

**KARELIA-AMMATTIKORKEAKOULU**  
Tieto- ja viestintäteknikan koulutusohjelma

Jusko Wihonen

**PROGRESSIIVISET VERKKOSOVELLUKSET**

Opinnäytetyö  
Toukokuu 2018



**OPINNÄYTETYÖ**  
**Toukokuu 2018**  
**Tieto- ja viestintäteknikan**  
**koulutusohjelma**

Tikkarinne 9  
80200 JOENSUU  
013 260 600

Tekijä  
Jusko Wihonen

Nimeke

Progressiiviset verkkosovellukset

Tiivistelmä

Tämä opinnäytetyö käsittelee progressiivisia verkkosovelluksia, jotka ovat verkkosovelluskehityksen uusi vakiintunut tapa kehittää sovelluksia verkkoon. Ne ovat jo muutamassa vuodessa nousseet verkkosovelluskehittäjien suosioon toimintojensa ansiosta. Ne ovat perinteisten sovelluskaupasta ladattavien sovelluksien kaltaisia, hyvällä käyttöliittymällä varustettuja sovelluksia, joilla pystytään luomaan toimivia ja monipuolisia verkkosovellusratkaisuja.

Opinnäytetyön teoreettisessa viitekehityksessä tutkitaan progressiivisten verkkosovellusten kehitykseen vaadittavia teknologioita, niiden hyötyjä ja ominaisuuksia sekä vertaillaan progressiivisia verkkosovelluksia perinteisiin natiivisovelluksiin. Lisäksi opinnäytetyössä esitellään käytännön esimerkki, kuinka näitä teknologioita voidaan hyödyntää yrityksen palvelujen parantamisessa. Progressiiviset verkkosovellukset hyödyntävät service workereita eli JavaScriptillä kirjoitettuja tiedostoja, application shell -arkkitehtuuria ja web app manifestia. Application shell tarkoittaa verkkosivuston perustoimintaan vaadittavia tiedostoja ja web app manifestest on JSON-pohjainen luettelo verkkosovelluksen tiedoista. Yhdessä nämä kokoavat yhteen natiivisovellusten ominaisuuksilla varustetun verkkosovelluksen.

Progressiiviset verkkosovellukset mahdollistavat verkkosovelluskehityksen kilpailun sovellusmarkkinoista natiivisovelluksien kanssa. Niiden avulla verkkosovelluksiin voidaan tuoda ominaisuuksia, jotka olivat ennen vain perinteisten sovelluskaupasta ladattavien sovellusten käytössä. Ne toimivat nopeasti internetyhteydestä riippumatta, ne voidaan avata suoraan aloitusnäytöltä ja niitä voidaan käyttää kokoruututilassa. Ne ovat kustannustehokas ratkaisu yrityksille, jotka eivät tarvitse sovellukselta pääsyä kaikkiin laitteen ominaisuuksiin, ja yrityksille, jotka eivät ole varanneet sovelluksen kehitykseen suurta budjettia.

Kieli  
suomi

Sivuja 40  
Liitteet 0  
Liitesivumäärä 0

Asiasanat  
progressiivinen verkkosovellus, PWA, verkkosovelluskehitys



**THESIS**  
**May 2018**  
**Degree Programme in Information and  
Communications Technology**

Tikkarinne 9  
80200 JOENSUU  
FINLAND  
+358 13 260 600

Author  
Jusko Wihonen

Title  
Progressive Web Apps

This thesis addresses the new established way of developing applications to the web. Progressive web applications have risen to the popularity of web application developers with their advanced features in only a few years. They function like applications that can be downloaded from the application stores, they have user-friendly interface and developers can create robust and versatile applications for the web with progressive web applications.

The theoretical framework of this thesis examines the technologies required to build progressive web applications, the benefits and features of progressive web applications and comparison of native applications versus progressive web applications. They utilize JavaScript-written files called service workers that together with application shell architecture and web app manifest combine a network application with native application's features.

Progressive web applications allow the web application development to compete over application market with native applications. With them, web applications can utilize features that were only available to traditional applications from the application store. They work fast regardless of the internet connection, they can be opened directly from the user's home screen and they can be used in full-screen mode. Progressive web applications are a cost-efficient solution for companies that do not need access to devices every feature, and for companies who have not allocated a large budget for the development of the application.

Language  
Finnish

Pages 40  
Appendices 0  
Pages of Appendices 0

Keywords  
progressive web apps, PWA, web application development

## Sisältö

Lyhenteet ja termit .....	5
1 Johdanto .....	6
2 Teknologiat .....	7
2.1 App Shell -arkkitehtuuri.....	7
2.2 Service workerit .....	8
2.2.1 Service workerin rekisteröinti.....	9
2.2.2 Service workerin käyttö.....	10
2.3 Web app manifest .....	11
2.3.1 Manifestin käyttö.....	11
2.3.2 Manifestin mahdolliset muut muuttajat.....	13
2.4 Progressiivisen verkkosovelluksen asennus.....	14
3 Progressiivinen verkkosovellus.....	16
3.1 Selainten tuki .....	16
3.2 Ominaisuudet.....	18
3.3 Natiivisovelluksesta verkkosovellukseen .....	20
3.4 Progressiiviset verkkosovellusratkaisut.....	22
3.4.1 Twitter Lite .....	22
3.4.2 Progressiivinen verkkosovellus verkkokaupan tukena.....	23
3.4.3 Käyttäjätavallisempia verkkosovelluksia .....	23
3.5 Progressiivisten verkkosovellusten liiketaloudelliset hyödyt.....	24
4 Käytännön toteutus.....	26
4.1 Verkkokauppa-alusta Shopify .....	27
4.2 Kanta-asiakassivusto .....	27
4.3 Käytetyt teknologiat .....	29
4.4 Kehitysvaihtoehdot .....	32
4.5 Tulokset ja kokemukset .....	33
5 Pohdinta.....	34
Lähteet.....	38

## Lyhenteet ja termit

App Shell	App Shell eli application shell -arkkitehtuuri on tapa rakentaa progressiivisia verkkosovelluksia. Siihen kuuluu verkkosivuston toimimisen kannalta ottaen vähimmäis HTML, CSS ja JavaScript tiedostot.
JSON	Lyhenne tulee sanoista JavaScript Object Notation. JSON on tiedostomuoto, jota käytetään pääosin tiedon siirtoon sovelluksien ja palvelimien välillä.
Natiivisovellus	Perinteisiä kehitysmenetelmiä käyttäen luotu sovellus, jonka voi ladata laitteelle sovelluskaupasta.
PWA	Lyhenne progressiivisille verkkosovelluksille. Lyhenne tulee englanninkielisestä vastikkeesta Progressive web apps.
Service Workerit	JavaScriptillä kirjoitettua koodia, jonka avulla verkkosivustolle saadaan natiivisovelluksen tapaisia ominaisuuksia käyttöön.
Web App Manifest	JSON-tiedosto, jonka avulla progressiivinen verkkosovellus saadaan näyttämään ja toimimaan natiivisovelluksen tavoin.

# 1 Johdanto

Tämän opinnäytetyön aiheena oli verkkosovelluskehityksen uusi kehitystapa, jolla pystytään toteuttamaan natiivisovelluksien tapainen verkossa toimiva sovellus käyttämällä nykyaikaisia verkkoteknologioita. Mobiilialustalle kehitettäessä natiivisovellukset ovat olleet suosituimpia ratkaisuja niiden hyvien käyttökokeusten ja sujuvan toiminnan ansiosta, mutta progressiiviset verkkosovellukset (jäljempänä PWA) alkavat kilpailla perinteisiä kehitysmenetelmiä vastaan (Uotila 2017). Progressiiviset verkkosovellukset tulivat ensimmäisen kerran esille vuonna 2015, kun Googlen Chrome-selaimen parissa työskentelevä insinööri Alex Russell kertoi niistä blogikirjoituksessaan. Hän esitteli joukon uusia verkkotekniikoita ja niiden käyttötarkoituksia ja nimesi ne Progressive Web Appseiksi. (Russell 2015.)

PWA:t eivät ole tarkastelun arvoisia vain sen takia, että ne ovat uusin trendi verkkosovelluskehityksessä, vaan niissä on myös paljon potentiaalia. Ne haastavat perinteiset natiivisovellusmarkkinat samankaltaisilla ominaisuuksilla ja käyttökokeuksella sekä tarjoavat lisäksi parannuskeinoja ongelmiin, joita natiivisovelluksilla on ollut. Responsiivisen asettelunsa ansiosta PWA:t mukautuvat päätelaitteen näyttöön sen koosta riippumatta sekä niitä pystytään selaamaan ja jakamaan kuin normaalia internetsivua. Ne sitovat käyttäjää sovelluksen käyttämiseen push-viestien avulla ja niissä on samankaltainen käyttöliittymä kuin sovelluskaupasta ladattavissa sovelluksissa. Lisäksi niiden kehittäminen ja julkaisu on helpompaa, mikä vähentää sovelluskehityksen kustannuksia. (Gazdecki 2018a.)

Opinnäytetyön tavoitteena oli tutkia, mitä progressiiviset verkkosovellukset ovat, miten niitä kehitetään ja mitkä ovat niiden edut perinteiseen sovelluskehitykseen verrattuna. Tämän lisäksi työssä tutkittiin, miten yritykset voivat hyödyntää progressiivisiä verkkosovelluksia palveluidensa parantamisessa. Lopuksi työssä esitellään käytännön toteutus, kuinka progressiivisten verkkosovellusten ominaisuuksia voidaan hyödyntää yrityksen kanta-asiakastoiminnan parantamisessa.

