers.google.com/web/fundamentals/primers/service-workers/

Gaunt, M. (g.d.-b). The Web App Manifest. Verkregen van https://developers.google.com/web/fundamentals/web-app-manifest/

Jobs, S. (2007). Steve Jobs keynote Macworld. Keynote van Apple waar Steve Jobs voor het eerst de voorloper van PWA voorstelt. Verkregen van https://9to5mac.com/2011/10/21/jobs-original-vision-for-the-iphone-no-third-party-native-apps/

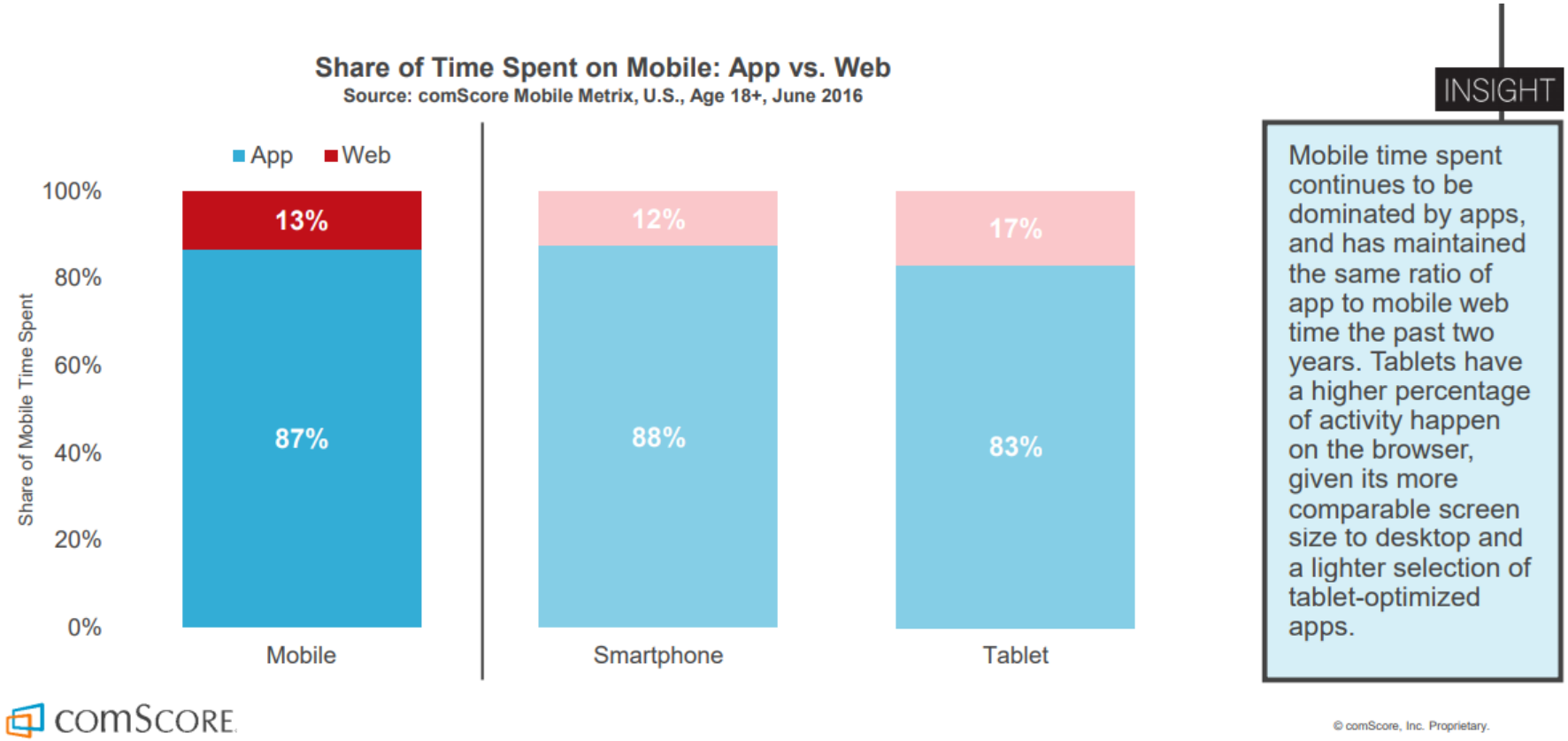
LePage, P. (g.d.). Your first progressive webapp. Verkregen van https://developers.google.com/web/fundamentals/codelabs/your-first-pwapp/

