

Responsiivinen sivusto ja erillinen applikaatio mobiiliratkaisuina

Tommi Soikkeli



Tekijä(t) Tommi Soikkeli	
Koulutusohjelma Tietojenkäsittelyn koulutusohjelma	
Opinnäytetyön nimi Responsiivinen sivusto ja erillinen applikaatio mobiiliratkaisuina	Sivu- ja liitesivumäärä 30 + 2
<p>Digitaalisten palveluiden käyttö mobiililaitteilla on yhä yleisempää. Mobiililaitteilla tapahtuva Internet-liikenne kattaa jo noin puolet kokonaisliikenteestä. Mobiililaitteilla palveluita käyttävien käyttäjien huomioiminen käyttäjäkokemuksessa on ensisijaisen tärkeää. Tämän tutkimustyypisen opinnäytetyön tarkoituksena oli tutustua kahteen mobiilipalveluiden kehitystapaan – responsiivisiin sivustoihin sekä erillisiin mobiilisovelluksiin.</p> <p>Työn tavoitteena oli saada selville, kuinka nämä menetelmät soveltuvat mobiiliratkaisuiksi sekä tutkia, kumpaa ratkaisutavoista käyttäjät suosivat digitaalisten palveluiden käytössä mobiililaitteilla.</p> <p>Teoriaosuudessa tutustuttiin ratkaisutapojen toteutustekniikoihin, tuettuihin ominaisuuksiin sekä hyötyihin ja haasteisiin. Työn empiirinen osio toteutettiin määrällisenä kyselytutkimuksena huhtikuussa 2019. Kysely suunnattiin mobiilipalveluiden käyttäjille, ja sen avulla saatiin kerättyä tietoa käyttäjien tottumuksista ja mielipiteistä mobiilipalveluiden käyttöön liittyen. Lisäksi kyselyn tavoitteena oli selvittää, kumpaa ratkaisutapaa käyttäjät suosivat ja vaikuttaako kenties palvelun tyyppi ratkaisutavan valintaan.</p> <p>Tutkimuksen tulokset osoittivat mobiilisovellusten olevan käyttäjien suosima vaihtoehto mobiiliratkaisuksi. Mobiilisovellukset pystyvät tarjoamaan responsiivista sivustoa paremmin mobiililaitteille optimoidun käyttäjäkokemuksen paremman käytettävyyden sekä laajojen laiteominaisuuksien hyödyntämisen myötä.</p> <p>Mobiiliratkaisujen tulevaisuutta pohtiessa esille on syytä nostaa mobiilisovellusten kaltaiset verkkosivustot eli progressiiviset web-sovellukset, joiden yleistyessä mobiilisovellukset ovat saamassa varteenotettavan haastajan mobiiliratkaisun muotona.</p>	
Asiasanat Mobiilisovellukset, responsiivisuus, mobiiliratkaisu, sovelluskehitys, käyttäjäkokemus	

Sisällys

1	Johdanto	1
2	Responsiivinen sivusto.....	2
2.1	Toteutustekniikat.....	2
2.2	CSS-ohjelmistokehykset.....	5
2.3	Tuetut ominaisuudet	6
2.4	Hyödyt ja haasteet	7
3	Mobiilisovellukset	8
3.1	Toteutustekniikat natiivisovelluksille.....	8
3.2	Muut toteutustekniikat.....	9
3.3	Tuetut ominaisuudet	11
3.4	Hyödyt ja haasteet	12
4	Käyttäjätutkimus	14
4.1	Tutkimusmenetelmän kuvaus.....	14
4.2	Tutkimuksen tulokset	15
5	Johtopäätökset ja pohdinta	23
5.1	Johtopäätökset.....	23
5.2	Pohdinta	25
	Lähteet	27
	Liitteet.....	31
	Liite 1. Kyselytutkimus.....	31

1 Johdanto

Mobiililaitteilla tapahtuva Internet-liikenne kattaa nykyisillään jo noin puolet kokonaisliikenteestä (Enge 2018). Mobiililaitteiden suosion myötä on syntynyt tarve huomioida Internet-sivustojen käyttöliittymä ja käyttäjäkokemus kaiken kokoisilla näytöillä. Sivuston toimimattomuus ja heikko käyttäjäkokemus mobiililaitteilla johtaa käyttäjän turhautumiseen ja mahdollisesti saa käyttäjän hakemaan ratkaisua muualta. Ohjelmistokehityksessä mobiilikäyttäjien huomiointiin on vakiintunut kaksi ratkaisutapaa: responsiiviset eli mobiililaitteille mukautuvat sivustot sekä erilliset, mobiililaitteeseen asennettavat mobiilisovellukset. Nämä menetelmät eroavat toisistaan erityisesti toteutustekniikoiden osalta, mutta myös ominaisuuksiltaan, joita ne pystyvät käyttäjälle tarjoamaan käyttäjäkokemuksen parantamiseksi.