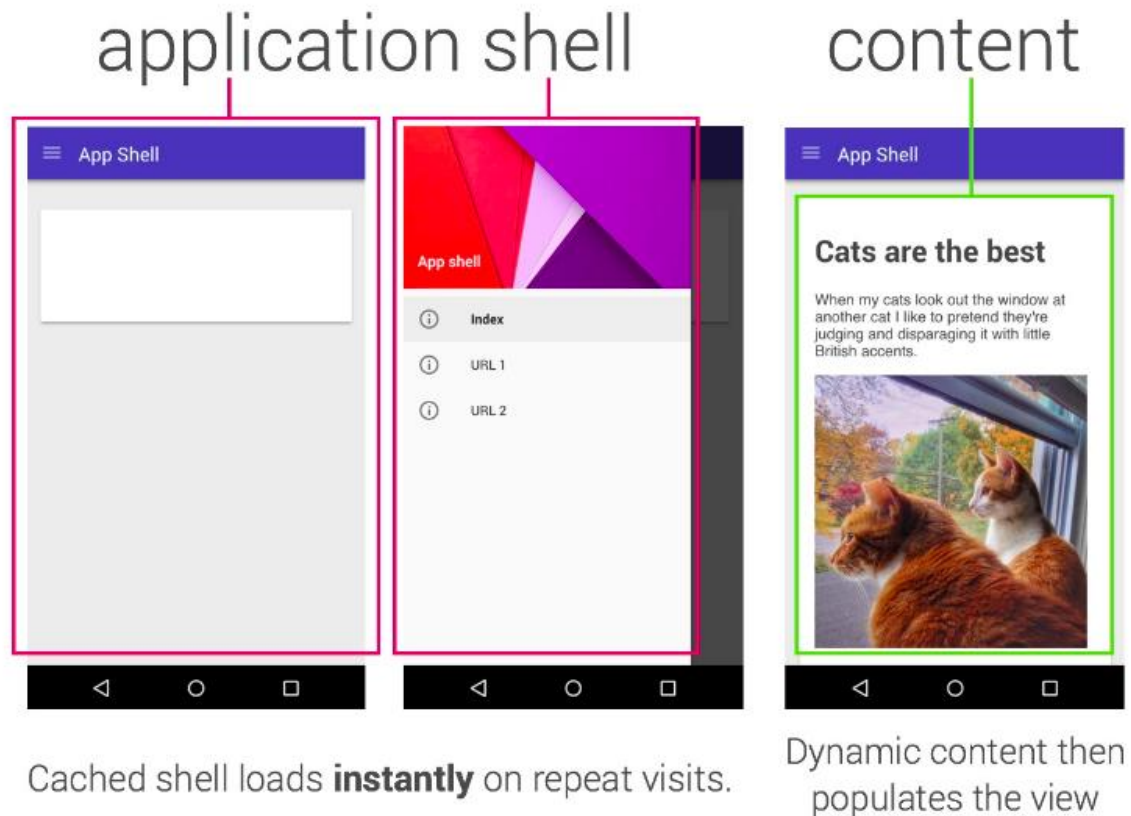
## 2 Teknologiat

Progressiiviset verkkosovellukset kokoavat yhteen natiivisovelluksen ja verkkosivun parhaat puolet. Yksinkertaisimmillaan ne ovat verkkosivuja, jossa hyödynnetään kehittyneitä verkkoteknologioita ja toimintoja, kuten service workereita. (Tivia 2017.) Progressiivisille verkkosovelluksille ominaista on niiden nopeus, luotettavuus sekä kyky sitouttaa käyttäjiä käyttämään palvelua. Service workereiden avulla sovelluksen tiedot tallennetaan laitteen välimuistiin, jolloin latausajat nopeutuvat ja samalla sovellusta pystytään käyttämään ilman internetyhteyttä. Käyttäjät voivat asentaa progressiivisen verkkosovelluksen aloitusnäyttöön suoraan verkkosivulta vierailematta sovelluskaupassa. (Google LLC 2018a.)

Googlen kehittäjille suunnatussa Chrome Dev Summit 2015 -tapahtumassa ensimmäistä kertaa esitellyt progressiiviset verkkosovellukset ovat jo muutamassa vuodessa nousseet kehittäjien suosimiksi ja käytettäviksi ratkaisuksi. Sovellusten progressiivisten ominaisuuksien ansiosta niitä on helppo lisätä verkkosivuille tarpeen mukaan. Jos sivustolla on valmiiksi sovelluksen tapainen käyttöliittymä, progressiivisten verkkosovellusten ominaisuuksien lisääminen sivustolle tekee siitä vielä paremman. (Farrugia 2016.)

### 2.1 App Shell -arkkitehtuuri

Application shell eli app shell -arkkitehtuuri on tapa rakentaa progressiivisia verkkosovelluksia, ja sen rakenteen ansiosta verkkosovellukset latautuvat yhtä nopeasti ja sulavasti kuin tavalliset natiivisovellukset. App shell kokoaa yhteen verkkosivun vähimmäistiedostot (HTML, CSS ja JavaScript). Nämä tiedostot tallennetaan laitteen välimuistiin, josta ne ladataan seuraavalla kerralla, kun käyttäjä avaa sivuston uudestaan. Sovelluksen toisella avauskerralla näitä tiedostoja ei tarvitse ladata internetin avulla, vaan ne ladataan laitteen välimuistista. Tämä nopeuttaa sivuston latausaikaa huomattavasti sekä mahdollistaa sovelluksen käytön ilman internetyhteyttä offline-tilassa. (Osmani 2018.) Kuvassa 1 havainnollistetaan application shell -arkkitehtuuria ja kuinka se toimii käytännössä.



Kuva 1. Kuvasarja, joka havainnollistaa application shell -arkkitehtuurin käytön (Osmani 2018).

## 2.2 Service workerit

Natiivisovellusten perinteiset ominaisuudet, kuten sovelluksen käyttäminen ilman internetyhteyttä, päivitys taustalla ja push-ilmoitukset, ovat nykyisin käytettävissä verkkosovelluksissa. Service workereiden ansiosta esimerkiksi kyseisiä ominaisuuksia voidaan käyttää verkkosovellusten kautta. Ne ovat JavaScriptillä kirjoitettuja skriptejä, jotka ajetaan sivun latauksen yhteydessä. (Gaunt 2018.)

Kaikki selaimet eivät kuitenkaan vielä tue service workereiden käyttöä. Ne toimivat täysin Chromella, Operalla, Samsungin omalla internet-selaimella ja Firefoxilla. Myös mobiilikäytössä suosituilla Applen Safari-selaimella service workereiden tuen kehitys on alkanut, mutta ne eivät vielä toimi täydellisesti. Service workereiden käyttöä ja toimivuutta eri selaimilla käsitellään tarkemmin kappaleessa 3.1.



Service workerit vaativat toimiakseen salatun HTTPS-yhteyden, joten ne pakottavat sivuston ottamaan käyttöön HTTPS-suojauksen. Salattu yhteys parantaa sivuston tietoturvaa ja mahdollistaa tiedon suojatun siirron verkossa. Ilman salatua yhteyttä kolmas osapuoli voisi käyttää service workereita sivustoa vastaan eikä sen hyödyksi. (Gaunt 2018.)

### 2.2.1 Service workerin rekisteröinti

Service worker täytyy rekisteröidä sivulla käyttäjälle, jotta sen ominaisuuksia voidaan hyödyntää. JavaScript -tiedostossa kerrotaan, mistä kansioista service worker -tiedostoa tulee etsiä. Koodi tarkastaa, onko service worker saatavilla sille määritellystä hakemistosta ja rekisteröi sen. Jos service worker on rekisteröity aiemmin, rekisteröintiä ei tehdä toista kertaa. (Gaunt 2018.) Kuvasta 2 löytyvällä esimerkkikoodilla service worker voidaan rekisteröidä sivulle.

```
if ('serviceWorker' in navigator) {  
  window.addEventListener('load', function() {  
    navigator.serviceWorker.register('/sw.js').then(function(registration) {  
      // Rekisteröinti onnistui  
      console.log('ServiceWorker registration successful with scope: ', registration.scope);  
    }, function(err) {  
      // Rekisteröinti epäonnistui!  
      console.log('ServiceWorker registration failed: ', err);  
    });  
  });  
};
```

Kuva 2. Kooditiedosto, joka rekisteröi service workerin selaimen.

## 2.2.2 Service workerin käyttö

Asennuksen jälkeen service worker tallentaa laitteen välimuistiin verkkosovelluksen staattiset resurssit eli application shellin, jotta sovellus voi käynnistyä seuraavalla kerralla nopeasti ja ilman internetyhteyttä. Jos yksikin määritellyistä tiedostoista ei lataudu laitteen välimuistiin, latausvaihe epäonnistuu ja service worker yrittää ladata tiedostoja seuraavalla vierailukerralla. On tärkeää, että ladattavien tiedostojen määrä pysyy kohtuullisena, koska suurempi tiedostojen määrä lisää latauksen epäonnistumisen riskiä. (Gaunt 2018.) Alla löytyvällä koodilla (kuva 3) voidaan testata sivuston tiedostojen tallentamista laitteen välimuistiin.

```
var CACHE_NAME = 'my-site-cache-v1';
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);
      })
  );
});
```

Kuva 3. Koodi, jolla sivuston tietoja voidaan tallentaa laitteen välimuistiin.

## 2.3 Web app manifest

Verkkosovellusluettelo eli web app manifest on yksinkertainen JSON-tiedosto, joka sisältää tiedot, joita sovellus tarvitsee toimiakseen progressiivisen verkkosovelluksen tavoin. Se määrittelee, miltä verkkosivu näyttää, kun käyttäjä lisää sen kuvakkeena aloitusnäyttöön, mitä sovellusikonin painalluksesta tapahtuu sekä minkälainen on käynnistyksen aikainen ulkoasu. (Gaunt & Kinlan 2018.) Web app manifestin avulla sovellus voidaan avata aloitusnäytöstä natiivisovelluksen tapaan ja silloin sen ydintiedostot ladataan laitteen välimuistista ja sitä pysytään käyttämään kokoruututilassa.

### 2.3.1 Manifestin käyttö

Web app manifest voidaan nimetä millä nimellä tahansa, mutta tiedostomuodon on oltava JSON-tiedosto. Yleinen käytäntö on kuitenkin nimetä se nimellä manifest.json tunnistettavuuden kannalta. Manifest lisätään verkkosivun HTML-tiedoston <head> osioon kuvan 4 osoittamalla tavalla.

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

Kuva 4. Sivuston HTML <head> osioon lisättävä koodirivi, mikä määrittelee manifest-tiedoston sijainnin

Manifest.json-tiedoston voi luoda millä tahansa tekstieditorilla, jossa voi valita tallennusmuodoksi JSON-tiedostotyyppin. Seuraavalta sivulta löytyvästä esimerkki manifest-tiedostosta (kuva 5) löytyvät kaikki tarvittavat perustiedot progressiivisen verkkosovelluksen testaukseen.

```
{
  "name": "Testi Applikaatio v1",
  "short_name": "TestiAppi",
  "icons": [
    {
      "src": "logo-48.png",
      "type": "image/png",
      "sizes": "48x48"
    },
    {
      "src": "logo-96.png",
      "type": "image/png",
      "sizes": "96x96"
    },
    {
      "src": "logo-192.png",
      "type": "image/png",
      "sizes": "192x192"
    }
  ],
  "start_url": "esimerkkisovellus.fi",
  "display": "standalone",
  "theme_color": "#FFFFFF",
  "background_color": "#000000"
}
```

Kuva 5. Esimerkki web app manifest -tiedostosta

Manifest-tiedostoon on hyvä lisätä sovellukselle tärkeät ominaisuudet kuten aloitusnäyttöön lisättävät ikonit, sovelluksen nimi sekä ikonin painalluksesta avattava aloitussivu. Seuraavaksi löytyvässä listassa on kerrottu, mitä kuvaan 5 määritellyt ominaisuudet tekevät sovellukselle.

- **Short\_name** määrittelee sovelluksen lyhennetyin nimen paikkoihin, minne sovelluksen koko nimi ei mahdu. Yleensä lyhennettyä nimeä käytetään aloitusnäytössä applikaation alla tilanpuutteen takia.
- **Name** pitää sisällään applikaation koko nimen. Se näytetään esimerkiksi silloin, kun sovellus käynnistetään aloitusnäytöstä.
- **Icons** määrittelee kuvien taulukon, josta löytyy erikokoiset sovelluskuvat eri päätelaitteille.
- **Start\_url** määrittelee sovelluksen aloitussivun. Kun käyttäjä käynnistää sovelluksen aloitusnäytön kautta, tämä sivu aukaistaan.
- **Display** kertoo sovellukselle, missä tilassa se aukaistaan. Standalone-tilassa sovellus aukaistaan kokoruututilassa ilman selaimen hakupalkkia.
- **Theme color** määrittelee sovelluksen päävärin, jota käytetään selaimen taustavärinä mobiililaitteella selattaessa.
- **Background color** määrittelee sovelluksen avausikkunan taustavärin. (Mozilla Foundation 2018a.)

### 2.3.2 Manifestin mahdolliset muut muuttujat

Web app manifestiin voidaan määritellä myös muita ominaisuuksia. Muut mahdolliset ominaisuudet on aseteltu taulukkoon 1, ja jokaisesta on annettu myös esimerkki, kuinka se tulee manifest-tiedostoon lisätä. (Mozilla Foundation 2018a.)

Taulukko 1. Web app manifest -tiedostoon muut mahdolliset lisättävät muuttujat (Mozilla Foundation 2018a.)