Montegriffo, N. (g.d.). What is an APK file and how do you install one. Verkregen van https://www.androidpit.com/android-for-beginners-what-is-an-apk-file

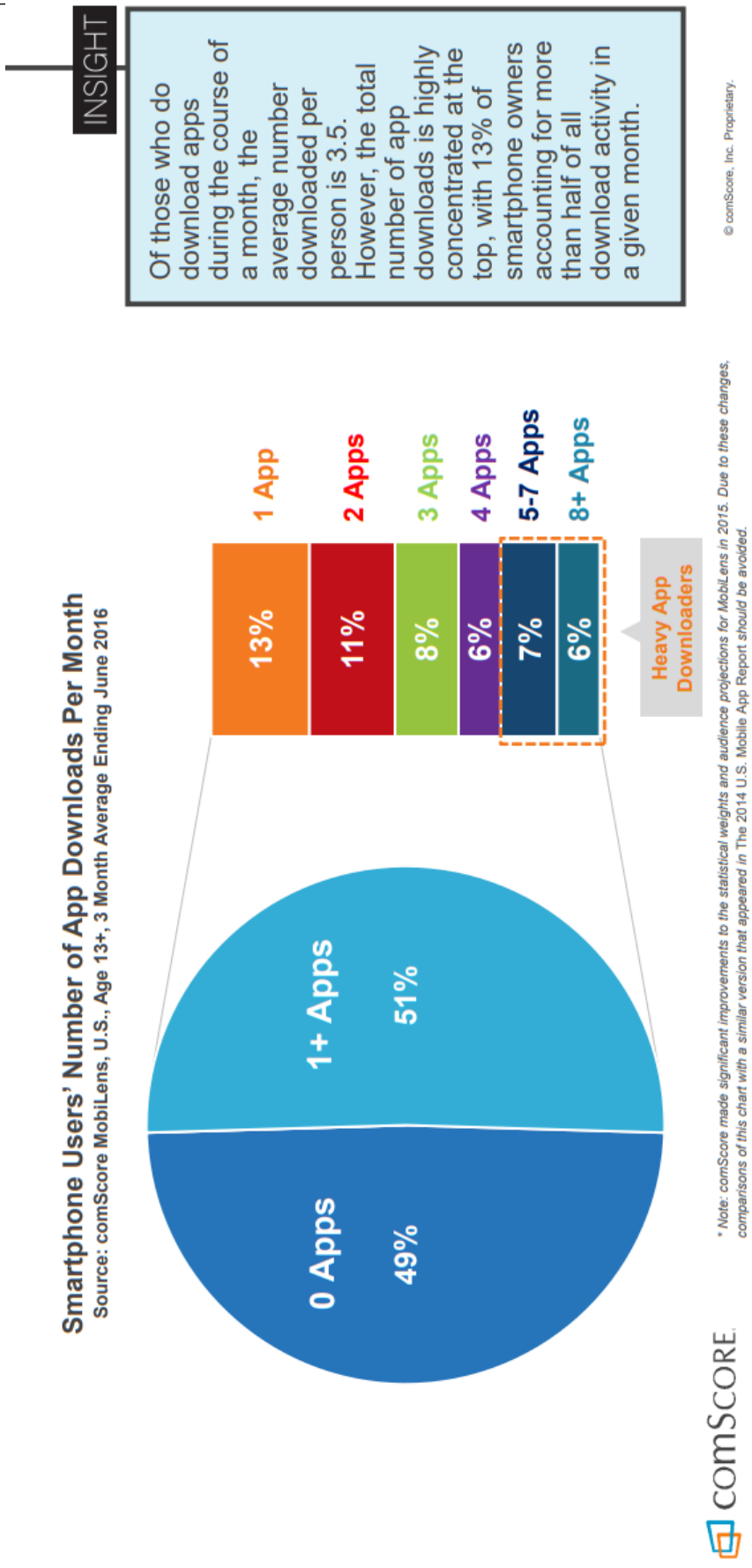
Osmani, A. (g.d.). The App Shell Model. Verkregen van https://developers.google.com/web/fundamentals/architecture/app-shell

Rousse, M. (g.d.). Native App. Verkregen van https://searchsoftwarequality.techtarget.com/definition/native-application-native-app

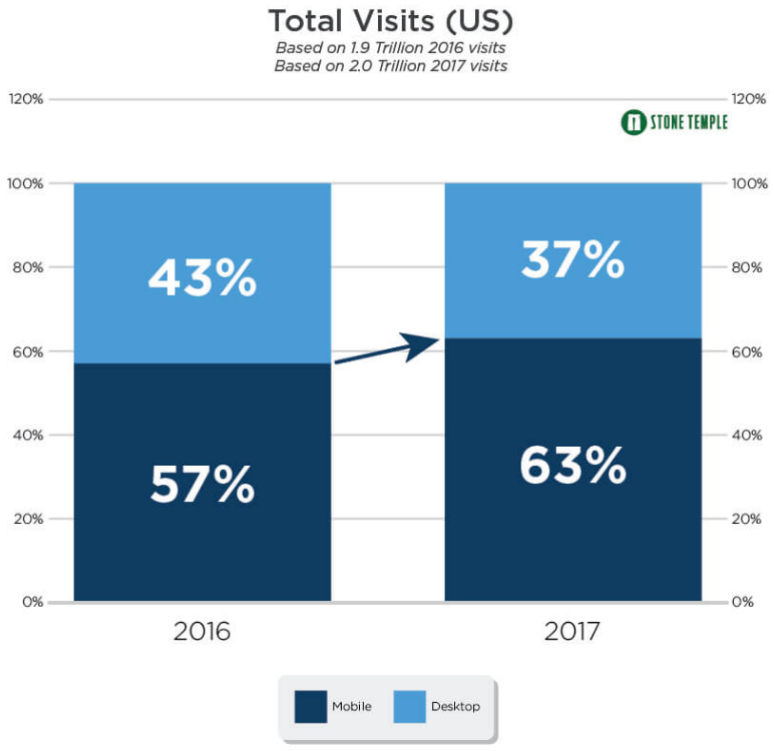
Lijst van figuren



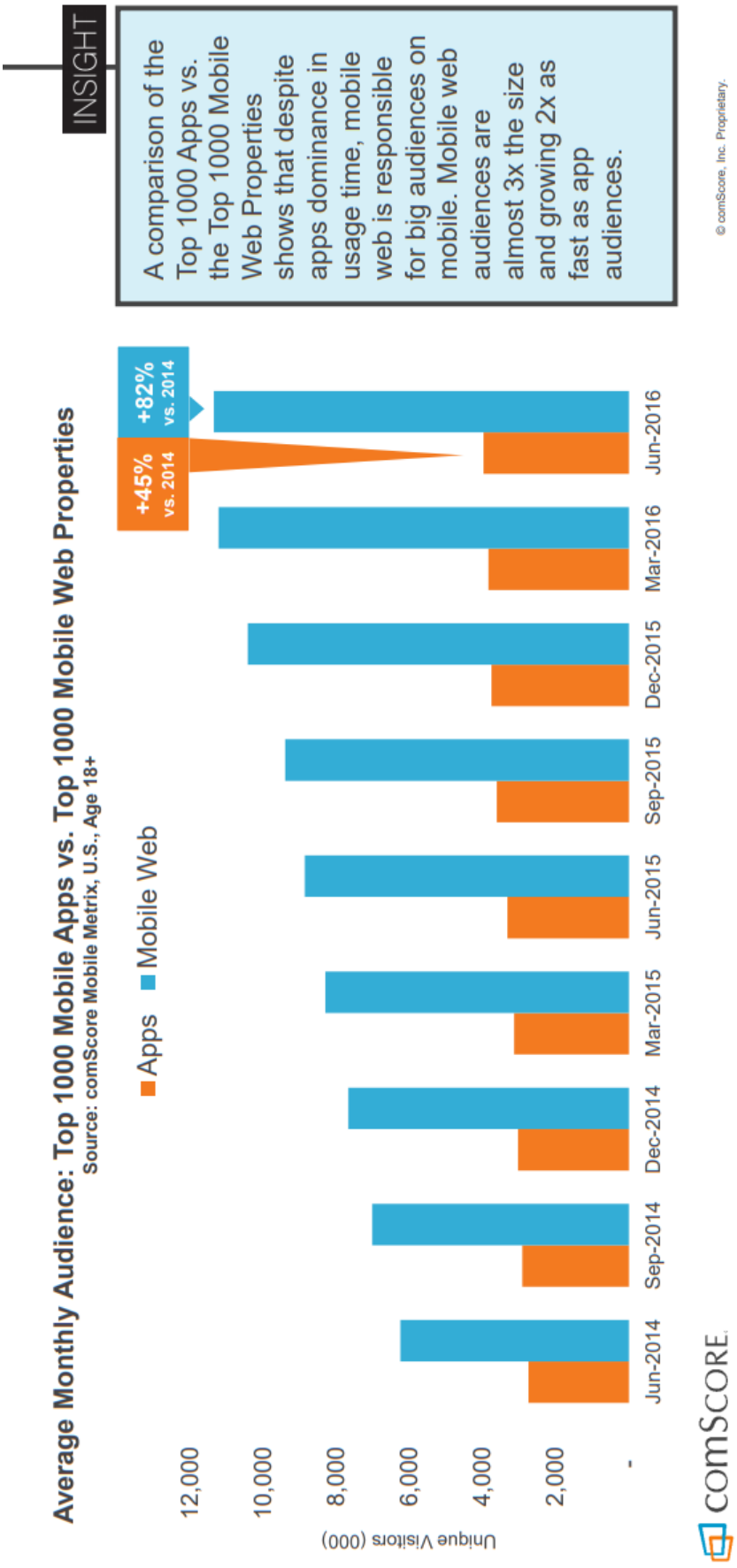
Figuur 3.1: Tijd gespendeerd op mobile: App vs Web