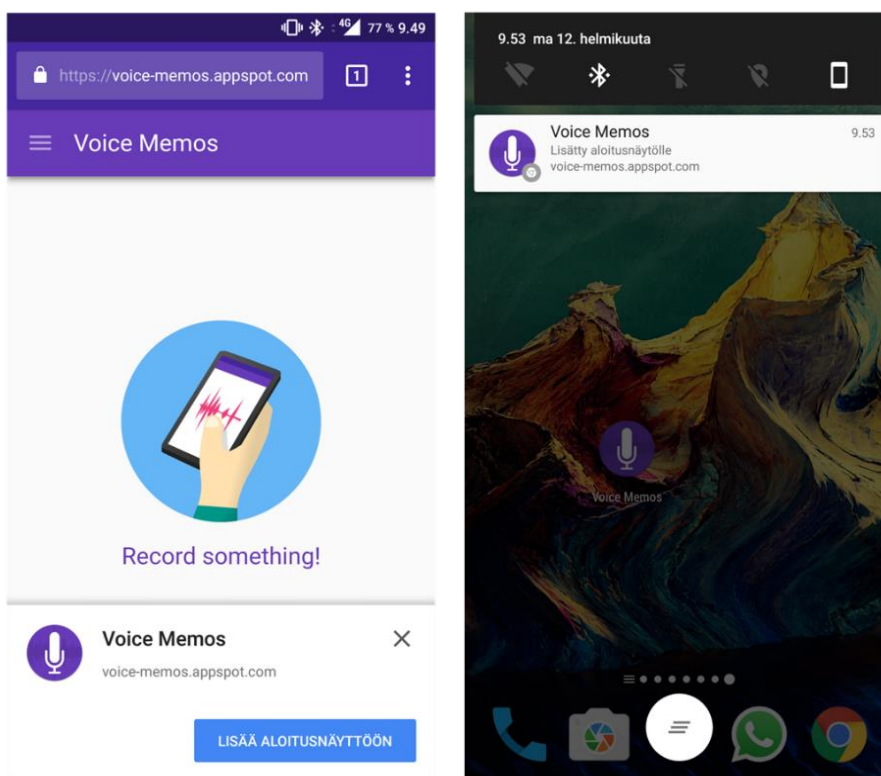
Muuttuja	Selitys	Esimerkki
Lang	Muuttujaan voidaan määritellä sovelluksessa käytettävä kieli. Kieli määritellään käyttäen kielen lyhennettyä tagia.	"lang": "en-US"
Orientation	Määrittelee missä suunnassa sovellus aukaistaan.	"orientation": "portrait"
Prefer_related_applications	Määrittelee totuusarvomuttujan, joka kertoo käyttäjälle, onko verkkosovelluksen lisäksi suositeltavaa ladata natiivisovellus.	"prefer_related_applications": false
Related_applications	Muuttuja sisältää olioista koostuvan taulukon, josta löytyy tieto natiivisovelluksesta.	"related_applications": [{"platform": "play", "url": "https://play.google.com/", "id": "com.esimerkki.1"}]
Scope	Määrittelee, minkä verkkoosoitteen sisällä sovellus pyörii. Jos käyttäjä menee pois määritellyltä alueelta, aukeaa normaali verkkosivu.	"scope": "/sovellus/"
Theme_color	Määrittelee sovelluksen päävärin.	"theme_color": "black"
Background_color	Määrittelee taustavärin sovelluksen latausnäkykseen.	"background_color": "white"

## 2.4 Progressiivisen verkkosovelluksen asennus

Progressiivisen verkkosovelluksen voi asentaa aloitusnäyttöön vieraillemalla internetsivulla, jossa progressiiviset ominaisuudet on otettu käyttöön. Erillistä listaa progressiivisista verkkosovelluksista ei ole, mutta hakemalla voi löytää sivuja,

jotka kokoavat yhteen toimivimpia sovelluksia mitä verkosta löytyy. Esimerkiksi PWA Rocks sivustolta löytyy lista suosituista progressiivisista verkkosovelluksista (PWA Rocks 2018).

Kun käyttäjä tulee sivustolle, joka käyttää progressiivisten verkkosovellusten ominaisuuksia, sivusto ehdottaa sovelluksen lisäämistä aloitusnäyttöön pop-up-ikkunassa. Sovellus voidaan asentaa aloitusnäyttöön yhden napin painalluksella ilman tarvetta vieraillla laitteella löytyvässä sovelluskaupassa. Lisäämisen jälkeen sovellus voidaan käynnistää laitteen aloitusnäytöstä ja hyödyntää progressiivisten verkkosovellusten tuomia ominaisuuksia. Alla olevassa kuvasarjassa (kuva 6) on havainnollistettu, kuinka sovellus lisätään aloitusnäyttöön internetsivulta Android laitteella käyttäen Googlen Chrome-selainta.



Kuva 6. Kuvakaappaus progressiivisen verkkosovelluksen ilmoituksesta ja kotinäyttönäkymästä sovelluksen asennuksen jälkeen.

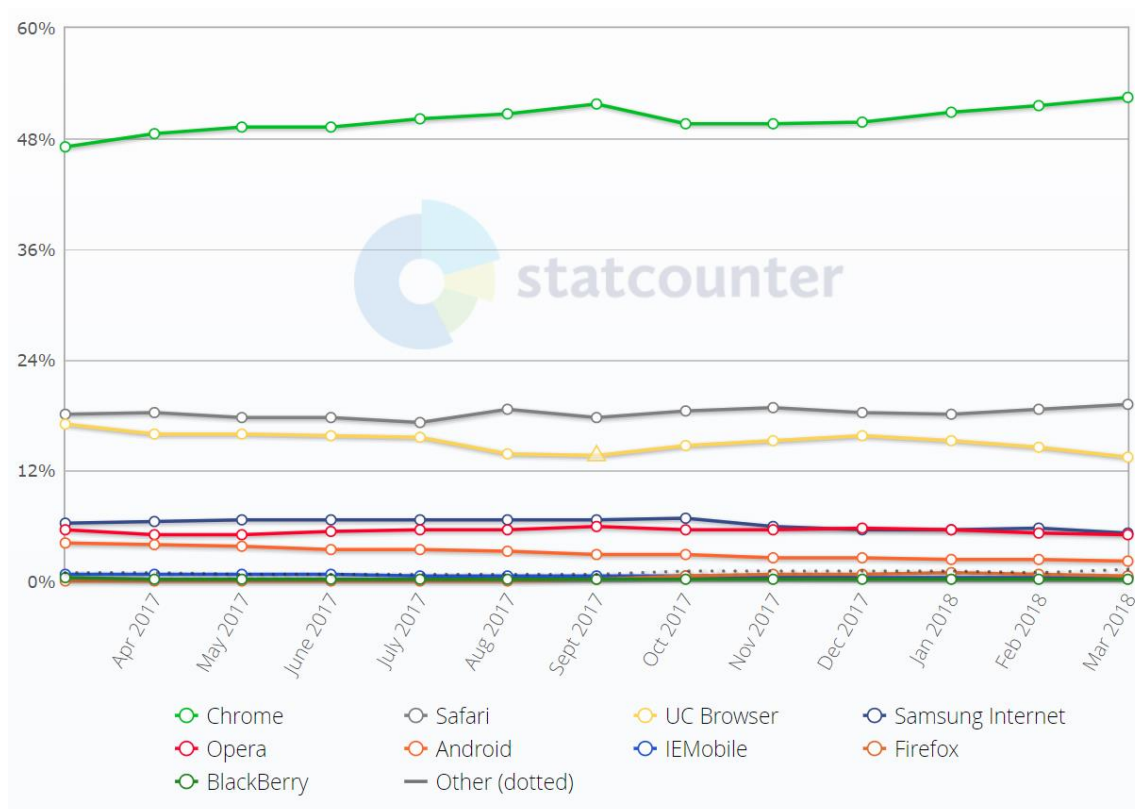
### 3 Progressiivinen verkkosovellus

Progressiiviset verkkosovellukset nousivat ensimmäisen kerran esille Googlen Dev Summit -tapahtumassa vuonna 2015. Tapahtuman pääpuheenvuorossa käytiin läpi mobiilikehittämisen historiaa ja puhuttiin progressiivisista verkkosovelluksista, jotka ovat nopeita, luotettavia ja takaavat natiivisovelluksen tapaisen käyttökokemuksen käyttäen moderneja ja nykyaikaisia verkkoteknologioita. Käytännössä progressiiviset verkkosovellukset voidaan erottaa perinteisistä verkkosivuista kolmella eri vertailukohteella. Ne toimivat tehokkaammin ja luotettavammin sekä niillä on kyky sitouttaa käyttäjiä vierailemaan sivuilla enemmän käyttäen nykyaikaisia verkkoteknologioita, kuten push-viestejä. (Fisher 2015.)

#### 3.1 Selainten tuki

Käyttäkseen progressiivisiä verkkosovelluksia selaimen tulee tukea niille tärkeimpiä ominaisuuksia, tässä tapauksessa service workereita ja web app manifestia. Luonnollisesti Googlen Chrome selain tukee molempia ja uusimmat päivitykset tulevat ensimmäisenä juuri Chromelle. Irlannissa sijaitsevan verkkoliikenneanalytiikkaan erikoistuneen yhtiön StatCounterin (2018) mukaan kaikista mobiilikäyttäjistä 51,66 % käyttää Chromea (kuvio 1).





Kuvio 1. Selainten osuudesta mobiilikäyttöjärjestelmillä (StatCounter 2018).

Toiseksi eniten käyttäjiä on Applen Safarilla, jota käyttää 18,55 % ihmisistä (StatCounter 2018). Apple vastusti progressiivisia verkkosovelluksia pitkään, kunnes päätti itsekin lähteä kehittämään tukea niille. Suurin syy Applen myöhästyneelle heräämiselle progressiivisten verkkosovellusten tuen kehittämiseksi löytyy sen suurista natiivisovellusmarkkinoista. Apple ei ole nähnyt tarpeeksi lisäarvoa progressiivisten verkkosovellusten tukemisessa. (Asay 2016.) Safarin tuki service workereille ja web app manifestille julkaistiin kuitenkin alkuvuodesta 2018 (Fablet 2018). Ennen tätä päivitystä verkkosovelluksissa pystyttiin jo hyödyntämään progressiivista kehittämistä, mutta ei täysin samalla tavalla kuin uusien ominaisuuksien kanssa. Sivustoille pystyttiin lisäämään Applen metatageja, jotka Safari tunnisti ja sovelluksesta pystyttiin tekemään progressiivisen verkkosovelluksen kaltainen ratkaisu. (Garg 2017.)

Safarin uusimman päivityksen mukana tuli tuki progressiivisten verkkosovellusten ominaisuuksille service workereille sekä web app manifestille. Nämä ominai-

suudet eivät kuitenkaan toimi Safarilla täysin samalla tavalla kuin muilla selaimilla. Muilla selaimilla service worker ladataan verkkosivustolta kerran ja se jää elämään laitteen välimuistiin myöhempää käyttöä varten ja täten sovellusta pystytään käyttämään myös ilman internetyhteyttä. Safarilla service worker pysyy laitteen välimuistissa vain kaksi viikkoa, jonka jälkeen se poistetaan. Progressiivisten verkkosovellusten kannalta tämä käytäntö ei ole paras mahdollinen, jos käyttäjä on ladannut tietyn sovelluksen myöhempää käyttöä varten. Kun käyttäjä yrittää avata sovellusta kolme viikkoa latauksen jälkeen ilman internetyhteyttä, sovellus ei toimi oletetusti vaan tarvitsee internetyhteyden sovelluksen lataamista varten. Apple on myös tehnyt tuen web app manifestille hieman poiketen muista selaimista. Safari ei tue kaikkia sen ominaisuuksia ja jotkut ominaisuudet toimivat eri tavalla Applen laitteilla kuin Androidilla. Web app manifestin ominaisuuksia Safarille kuitenkin kehitetään eteenpäin ja ominaisuuksia lisätään ja parannellaan seuraavissa päivityksissä. (Firtman 2018.)

Viisi seuraavaksi suosituinta selainta ovat kiinalainen UC Browser (käytössä 14,48 %:lla käyttäjistä), Samsung Internet (5,65 %), Opera (5,15 %), Androidin oma internetselain (2,31 %) sekä Firefox (0,76 %) (StatCounter 2018). Kaikki edellä mainitut selaimet tukevat service workereita ja web app manifestia niiden alkuperäisellä suunnittelutavalla (Caniuse 2018a, Caniuse 2018b).

### **3.2 Ominaisuudet**

Vaikka progressiiviset verkkosovellukset ovat olleet julkisuudessa jo muutaman vuoden, niiden monipuoliset käyttötarkoitukset ovat monille vielä vieraita. Niiden tekniikoilla pystytään lisäämään suuri osa ominaisuuksista verkkosovelluksiin, mitkä olivat ennen käytössä vain natiivisovelluksissa. Näitä ominaisuuksia ovat esimerkiksi push-viestit, toiminta ilman verkkoyhteyttä ja sovelluksen käynnistäminen aloitusnäytöstä. (Dascalescu 2018.) Alla löytyvästä kuvasta näkyy 7, mitkä kaikki ominaisuudet ovat saatavilla verkkosovelluksiin kirjoitushetkellä (7.3.2018). Kuvakaappaus on otettu Android-laitteella käyttäen Chrome-selainta (What Web Can Do Today 2018).

Camera & Microphone	Input	Surroundings
<ul style="list-style-type: none"> <li>AUDIO &amp; VIDEO CAPTURE ✓</li> <li>ADVANCED CAMERA CONTROLS ✓</li> <li>RECORDING MEDIA ✓</li> <li>REAL-TIME COMMUNICATION ✓</li> </ul>	<ul style="list-style-type: none"> <li>TOUCH GESTURES ✓</li> <li>SPEECH RECOGNITION ✓</li> <li>CLIPBOARD (COPY &amp; PASTE) ✓</li> <li>POINTING DEVICE ADAPTATION ✓</li> </ul>	<ul style="list-style-type: none"> <li>BLUETOOTH ✓</li> <li>USB ✗</li> <li>NFC ✗</li> <li>AMBIENT LIGHT ✗</li> </ul>
Device Features	Location & Position	Seamless Experience
<ul style="list-style-type: none"> <li>NETWORK TYPE &amp; SPEED ✓</li> <li>ONLINE STATE ✓</li> <li>VIBRATION ✓</li> <li>BATTERY STATUS ✓</li> <li>DEVICE MEMORY ✓</li> </ul>	<ul style="list-style-type: none"> <li>GEOLOCATION ✓</li> <li>GEOFENCING ✗</li> <li>DEVICE POSITION ✓</li> <li>DEVICE MOTION ✓</li> <li>PROXIMITY SENSORS ✗</li> </ul>	<ul style="list-style-type: none"> <li>OFFLINE MODE ✓</li> <li>BACKGROUND SYNC ✓</li> <li>INTER-APP COMMUNICATION ✓</li> <li>PAYMENTS ✓</li> <li>CREDENTIALS ✓</li> </ul>
Screen & Output	Operating System	Native Behaviors
<ul style="list-style-type: none"> <li>VIRTUAL &amp; AUGMENTED REALITY ✓</li> <li>FULLSCREEN ✓</li> <li>SCREEN ORIENTATION &amp; LOCK ✓</li> <li>WAKE LOCK ✗</li> <li>PRESENTATION FEATURES ✓</li> </ul>	<ul style="list-style-type: none"> <li>OFFLINE STORAGE ✓</li> <li>FILE ACCESS ✓</li> <li>CONTACTS ✗</li> <li>SMS ✗</li> <li>STORAGE QUOTAS ✓</li> <li>TASK SCHEDULING ✗</li> </ul>	<ul style="list-style-type: none"> <li>LOCAL NOTIFICATIONS ✓</li> <li>PUSH MESSAGES ✓</li> <li>HOME SCREEN INSTALLATION ✓</li> <li>FOREGROUND DETECTION ✓</li> <li>PERMISSIONS ✓</li> </ul>

Kuva 7. Verkkosovelluksiin saatavilla olevat ominaisuudet Android-laitteella (What Web Can Do Today 2018).

PWA:t voivat käyttää laitteen ominaisuuksia ja toimintoja kuten kameraa, sisäistä muistia, paikan- ja liikkeentunnistimia, kasvontunnistusta sekä paljon muuta. Näiden ominaisuuksien ansiosta myös nykyajan modernit lisätyn todellisuuden AR- ja VR-tekniikat ovat käytettävissä verkkosovelluksissa. (Mozilla Foundation 2018b.) Verkkosovellukset eivät voi kuitenkaan vielä käyttää kaikkia ominaisuuksia, mitkä ovat natiivisovelluskehityksen puolella toteutettavissa. Näitä ominaisuuksia ovat muun muassa pääsy puhelimen yhteystietoihin ja kalenteriin, hälytysten käyttö, puhelinpalveluiden käyttö sekä puhelimen laitteiston kuten kameran käyttö.

Useimmissa käyttötapauksissa progressiivisen verkkosovelluksen valitseminen natiivisovelluksen sijaan olisi parempi investointi. Verkkosovellukset tarjoavat tarvittavat ominaisuudet sovellukselle, eikä sovellusta tarvitse erikseen kehittää eri

käyttöjärjestelmille. Progressiivisilla verkkosovelluksilla on lisäksi tiettyjä ominaisuuksia, mitkä eivät ole saatavilla natiivisovelluskehityksen puolella. Koska verkkosovellusten käytettävissä olevien ominaisuuksien määrä on kasvanut suureksi, olisi osasta suosituista natiivisovelluksista mahdollista toteuttaa toimivat ratkaisut verkkoon progressiivisiksi verkkosovelluksiksi. (Dascalescu 2018.)

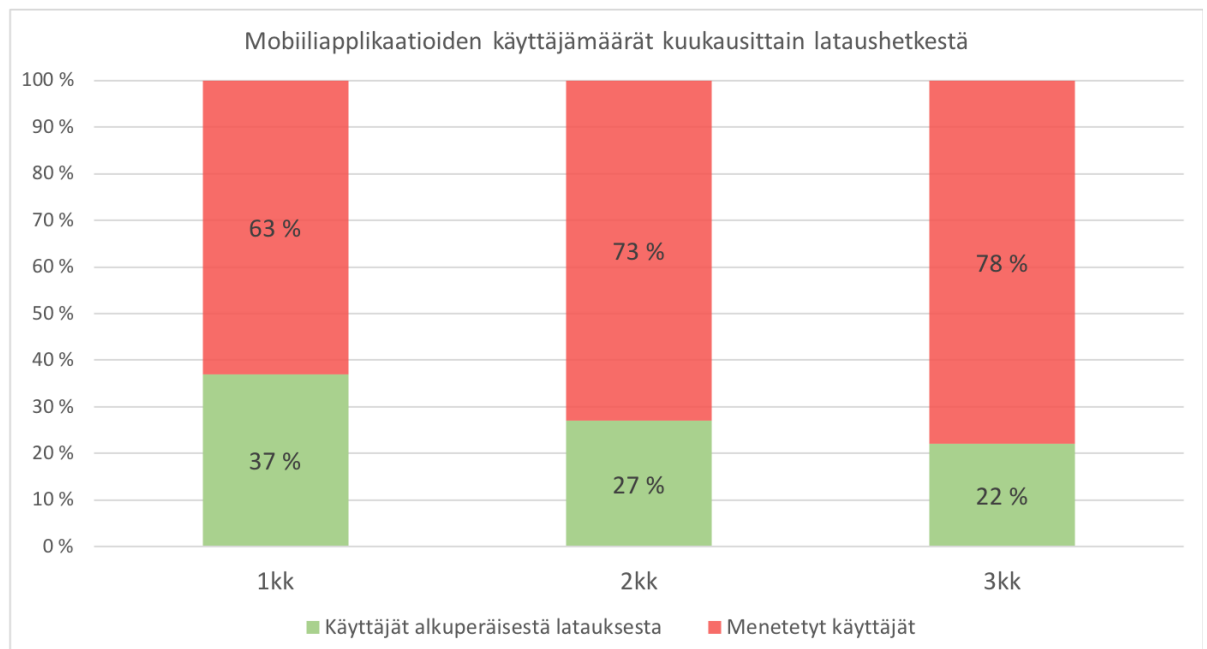
### **3.3 Natiivisovelluksesta verkkosovellukseen**

Mobiilissa toimivat internetsivut tavoittavat noin kolme kertaa enemmän käyttäjiä kuin sovelluskaupasta löytyvät sovellukset (Campbell-Moore 2016). Niinpä yritykset ja palvelut ovat alkaneet hyödyntää tätä suurta mahdollista asiakasvirtaa omaksi hyödykseen ja ovat rakentaneet verkossa toimivan progressiivisen verkkosovelluksen. Yleisesti vanhan sivuston päivittäminen uuteen progressiivisia ominaisuuksia sisältävään ratkaisuun tuo mukanaan paljon hyötyjä. Sivustot jotka käyttävät näitä ominaisuuksia latautuvat nopeammin service workereiden avulla kuin tavalliset verkkosivut ilman niitä. Ne lähettävät käyttäjilleen push-ilmoituksia ja päivittyvät vaivattomasti ilman vierailua sovelluskaupassa. Niitä voi myös käyttää ilman internetyhteyttä kuten natiivisovelluksia ja ne on helppo löytää internetistä hakukoneita käyttäen kuten mikä tahansa verkkosivusto.

Sovelluskaupoista löytyy suuri määrä sovelluksia eri käyttötarkoituksiin ja niitä ladataan päivittäin miljoonia kertoja. Keskimäärin vain 26,4 % käyttäjistä, jotka vierailevat sovelluksen sivulla, lataavat kyseisen sovelluksen. Loput käyttäjistä eivät edes testaa sovelluksen toimivuutta ja potentiaaliset asiakkaat menetetään. Amerikkalaisen media- ja analytiikkatalon ComScoren mukaan suosituimmat tuhat internetsivua, jotka toimivat responsiivisesti mobiililaitteilla, saavat 270 % enemmän käyttäjiä kuukausittain verrattuna tuhanteen suosituimpaan natiivisovellukseen. Saman tutkimuksen mukaan käyttäjät käyttävät kuitenkin 18,5 kertaa enemmän aikaa suosituimpien sovellusten käyttämiseen kuin verkkosivujen käyttämiseen. Mobiilisovelluksien parissa vietetään enemmän aikaa, koska niiden käynnistäminen on helppoa ja nopea suoraan aloitusnäytöstä ja niillä on ennen ollut pääosin selkeämpi käyttöliittymä kuin verkkosovelluksilla. Ne myös lähettävät ilmoituksia käyttäjilleen säännöllisin väliajoin ja kannustavat heitä käyttämään

enemmän aikaa sovelluksen parissa. Nämä tiedot kattavat kuitenkin vain noin neljänneksen kaikista mahdollisista käyttäjistä, sillä lataajamäärät ovat pieniä verrattuna sovellusta katsoneisiin ihmisiin. (Campbell-Moore 2016.)