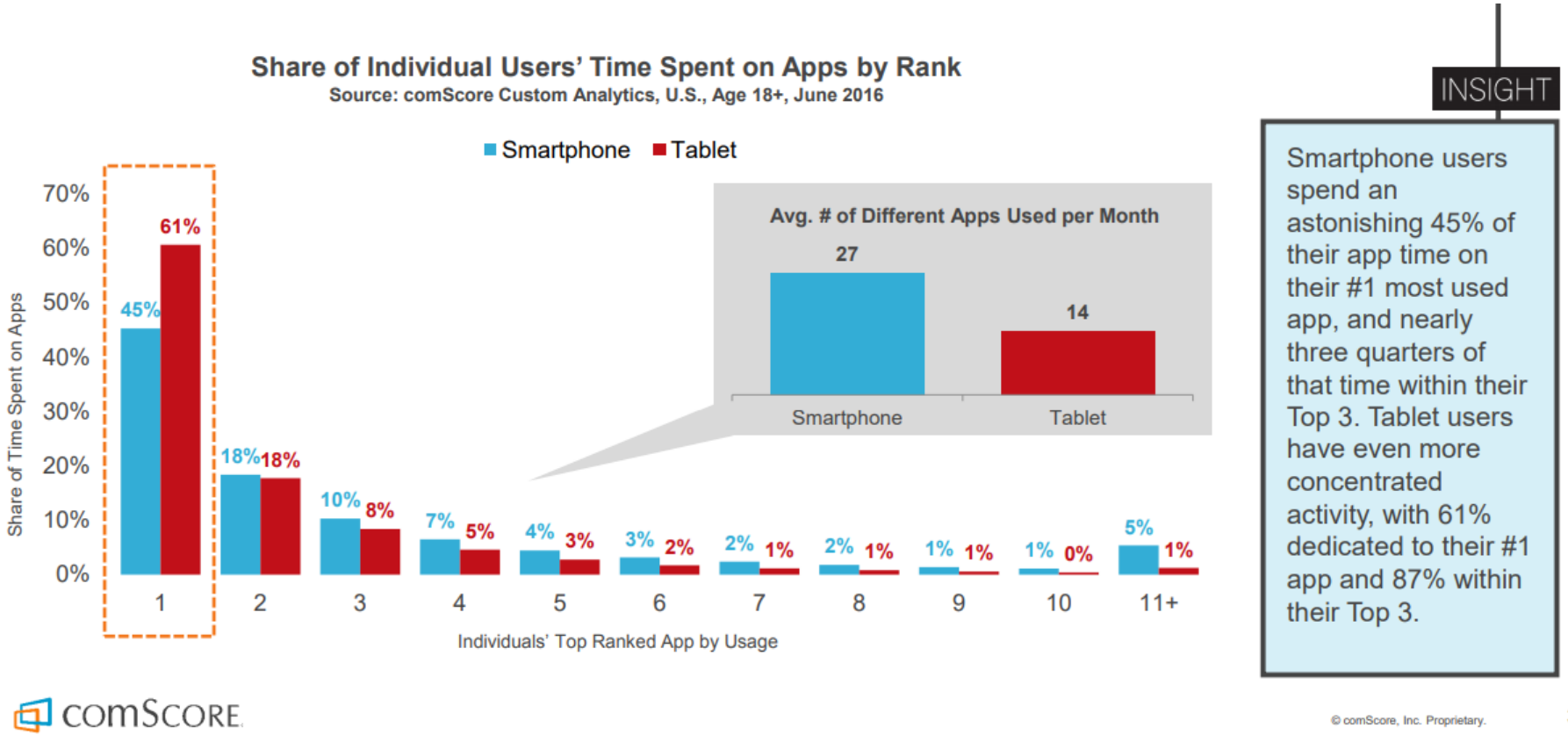
Figuur 3.2: Smartphone gebruikers, gedownloade apps per maand



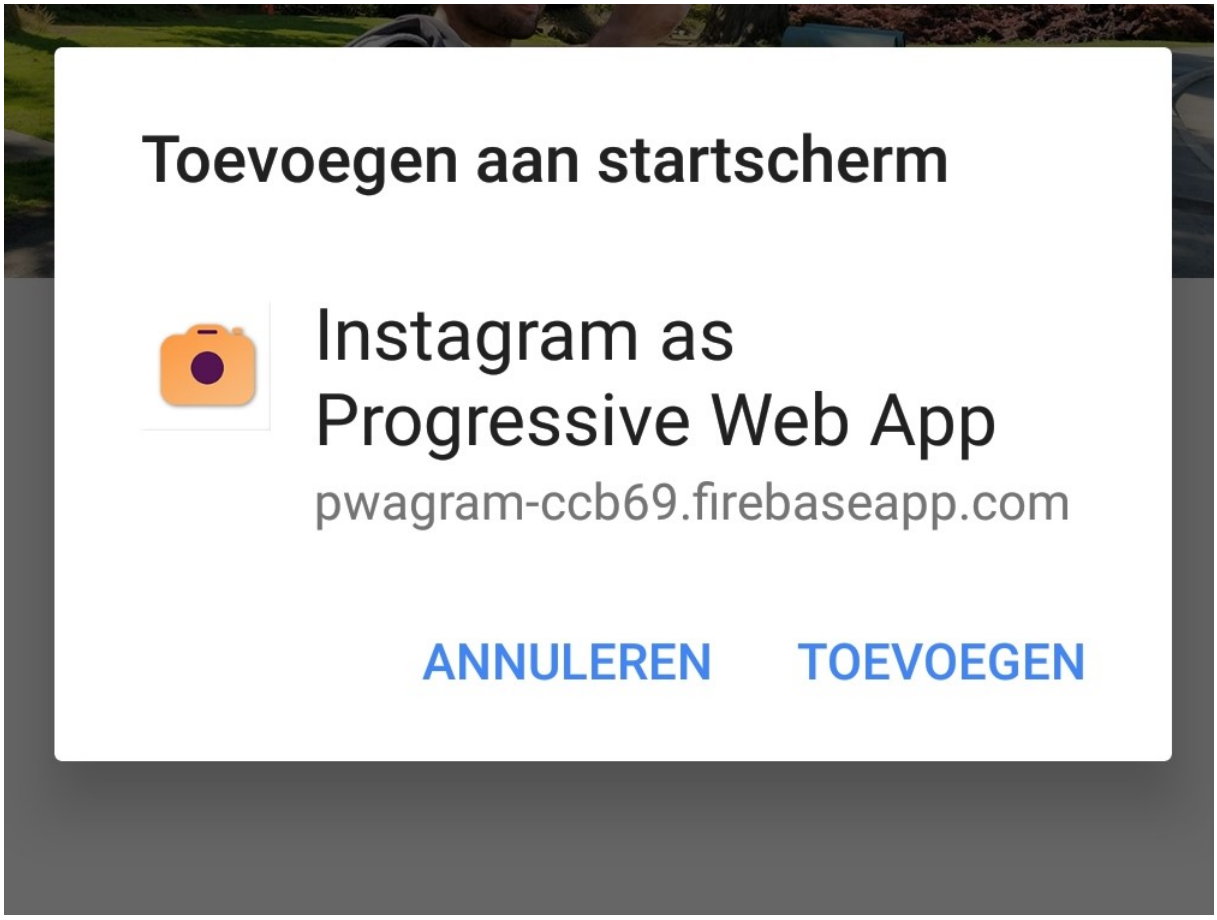
Figuur 3.3: Totaal bezoeken web Mobile vs Desktop