Alla olevasta kuvioista 2 näkyy amerikkalaisen mobiili- ja verkkosovellusanalytiikkaan ja markkinointiin perustuvan yrityksen, Localyticsin, tutkimustulokset mobiiliapplikaatioiden käyttämiseen perustuvasta datasta. Ensimmäisen kuukauden jälkeen sovelluksen lataamisesta vain 37 % käyttäjistä käyttää sovellusta ja kolmen kuukauden jälkeen käyttäjiä on jäljellä enää vain 22 % alkuperäisistä sovelluksen lataajista. (Localytics 2017.)



Kuvio 2. Mobiiliapplikaatioiden uudelleenvierailukerrat ja menetetyt käyttäjät (Localytics 2017).

Verkkosovellukset ovat kehittyneet jo muutamassa vuodessa sille tasolle, että ne pystyvät kilpailemaan perinteisiä natiivisovelluksia vastaan. Ne pystyvät käyttämään suurta osaa ominaisuuksista, mitkä olivat ennen vain natiivisovellusten käytettävissä (Agrawal 2017). Yhdistämällä natiivi- ja verkkosovellusten parhaat puolet saadaan lopputulokseksi ratkaisu, joka miellyttää sovelluksen käyttäjiä ja saa heidät palaamaan sovelluksen pariin myös tulevaisuudessa. Verkkosivusto-

jen suurelle käyttäjämäärälle voidaan ehdottaa sovelluksen lisäämistä aloitusnäyttöön suoraan sivustolta, eikä käyttäjän tarvitse vierailla erikseen sovelluskaupassa.

### **3.4 Progressiiviset verkkosovellusratkaisut**

Monet suuret yritykset ovat siirtyneet hyödyntämään progressiivisten verkkosovellusten tuomia ominaisuuksia ja rakentaneet oman sovelluksensa. Mobiiliresponsiivisilla sivustoilla ja natiivisovelluksilla on molemmilla omat hyötynsä ja haittansa, mutta progressiiviset verkkosovellukset pyrkivät hyödyntämään molempien ratkaisun parhaita puolia. (Gazdecki 2018b.) Kolmessa seuraavassa luvussa käsitellään tunnettujen yritysten ratkaisuja, jotka hyödyntävät progressiivisiä verkkosovelluksia.

#### **3.4.1 Twitter Lite**

Vuonna 2017 Twitter uudisti mobiilisivustonsa yhteistyössä Googlen kanssa ja päivitti mobiililaitteilla toimivan verkkosivustonsa hyödyntämään progressiivisten verkkosovellusten ominaisuuksia. Twitterin käyttäjistä yli 80 % käyttää sovellusta mobiililaitteilla, ja uusi palvelu Twitter Lite kehitettiin parantamaan sivuston latausnopeutta ja luotettavuutta sekä sitouttamaan käyttäjiä paremmin. Miljoonat ihmiset selaavat sivustoa päivittäin, mutta vain pieni osa käyttää palvelua useamman kerran päivässä.

Twitter Liteen lisättiin käyttäjälle mahdollisuus asentaa sovellus suoraan aloitusnäyttöön. Lisäyksen myötä 250 000 käyttäjää aukaisee sovelluksen aloitusnäytöstään päivittäin ja käyttää sitä keskimäärin neljä kertaa päivässä. Uusi palvelu käyttää myös vähemmän dataa ja pyrkii lataamaan suurimman osan tiedostoista laitteen välimuistista. Se myös optimoi käyttäjän syötteeseen ilmestyvät kuvat ja vähentää niiden tiedostokokoa jopa 70 %, kun käyttäjä selailee aikajanaa.

Pienentämällä datan kokoa ja hyödyntämällä service workereiden mahdollistamaa välimuistiin tallentamista Twitter Litestä on tullut nopein, halvin ja luotettavin tapa käyttää Twitteriä. (Google LLC 2017a.)

### **3.4.2 Progressiivinen verkkosovellus verkkokaupan tukena**

AliExpress on verkkokauppa, joka myy kaikkea lastenvaatteiden ja kodinkoneiden väliltä suoraan Kiinasta kuluttajille. Se kuuluu kansainväliseen vähittäistuotteita myyvään verkkokauppaketjuun, Alibaba Groupiin, joka on tunnettu Yhdysvalloissa, Venäjällä ja Brasiliassa. Suurin osa käyttäjistä selaa sivustoa mobiililialustalla, joten AliExpress on jo pidemmän aikaa panostanut mobiilisivustonsa suunnitteluun ja toiminnallisuuteen. Mobiilisivuston tarkoituksena oli saada käyttäjä tykästymään palveluun ja näin ollen myös lataamaan heidän mobiilisovelluksensa. Ongelmana oli kuitenkin se, että vain pieni osa käyttäjistä lopulta latasi mobiilisovelluksen ja uusien käyttäjien hankkiminen oli haastavaa ja kallista.

AliExpress julkaisi kaikilla selaimilla toimivan progressiivisen verkkosovelluksen, joka yhdisti heidän mobiiliapplikaation ja verkkosivuston parhaat puolet yhteen. Uusi sivusto toimii nopeasti kuin natiivisovellukset, ehdottaa käyttäjälle sivuston lisäämistä aloitusnäyttöön ja sitä on mahdollista käyttää myös ilman internetyhteyttä. Uuden palvelun julkistamisen myötä uusien käyttäjien konversio nousi 104 %, käyttäjät selaavat kaksi kertaa enemmän sivuja yhden istunnon aikana ja sivustolla vierailuaika kasvoi keskimäärin 74 % kaikilla selaimilla. (Google LLC 2017b.)

### **3.4.3 Käyttäjäystävällisempiä verkkosovelluksia**

Yhdysvaltalainen talouslehti/mediatalo Forbes päivitti vuonna 2017 verkkosivustonsa hyödyntämään progressiivisten verkkosovellusten tuomia etuja. He halusivat luoda uuden, personalisoidumman ja nopeamman käyttökokemuksen käyttäjilleen. Nykyisin käyttäjä voi lisätä sovelluksen puhelimeen suoraan verkkosivus-

tolta, tilata oman mielenkiinnon mukaista sisältöä push-viesteillä sekä ladata uutisia valmiiksi luettavaksi ne myöhemmin myös ilman internetyhteyttä. (Google LLC 2017c.) Sivuston latausnopeus säästää lukijoilta aikaa 2–10 sekuntia jokaisen sivun latauksen yhteydessä (Forbes 2017). Uuden sivuston julkaisemisen jälkeen käyttäjien keskimääräisten istuntojen määrä nousi 43 % ja heille pystyttiin näyttämään 20 % enemmän mainoksia. Käyttäjät myös selasivat sivuston sisältöä 3 kertaa pidemmälle kuin ennen ja käyttäjiä saatiin sitoutettua palvelun käyttämiseen 100 % enemmän kuin ennen. (Google LLC 2017c.)

Myös isobritannialainen vaatealan verkkokauppa George.com päätti hyödyntää progressiivisten verkkosovellusten tuomia etuja ja julkaisi uuden verkkosivustonsa alkuvuodesta 2018. George.com on osa Iso-Britannian toiseksi suurinta vähittäiskaupan ketjua Asdaa, jonka omistaa yhdysvaltalainen vähittäiskauppa-ketju Walmart. Georgen entiset verkkosivut olivat vanhentuneet ajan mittaan ja asiakkaille haluttiin tarjota entistä parempi käyttökokemus verkkokaupassa. Uutta sivustoa lähdettiin kehittämään mobiililaitteita ajatellen ja siinä kiinnitettiin huomiota sivuston käyttäjäkokemuksen parantamiseen, toimintanopeuden lisäämiseen sekä mobiilissa tapahtuvan myynnin lisäämiseen. Uusi progressiivisiä ominaisuuksia käyttävä verkkosivusto latautuu 3,8 kertaa nopeammin kuin vanha sivusto. Käyttäjät selaavat verkkosivustoa 28 % pidempään ja vierailevat 20 % enemmän sivuja yhden vierailukerran aikana. Verkkokaupan konversio nousi 31 % vanhaan verkkosivustoon nähden ja verkkokaupan tietoturva päivitettiin nykyaikaan lisäämällä sivustolle HTTPS-salaus. (Google LLC 2018b.)

### **3.5 Progressiivisten verkkosovellusten liiketaloudelliset hyödyt**

Progressiivisen verkkosovelluksen kehityskulut verrattuna perinteiseen natiivisovellukseen ovat pienemmät. Sovellus tehdään toimivaksi kaikille päätelaitteille yhdellä koodipohjalla, eikä siitä tarvitse tehdä erillistä versiota eri käyttöjärjestelmille. Myös sen päivittäminen on vaivattomampaa ja nopeampaa, kun muutokset tulevat näkyviin päivityksen yhteydessä kaikille käyttäjille, eikä sitä tarvitse erikseen hyväksyttää kauppapaikan sääntöjen mukaisesti. (Zaidi 2017.) Jos yrityksen nykyinen verkkosivusto omaa hyvän käyttöliittymän valmiiksi, sillä on



mahdollisuus päivittää nykyisin käytössä olevat verkkosivut hyödyntämään progressiivisia ominaisuuksia uuden sivuston rakentamisen sijaan ja sovelluksen tekemiseen menevät kustannukset pienenevät (Icardi 2018). Näin ollen aikaa ja rahaa säästyy sovelluskehittäjien käyttäessä vähemmän työtunteja sovelluksen suunnittelusta sovelluksen julkaisemiseen.

Koska PWA:n voi ladata suoraan verkkosivustolta, yrityksen ei tarvitse miettiä sovelluksen julkaisua eri kauppapaikoissa. Apple on tunnetusti tarkka sovelluskauppaansa lisättävistä sovelluksista. Sovellusten, jotka lopulta julkaistaan App Storessa, täytyy täyttää tietyt vaatimukset ja olla ohjeiden mukainen. Lisäksi sovelluksen julkaisijan täytyy täyttää erilaisia lomakkeita ennen sovelluksen päätymistä kauppapaikalle. Kehitettäessä Android-laitteelle julkaisuprosessi Googlen Play Storeen on samankaltainen kuin Appllellakin. Android-sovelluksia voi kuitenkin julkaista monessa muussakin paikassa, kuten Amazonin App Storessa ja Ap-toidessa.

Kun sovellusta täytyy päivittää uuteen versioon, jokaisen päivityksen yhteydessä on käytävä läpi sama sovelluksen julkaisuprosessi kaikissa kauppapaikoissa, minne uusi päivitys julkaistaan. (Naylor 2017.) Yrityksen on myös mahdollista säästää paljon rahaa mahdollisissa markkinointikustannuksissa, kun sovellusta ei tarvitse markkinoida eri kauppapaikoilla. Sovellusten markkinointikustannukset eri kauppapaikoilta häviävät ja jäljelle jää vain verkkosivusto markkinoitavaksi. Esimerkiksi Housing.com-sivusto teetti olemassa olevasta verkkosivustostaan progressiivisen verkkosovelluksen ja heidän asiakashankintakustannukset natiivisovellukseen verrattuna tippuivat 3,75 Yhdysvaltain dollarista seitsemään senttiin asiakasta kohden. (Icardi 2018.)

Progressiiviset verkkosovellukset tuovat mukanaan myös hyötyjä käyttäjälleen. Ne toimivat tarpeen tullen internetyhteydestä riippumatta ja vievät vähemmän puhelimen tallennustilaa kuin natiivisovellukset. Jos käyttäjä valitsee yrityksen PWA:n natiivisovelluksen sijasta, hän käyttää vähemmän dataa ja säästää näin operaattorikuluissa. (Gazdecki 2018b.) Vertailu datan käytöstä Twitterin osalta natiivisovelluksen ja progressiivisen verkkosovelluksen välillä löytyy kappaleesta 3.3.1.

## 4 Käytännön toteutus

Ravintola Kerubi halusi kesällä 2017 uudistaa kanta-asiakastoimintansa nykyi-  
kaan ja siirtyä hyödyntämään mobiilista kanta-asiakaskorttia vanhan muovisen  
kortin sijaan. Kerubi on Joensuun Ilosaassa toimiva ruokaravintola ja esiinty-  
misareena, joka järjestää myös musiikkipainotteisia tapahtumia Joensuussa (Ke-  
rubi 2018). Kerubin kanta-asiakkailla oli ennen ollut käytössä muovinen kanta-  
asiakaskortti, jota näyttämällä sai paikan päältä tuotteita ja tapahtumia alennet-  
tuun kanta-asiakashintaan.

Ennen käytössä olleet kanta-asiakaskortit poistettiin käytöstä ja uutena palveluna  
otettiin käyttöön verkkokauppasivusto, joka hyödyntää progressiivisten verkko-  
sovellusten ominaisuuksia. Sivustolla käyttäjä voi ostaa Kerubin kanta-asiakkuu-  
den, seurata sitä koskevia ajankohtaisia tapahtumia ja uutisia sekä todentaa pai-  
kan päällä olevansa kanta-asiakas mobiilin kanta-asiakaskortin avulla. Muovisen  
kanta-asiakaskortin ollessa käytössä Kerubin henkilökunta ei tiennyt, kuinka  
monta kanta-asiakasta heillä tarkalleen oli. Kortin oston yhteydessä he keräsivät  
kanta-asiakkaan nimen ja sähköpostin talteen, mutta eivät hyödyntäneet sähkö-  
postia mahdollisena markkinointikanavana. Näin ollen projektin toisena lähtökoh-  
tana oli kanta-asiakaskortin uudistamisen ohessa myös hyödyntää kanta-asiak-  
kailta kerättyjä tietoja ja tarjota näin heille kohdistetumpaa markkinointia ja pa-  
rempaa palvelua.

Projektin aikana toteutettu kanta-asiakkailla tarkoitettu sivusto löytyy osoitteesta  
<https://kantis.kerubi.fi>. Käytän projektia opinnäytetyön käytännön toteutuksen  
esimerkkinä ja käyn läpi, kuinka projektissa hyödynnettiin progressiivisten verk-  
kosovellusten ominaisuuksia sekä kerron mahdollisista kehittämisvaihtoehdoista.

#### **4.1 Verkkokauppa-alusta Shopify**

Kanta-asiakassivuston toteuttamiseen sivustolla täytyi olla mahdollisuus myydä kanta-asiakkuuksia ja alustaksi valittiin kanadalainen verkkokauppa-alusta Shopify. Alusta oli kirjoittajalle entuudestaan tuttu ja se oli todettu toimivaksi ja kustannustehokkaaksi ratkaisuksi myös aiemmissa projekteissa.

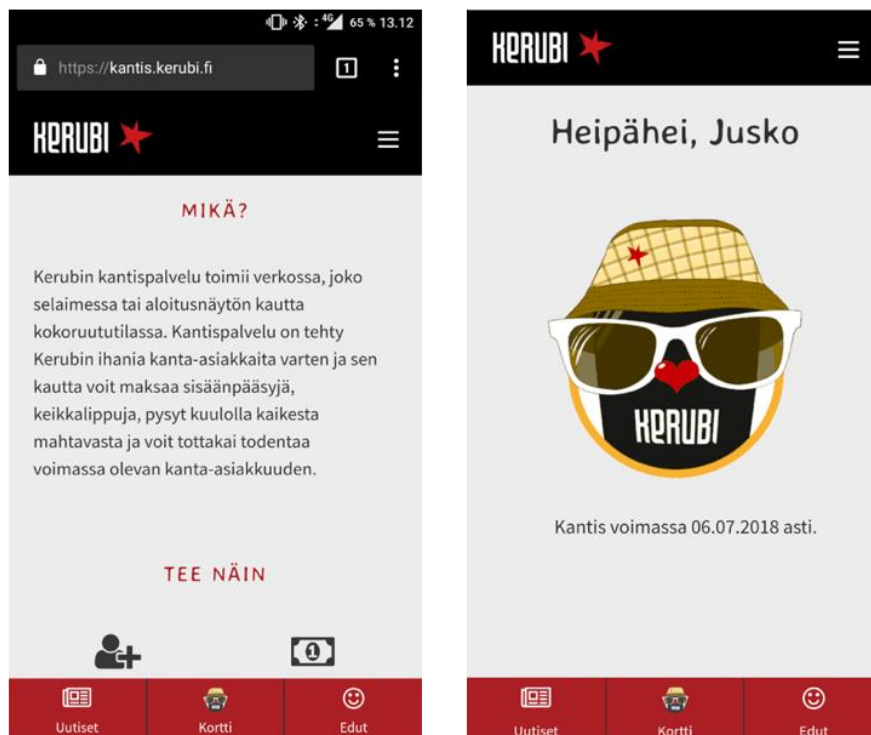
Shopify tarjoaa asiakkailleen täysin hostatun ja HTTPS-suojatun palvelun, jonka avulla asiakkaat voivat perustaa verkkokaupan kiinteään kuukausihintaan. Sillä on jo yli miljoona aktiivista käyttäjää ja sitä käyttää 600 000 yritystä ympäri maailman. Shopifyyn voi kirjautua miltä tahansa laitteelta internetselaimella tai käyttäen sen kehittämää sovellusta, joka löytyy yleisimmistä sovelluskaupoista. Shopify tarjoaa käyttäjilleen myös laajan sovelluskaupan, mistä käyttäjät voivat ladata lisäosia verkkokauppoihinsa joko kuukausihinnonnoittelulla tai kertamaksulla. Sovelluskaupan avulla verkkokauppiaan ei tarvitse itse lähteä kehittämään tarvittavia ominaisuuksia kauppaansa, vaan ne voi lisätä helposti muutaman napin painalluksella. (Shopify 2018.)

Shopifystä on saatavilla kolme eri kuukausihintaista pakettia: 29, 79 ja 299 dollaria kuukaudessa. Jokaisen paketin kuukausihintaan kuuluu muun muassa verkkokaupan ylläpito HTTPS-salauksella, ilmainen domain verkkokaupalle (muotoa [www.kaupan-nimi.myshopify.com](http://www.kaupan-nimi.myshopify.com)), verkkomaksut, sivueditori sekä blogialusta. Halvin vaihtoehto sisältää kaiken tarvittavan aloittelevalle verkkokaupalle ja nostamalla kuukausipaketin tasoa saa käyttöönsä enemmän ominaisuuksia Shopifyssä sekä esimerkiksi transaktiomaksujen provisio pienenee. (Shopify 2018.)

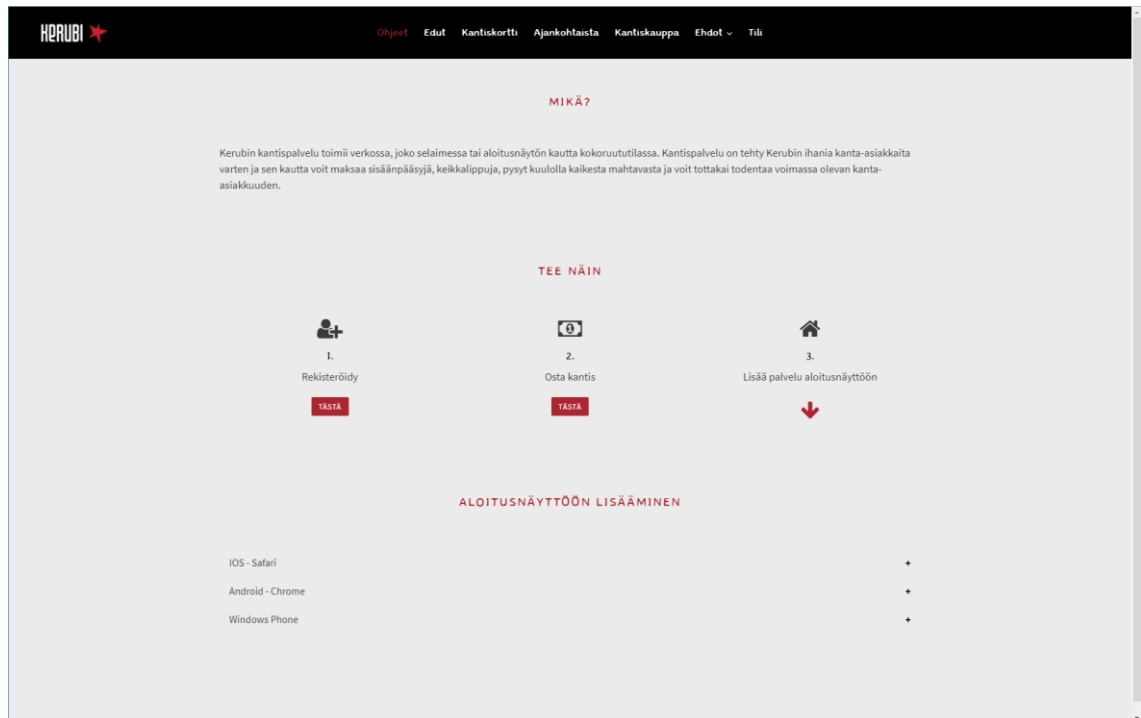
#### **4.2 Kanta-asiakassivusto**

Kanta-asiakassivusto on suunniteltu käytettäväksi mobiililaitteilla, mutta responsiivisuutensa ansiosta se toimii myös isommilla näytöillä. Kuvassa 9 näkyy, miltä sivusto näyttää tietokoneen näytöltä (1920x1080 px) katsottaessa. Sivustolle rakennettiin sovelluksen tapainen käyttöliittymä ja mobiililaitteella selattaessa si-

vuston alalaitaan ilmestyy navigaattorivi, mistä voi nopeasti siirtyä kanta-asiakassivulle todentamaan jäsenyytensä. Seuraavassa kuvasarjassa on esitetty, miltä verkkosivusto näyttää mobiililaitteella, kun sitä katsotaan Google Chrome -selaimella (kuva 8, vasen) sekä miltä se näyttää, kun se aukaistaan laitteen aloitusnäytöstä (kuva 8, oikea). Kun sovellus aukaistaan aloitusnäytöstä, selaimen navigaatiopalkki häviää käyttäjän näkymästä ja sovellus aukaistaan kokoruuduntilassa.



Kuva 8. Kuvasarja Kerubin kanta-asiakassivustosta Android-laitteella.



Kuva 9. Kuvakaappaus Kerubin sivustosta suurempaa resoluutiota käyttäen.

### 4.3 Käytetyt teknologiat

Kerubin kanta-asiakassivustolle lisättiin käyttäjille mahdollisuus kiinnittää sovellus aloitusnäyttöön ja käyttää sitä natiivisovelluksen tapaan manifest.json-tiedoston avulla. Kun sovellus aukaistaan laitteen aloitusnäytön kautta se aukeaa koko näytön tilassa ja aukaisee suoraan kanta-asiakaskortti sivun, jolla käyttäjä voi todentaa kuuluvansa Kerubin kanta-asiakkaaksi. Manifest-tiedostossa on myös määritelty eri kokoiset kuvat näytölle lisättävään ikoniin. Seuraavasta kuvasta 10 näkyy sivustolta löytyvä manifest.json-tiedosto.

```
{
  "short_name": "Kerubi",
  "name": "Kerubin kantispalvelu",
  "icons": [
    {
      "src": "https://cdn.shopify.com/s/files/1/1977/8349/files/96kerubi.png",
      "sizes": "96x96",
      "type": "image/png"
    },
    {
      "src": "https://cdn.shopify.com/s/files/1/1977/8349/files/144kerubi.png",
      "sizes": "144x144",
      "type": "image/png"
    },
    {
      "src": "https://cdn.shopify.com/s/files/1/1977/8349/files/192kerubi.png",
      "sizes": "192x192",
      "type": "image/png"
    }
  ],
  "start_url": "https://kantis.kerubi.fi/pages/kantiskortti/",
  "display": "fullscreen",
  "orientation": "portrait"
}
```

Kuva 10. Kerubin verkkosivustolle lisätty manifest -tiedosto.