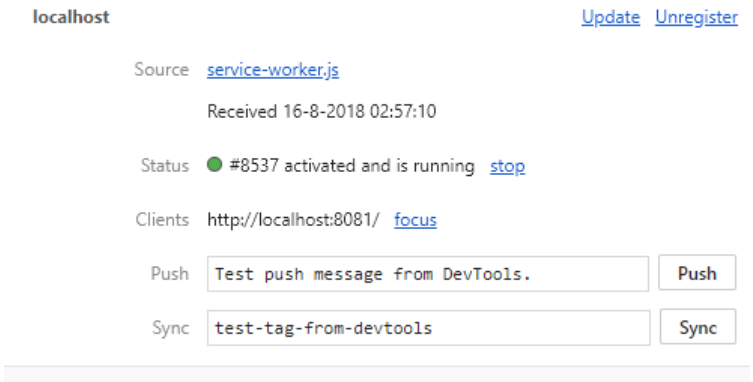
Figuur 3.4: Gemiddeld aantal bezoekers per maand native vs web



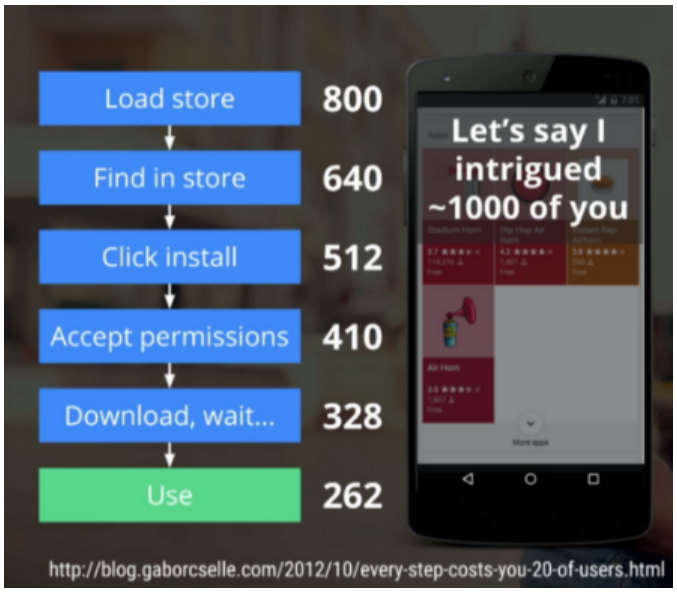
Figuur 3.5: Tijd gespendeerd door gebruikers per app



Figuur 3.6: Installeren app op homescherm



Figuur 3.7: De service worker in developer tools in chrome



Figuur 3.8: Het verlies van gebruikers bij native apps

Lijst van tabellen

	Native app	**Webapp**
Platform	Afhankelijk	Onafhankelijk
Opslag gegevens	Apparaat gebruiker	Doorgaans op webserver
Gebruik functionaliteit apparaat	Alle functionaliteiten	Geen. Sommige webapps kunnen gebruik maken van beperkte
Installatie	Downloaden in app store en installeren	Naar de website gaan, zonder installatie
Updates	Updates moeten worden gedownload en geïnstalleerd	Updates worden meteen gezien door alle gebruikers zonder
Internetverbinding	Meestal niet noodzakelijk	Meestal noodzakelijk

Tabel 3.1: Overzicht native app vs webapp