Kesällä 2017 Applen Safari-selain ei tukenut progressiivisten verkkosovellusten web app manifestia, joten sivuston HTML <head> -osioon on lisätty Applen metatageja. Niiden avulla sovelluksen käytön saa samankaltaiseksi Safarilla kuin progressiivisen verkkosovelluksen käytön Androidilla. Metatageissa määritellään kuvat erikokoisille laitteille, mitä käytetään sovelluskoneina, kun sovellus lisätään aloitusnäyttöön. Tagien avulla sovellus saadaan myös toimimaan koko näytön

tilassa sekä määritellään sovelluksen aloitusnäyttöön lisättävä nimi. Kuvassa 11 kuvataan Kerubin verkkosivustolle lisätyt Applen metatagit.

```
<link rel="apple-touch-startup-image"
href="https://cdn.shopify.com/s/files/1/1977/8349/files/kerubi.png?311947958
0266514967">
<link rel="apple-touch-icon"
href="https://cdn.shopify.com/s/files/1/1977/8349/files/kerubi.png?311947958
0266514967">
<link rel="apple-touch-icon" sizes="152x152"
href="https://cdn.shopify.com/s/files/1/1977/8349/files/152kerubi.png?743512
9920324904235">
<link rel="apple-touch-icon" sizes="167x167"
href="https://cdn.shopify.com/s/files/1/1977/8349/files/167kerubi.png?743512
9920324904235">
<link rel="apple-touch-icon" sizes="180x180"
href="https://cdn.shopify.com/s/files/1/1977/8349/files/180kerubi.png?743512
9920324904235">
<meta name="apple-mobile-web-app-title" content="Kerubi">
<meta name="apple-mobile-web-app-capable" content="yes">
<meta name="apple-mobile-web-app-status-bar-style" content="black-translu-
cent">
```

Kuva 11. Kerubin sivustolle HTML tiedoston <head>-osioon lisätyt Applen metatagit.

Koska Safari ei tukenut progressiivisten verkkosovellusten manifest-tiedostoa, lisättiin sivulle myös yksi JavaScript-tiedosto (kuva 12), jonka avulla tarkastellaan sivustolla löytyviä linkkejä. Jos linkki on sivuston sisäinen eikä juuriosoite muutu, niin se aukaistaan kokonäytön tilassa. Jos linkki vie ulkoiseen lähteeseen eli pois Kerubin sivustolta, se aukaistaan uudessa ikkunassa oletusselaimella.

```
<script>
// Mobile Safari in standalone mode
if(("standalone" in window.navigator) && window.navigator.standalone){
  var noddy, remotes = false;
  document.addEventListener('click', function(event) {
    noddy = event.target;

    while(noddy.nodeName !== "A" && noddy.nodeName !== "HTML") {
      noddy = noddy.parentNode;
    }
    if('href' in noddy && noddy.href.indexOf('http') !== -1 && (noddy.href.indexOf(document.location.host) !== -1 || remotes))
    {
      event.preventDefault();
      document.location.href = noddy.href;
    }
  },false);
}
</script>
```

Kuva 12. Kerubin sivustolle lisätty JavaScript-koodi, joka tarkistaa sivustolta löytyvien linkkien kohteen.

#### 4.4 Kehitysvaihtoehdot

Service workereiden käyttäminen kyseisessä projektissa ei ollut mahdollista projektin tekohetkellä. Niiden rekisteröimiseen ja käyttämiseen service worker -tiedoston täytyy sijaita verkkosivuston juurikansiossa ([www.esimerkkisivu.fi/serviceworker.js](http://www.esimerkkisivu.fi/serviceworker.js)), eikä Shopifyllä tämä ollut mahdollista. Shopifyllä tiedostot ladataan sen omaan sisällönjakeluverkkoon (CDN) ja jaetaan sieltä käyttöä varten. Ongelma ilmenee service workeria ladattaessa, kun jakeluosoite ei täsmää verkko-



sivuston kanssa. Service workerit on mahdollista ottaa käyttöön myöhemmin tulevaisuudessa, kun progressiiviset verkkosovellukset yleistyvät ja tulevat käyttöön laajemmin ja pääsy verkkokaupan juurikansioon sallitaan.

Service workereiden avulla progressiiviset verkkosovellukset saavat käyttöönsä push-viestien lähetyksen ja mahdollisuuden toimia ilman internetyhteyttä offline-tilassa. Push-viestejä varten Kerubille ladattiin Shopify:n sovelluskaupasta sovellus, jonka avulla pystytään lähettämään Chrome-selainta käyttäville kanta-asiakkaille ilmoituksia tapahtumista ja ajankohtaisista uutisista. Kerubin tiloista löytyy asiakkaille tarkoitettu Wi-Fi-yhteys, jonka ansiosta offline-tuen lisääminen sovellukselle ei ole välttämätöntä.

Kerubi järjestää tiloissaan erilaisia tapahtumia ja myy myös itse niihin lippuja. Yleensä kanta-asiakkaille on tarjolla halvemmat liput tapahtumiin ja lippuja pystyy nykyisin ostamaan Kerubin ravintolan tiskiltä alennettuun kanta-asiakashintaan. Koska uusi kanta-asiakassivusto on toteutettu verkkokauppa-alustalla, Kerubilla olisi mahdollisuus myydä kanta-asiakaslippuja tapahtumiin myös suoraan verkkosivustolta.

#### **4.5 Tulokset ja kokemukset**

Projektin lopputuloksena syntyi toimiva uusi palvelu Kerubin kanta-asiakkaille. Kanta-asiakaskortin siirtyminen mobiililaitteille on kustannustehokkaampi ratkaisu verrattuna vanhaan fyysiseen korttiin. Sen ylläpitokustannukset ovat pienemmät ja sen avulla voidaan seurata tarkkaa määrää, kuinka moni ihminen todella käyttää palvelua ja kuuluu kanta-asiakasohjelmaan. Verkkosivustolle rekisteröidytään sähköpostiosoitteella ja Kerubin kanta-asiakkaat saavat säännöllisin väliajoin uutiskirjeen, missä kerrotaan kanta-asiakkaille suunnatuista tulevista tapahtumista sekä eduista.

Kanta-asiakkaille pystytään nykyään tarjoamaan parempaa ja laajempaa palvelua uuden kanavan kautta. Shopifysta löytyvän blogialustan ansiosta kanta-asiakkaille voidaan ilmoittaa uusista tapahtumista ja tarjouksista myös blogialustan

kautta. Sivustolta ostettu kanta-asiakkuus on voimassa vuoden eteenpäin ostopäivästä. Vuoden kuluttua ostosta käyttäjän täytyy ostaa kortti uudestaan saadaakseen pääsyn kanta-asiakaskorttisivulle. Aikaisemmin muoviset kanta-asiakskortit uusittiin kesäisin ja niiden voimassaoloaika oli ostohetkestä riippumatta heinäkuuhun ja kortti oli myös samanhintainen riippumatta ostoajankohdasta. Uuden ratkaisun avulla kanta-asiakkaille voidaan tarjota vuoden kanta-asiakkuus ostohetkestä riippumatta.

Projekti oli mielekäs toteuttaa, koska sen lopputuloksena syntyi konkreettinen uusi palvelu, joka on Kerubin kanta-asiakkaiden jokapäiväisessä käytössä. Uuden palvelun avulla Kerubi siirsi kanta-asiakastoimintansa nykyaikaan ja pystyy nyt tarjoamaan kanta-asiakkailleen ajankohtaisempaa palvelua. Projektin aikana kirjoittaja oppi myös paljon uutta aihealueeseen liittyvää. Esimerkiksi kaikkia progressiivisia ominaisuuksia ei voi suoraan hyödyntää sivustolla, jos service worker -tiedostoa ei pysty jakamaan sivuston juurikansiosta. Verkkokauppa-alustan hyödyntäminen projektissa osoittautui hyväksi ratkaisuksi, vaikka kaikkia progressiivisten verkkosovellusten ominaisuuksia ei Shopifyllä voi vielä täysin hyödyntää. Verkkokaupan ominaisuudet tuovat paljon lisäarvoa yritykselle, koska sen kautta voidaan myydä ja markkinoida yrityksen tarjoamia palveluita tai tuotteita suoraan kohdistetulle asiakasryhmälle.

Projekti toteutettiin suunnitellulla aikataululla ilman suurempia ongelmia. Sivusto julkaistiin pysyvästi kanta-asiakkaiden käyttöön elokuussa 2017. Projekti onnistui kirjoittajan mielestä hyvin ja lopputulokseen oltiin myös tyytyväisiä Kerubin puolelta. Palvelu on ollut kirjoitushetkellä käytössä noin kymmenen kuukautta, eikä sen toimivuudessa ole ollut puutteita.

## **5 Pohdinta**

Progressiiviset verkkosovellukset ovat jo muutaman vuoden aikana kehittyneet sille tasolle, että ne voivat haastaa perinteiset sovelluskehitysmenetelmät ja jopa korvata olemassa olevia sovelluksia joustavimmilla ratkaisuillaan. Ne ovat joukko

kehittyneitä verkossa toimivia teknologioita, jotka yhdessä tarjoavat kokonaisvaltaisen sovellusratkaisun perinteisten sovelluskehitysmenetelmien ohelle. PWA:t tarjoavat suuren osan ominaisuuksista, jotka olivat ennen käytössä vain natiivisovelluskehityksen puolella. Niillä pystytään kehittämään verkkosovelluksia, jotka yhdistävät parhaat puolet verkkosivustoista ja natiivisovelluksista yhteen ratkaisuun. Tulevaisuudessa tuettujen ominaisuuksien määrä kasvaa suuremmaksi ja kaikki ominaisuudet, joita natiivisovelluksetkin käyttävät, voivat tulla verkkosovellusten käyttöön. Se lisää sovelluskehitysmarkkinoille uuden haastajan natiivi- ja hybridisovellusratkaisujen rinnalle.

Progressiiviset verkkosovellukset tarjoavat sovelluskehittäjille mahdollisuuden rakentaa nopeita, käyttäjäystävällisiä ja luotettavia sovelluksia verkkoon. Ne ovat kustannustehokas vaihtoehto perinteisten sovelluskaupoista ladattavien sovelluksien ohelle. Vaikka ne eivät vielä pysty käyttämään kaikkia ominaisuuksia, mitä natiivisovellukset pystyvät tarjoamaan, niiden suosio tulee lähivuosina kasvamaan koska niillä voidaan toteuttaa suuri määrä mahdollisista käyttötapauksista.

Progressiivinen verkkosovellus sopii parhaiten yrityksille, jotka eivät tarvitse sovellukselta pääsyä laitteen kaikkiin ominaisuuksiin, ja jotka eivät ole varanneet sovelluksen kehitykselle suurta budjettia. Progressiivisen verkkosovelluksen kehityskulut ovat huomattavasti pienemmät verrattuna natiivisovelluksiin, koska yksi ja sama sovellus toimii kaikilla päätelaitteilla samalla tavalla, eikä sovellusta tarvitse erikseen julkaista eri käyttöjärjestelmille. PWA:t ovat myös käyttäjän näkökulmasta hyödyllinen lisä sovellusmarkkinoille. Ne ovat tiedostokooltaan pieniä ja niiden lataaminen sekä päivittäminen vie huomattavasti vähemmän dataa perinteisiin sovelluksiin verrattuna. Suomessa datan pienentämisellä ja ladattavan sovelluksen koolla ei ole niin suurta vaikutusta, koska 70 % suomalaisista käyttää liittymää, jossa on käytössä rajaton määrä dataa (Viestintävirasto 2018). Sen sijaan esimerkiksi Yhdysvalloissa vain 2 % mobiililiittymistä käyttää rajattoman datan liittymää ja käyttäjä joutuu maksamaan datansiirrosta, jota verkkosivustojen ja sovellusten lataamiseen tarvitaan (WhatPhone 2018). Esimerkkinä Twitter Lite PWA:n lataaminen vie 600 KB dataa, kun taas Android-laitteelle saatavilla oleva

sovellus on kooltaan 23.5 MB, eli 39 kertaa suurempi kuin saatavilla oleva progressiivinen verkkosovellus.

Kerubin kanta-asiakassovellusta tehdessä näin, kuinka progressiivisten verkkosovellusten ominaisuuksia voidaan hyödyntää käytännössä. Huomasin, kuinka niiden käyttöönottoaminen on helppoa ja nopeaa. Ne voidaan julkaista lyhyessä ajassa, koska ne ovat kehitetty verkkosovelluskehittäjille jo tunnetuilla teknologioilla. Progressiivisen verkkosovelluksen kehittäminen on myös kustannustehokas vaihtoehto, koska sitä ei tarvitse kehittää erikseen eri käyttöjärjestelmille, eikä julkaisua tarvitse miettiä mahdollisille kauppapaikoille. Verkkokauppa-alustan hyödyntäminen sovelluksen runkona mahdollistaa suoran maksupalveluiden käyttöönoton, mikä myös vähentää kehitykseen vaadittavaa aikaa ja kuluja. Projekti pystyttiin suorittamaan nopealla aikataululla ja jo ensimmäisen kahden kuukauden aikana käyttäjämäärät nousivat odotusten mukaisiin lukemiin.

Opinnäytetyön aihe oli tuore, eikä siitä ollut saatavilla paljoa kirjallista lähdeaineistoa. Tämän takia opinnäytetyön lähteinä onkin käytetty enimmäkseen verkkomateriaalia ja -lähteitä. Verkkosovelluskehitys kehittyy nopeaan tahtiin, eikä kirjallisuus pysy kehittyvän ympäristön tahdissa. Jos aiheesta kirjoittaisi kirjan, pitäisi sen julkaisuaikataulu tehdä lyhyellä aikajänteellä, sillä kehittyvän ympäristön myötä toimivat ratkaisut ja tiedot voivat vanheta muutaman kuukauden aikana. Aiheeseen liittyviä lähteitä ei löytynyt suomenkielisiltä sivustoilta kuin rajoitetusti, joten suurin osa lähdemateriaalista onkin englanninkielisiltä sivustoilta. Progressiiviset verkkosovellukset ovat maailmalla ja erityisesti Suomessa vielä suhteellisen uusi käsitys eikä niiden olemassaolo ole vielä tullut esille suuremmalle käyttäjäkunnalle.

Kaiken kaikkiaan koin tämän opinnäytetyön toteuttamisen todella mielekkääksi. Aiheen tuoreus ja täten kotimaisten lähteiden niukkuus toi toteuttamiseen omat haasteensa, mutta se osoittautui myös positiiviseksi asiaksi. Ulkomaalaiset lähteet muodostuivat todella merkityksellisiksi juurikin sen vuoksi, että aihe on suhteellisen tuntematon Suomessa. Opinnäytetyössä esittelemäni tapausesimerkit, kuten Twitter Lite ja talouslehti Forbes, toimivat mallina muille yrityksille, joita siir-

tyminen käyttämään uuden aikakauden verkkosovelluksia kiinnostaa. Opinnäytetyön teoriaosuuden kirjoittamisen myötä opin todella paljon uutta, vaikka olin toteuttanut työn toiminnallisen osuuden jo aiemmin. Opinnäytetyön toteuttaminen on tuonut minulle uusia näkökulmia ja uutta tietotaitoa verkkosovelluskehityksestä ja olen innokas hyödyntämään oppimaani jatkossakin. Uskon, että progressiiviset verkkosovellukset saavuttavat suuren suosion Suomessa ja muualla maailmassa.

## Lähteet

- Agrawal, T. 2017. Why Progressive Web Apps Are The Future Of Web Development. <https://medium.freecodecamp.org/why-progressive-web-apps-are-the-future-of-web-development-13db7dd5f640>. 28.2.2018.
- Asay, M. 2016. Apple could lose billions on Progressive Web Apps, but it has no choice. <https://www.techrepublic.com/article/apple-could-lose-billions-on-progressive-web-apps-but-it-has-no-choice/>. 7.3.2018.
- Campbell-Moore, O. 2016. The surprising tradeoff at the center of the question whether to build a Native or Web App. <https://medium.com/@owencm/the-surprising-tradeoff-at-the-center-of-question-whether-to-build-an-native-or-web-app-d2ad00c40fb2>. 27.2.2018.
- Caniuse. 2018a. Can I use service workers. <https://caniuse.com/#search=service%20workers>. 6.3.2018.
- Caniuse. 2018b. Can I use web app manifest. <https://caniuse.com/#search=web%20app%20manifest>. 6.3.2018.
- Dascalescu, D. 2018. Why “Progressive Web Apps vs. native” is the wrong question to ask. <https://medium.com/dev-channel/why-progressive-web-apps-vs-native-is-the-wrong-question-to-ask-fb8555addcbb>. 7.3.2018.
- Fablet, Y. 2018. Workers at Your Service. <https://webkit.org/blog/8090/workers-at-your-service/>. 7.3.2018.
- Farrugia, K. 2016. A Beginner’s Guide To Progressive Web Apps. Smashing Magazine. <https://www.smashingmagazine.com/2016/08/a-beginners-guide-to-progressive-web-apps/>. 6.1.2018
- Firtman, M. 2018. PWAs are coming to iOS 11.3: Cupertino, we have a problem. <https://medium.com/@firt/pwas-are-coming-to-ios-11-3-cupertino-we-have-a-problem-2ff49fd7d6ea>. 6.3.2018.
- Fisher, D. 2015. Chrome Dev Summit 2015: That’s a wrap!. Chromium Blog. <https://blog.chromium.org/2015/11/chrome-dev-summit-2015-thats-wrap.html>. 9.2.2018
- Forbes. 2017. Forbes Launches All-New Mobile Web Experience for Forbes.com. <https://www.forbes.com/sites/forbespr/2017/03/07/forbes-launches-all-new-mobile-web-experience-for-forbes-com/#7da9ca9b5168>. 2.3.2018.
- Garg, S. 2017. Web App Manifest File— Make your web app Installable. <https://medium.com/@subodhgarg/web-app-manifest-file-make-your-web-app-installable-b5fcdb2919b9>. 6.3.2018.
- Gaunt, M & Kinlan, P. 2018. The Web App Manifest. <https://developers.google.com/web/fundamentals/web-app-manifest/>. 12.2.2018.
- Gaunt, M. 2018. Service Workers: an Introduction. <https://developers.google.com/web/fundamentals/primers/service-workers/>. 12.2.2018.
- Gazdecki, A. 2018a. Progressive web apps vs native apps: Showdown in 2018. <https://appdeveloperomagazine.com/5877/2018/3/1/progressive-web-apps-vs-native-apps:-showdown-in-2018/>. 13.4.2018.

- Gazdecki, A. 2018b. Why Progressive Web Apps Will Replace Native Mobile Apps. <https://www.forbes.com/sites/forbestechcouncil/2018/03/09/why-progressive-web-apps-will-replace-native-mobile-apps/#61e6eae82112>. 1.4.2018.
- Google LLC. 2017a. Twitter Lite PWA Significantly Increases Engagement and Reduces Data Usage. <https://developers.google.com/web/showcase/2017/twitter>. 28.2.2018.
- Google LLC. 2017b. AliExpress. <https://developers.google.com/web/showcase/2016/aliexpress>. 2.3.2018.
- Google LLC. 2017c. Forbes. <https://developers.google.com/web/showcase/2017/forbes>. 2.3.2018.
- Google LLC. 2018a. Progressive Web Apps. <https://developers.google.com/web/progressive-web-apps/>. 16.1.2018.
- Google LLC. 2018b. George.com enhances the mobile customer experience with new Progressive Web App. <https://developers.google.com/web/showcase/2018/asda-george>. 23.4.2018.
- Honkonen, K. 2017. TIVIA News: Digitaalisten lähiympäristöjen tekijä. Tivia. <http://www.tivia.fi/tiviassa-tapahtuu/tivia-news/digitaalisten-lahiymparistojen-tekija>. 7.1.2018
- Icardi, E. 2018. Should enterprises start building progressive web apps?. <https://techbeacon.com/should-enterprises-start-building-progressive-web-apps>. 13.4.2018.
- Kerubi. 2018. <http://kerubi.fi/>. 23.4.2018.
- Localytics. 2017. Cheat Sheet Overall App Benchmarks H1 2017. <https://www.localytics.com/lp/cheat-sheet-overall-app-benchmarks-h1-2017/>. 6.3.2018.
- Naylor, I. 2017. Progressive Web Apps vs Native Apps – Who Wins?. <https://appinstitute.com/pwa-vs-native-apps/>. 2.4.2018.
- Osmani, A. 2018. The App Shell Model. <https://developers.google.com/web/fundamentals/architecture/app-shell>. 18.1.2018.
- PWA Rocks. 2018. Progressive web apps. <https://pwa.rocks/>. 13.2.2018.
- Russell, A. 2015. Progressive Web Apps: Escaping Tabs Without Losing Our Soul. <https://medium.com/@slightlylate/progressive-apps-escaping-tabs-without-losing-our-soul-3b93a8561955>. 13.4.2018.
- Shopify. Set up your store, pick a plan later. <https://www.shopify.com/pricing/>. 13.4.2018.
- StatCounter. 2018. Mobile Browser Market Share Worldwide. <http://gs.statcounter.com/browser-market-share/mobile/worldwide>. 5.3.2018.
- The Mozilla Foundation. 2018a. Web App Manifest. <https://developer.mozilla.org/en-US/docs/Web/Manifest>. 12.2.2018
- The Mozilla Foundation. 2018b. Progressive Web Apps are here. What's the big deal?. <https://blog.mozilla.org/firefox/progressive-web-apps-whats-big-deal/>. 17.4.2018.
- Uotila, M. 2017. Progressive web appit haastavat jo mobiilisovellukset. <https://qvik.com/news/progressive-web-app/>. 13.4.2018.
- Viestintävirasto. 2018. Matkaviestinverkon liittymät tiedonsiirtosopimuksen mukaan. <https://www.viestintavirasto.fi/tilastotjatutkimukset/tilastot/2013/matkaviestinverkonliittymattiedonsiirtosopimuksenmukaan.html>. 19.4.2018.

- What Web Can Do Today. 2018. <https://whatwebcando.today/>. 7.3.2018.
- WhatPhone. 2018. <https://whatphone.com.au/sim-only-byo-plans/best-prepaid-plans>. 19.4.2018.
- Zaidi, M. 2017. Native Mobile Apps vs React Native Apps vs Progressive Web Apps. [https://medium.com/@murtazazaidi\\_/native-mobile-apps-vs-react-native-apps-vs-progressive-web-apps-be81b314cdec](https://medium.com/@murtazazaidi_/native-mobile-apps-vs-react-native-apps-vs-progressive-web-apps-be81b314cdec). 2.4.2018.