Tämän opinnäytetyön tarkoituksena on tutustua tarkemmin näihin kahteen ratkaisutapaan. Teoriaosuudessa tutustutaan molempien ratkaisujen toteutustekniikoihin, hyötyihin, haasteisiin sekä ominaisuuksiin, joita ratkaisutavat tarjoavat mobiilikäyttäjille. Ensin tutustutaan responsiivisiin sivustoihin luvussa 2, jonka jälkeen tutustutaan mobiiliapplikaatioihin luvussa 3.

Tutkimusosuudessa toteutetaan teorian tueksi kyselytutkimus, jonka avulla kartoitetaan mobiilikäyttäjien kokemuksia ja mielipiteitä molempiin ratkaisutapoihin liittyen. Tavoitteena on selvittää, onko jompikumpi ratkaisutavoista tavallisten mobiilikäyttäjien mielestä selvästi mieluisampi vai vaikuttaako palvelun toimiala ratkaisun mielekkyyteen.

Teorian ja tutkimuksen tulosten pohjalta tarkoituksena on saada vastaus seuraaviin tutkimuskysymyksiin:

- Miten responsiivinen sivusto ja erillinen mobiilisovellus soveltuvat mobiiliratkaisuiksi?
- Vaikuttaako digitaalisen palvelun tyyppi tai toimiala mobiiliratkaisun valintaan?

2 Responsiivinen sivusto

Sivuston responsiivisuudella tarkoitetaan yksinkertaisuudessaan sitä, että sivuston sisältö mukautuu sopivaksi kaiken kokoisille näytöille. Responsiivinen sivusto reagoi näyttökoon muutokseen, esimerkiksi selainikkunan pienentämiseen ja mukauttaa sisältöään sen mukaan. Sisällön mukauttaminen voi tarkoittaa sivuston elementtien, kuten kuvien, tekstien ja painikkeiden pienentämistä, leveyden muokkaamista tai kokonaisten elementtien piilottamista. Responsiivinen sivusto pyrkii takaamaan laadukkaan käyttäjäkokemuksen kaikille käyttäjille laitteesta riippumatta. (Smashing Magazine 2011.)

Artikkelissaan *Responsive web design* Ethan Marcotte (2010) esitteli ensimmäistä kertaa responsiivisen suunnittelun käsitteen. Älypuhelinien ja tablettien yleistyessä verkkosivustojen suunnittelussa haasteena oli suuri määrä laitteita: tuleeko jokaiselle laitteelle suunnitella oma versio sivustosta? Marcotte esitteli responsiivisen suunnittelun käsitteen ajatuksena, jonka mukaan verkkosivustojen suunnittelussa tulee hyödyntää olemassa olevia, standardisoituja teknologioita, joiden avulla sivustot osaavat mukautua laitteisiin, joilla niitä luetaan. (Marcotte 2010.)

2.1 Toteutustekniikat

Sivuston responsiivisuus saavutetaan CSS-tyylimäärittelykieltä hyödyntämällä. CSS (Cascading Style Sheets) on tekniikka, jonka avulla kuvataan sivuston ulkonäköä. CSS:n avulla annetaan verkkoselaimelle ohjeet, miten HTML-elementit tulee tyylitellä ja miltä sivuston tulee näyttää (W3Schools 2019a). HTML (Hypertext Markup Language) puolestaan on kuvauskieli, jolla kuvataan verkkosivuston sisältöä ja rakennetta (W3Schools 2019b). CSS:n avulla pystytään määrittelemään esimerkiksi elementtien värit, marginaalit, leveydet, fonttikoot ja reunukset. Lisäksi CSS sisältää responsiivisen kehityksen mahdollistavia tekniikoita, kuten media queryt sekä Flexible Box- ja Grid Layout -tekniikat.

Media queryt ovat responsiivisen kehityksen ydin. Niiden avulla pystytään tarkastelemaan laitetta, jolla sivustoa luetaan ja muokkaamaan sivuston sisältöä laitteesta riippuen. Media queryt tulivat osaksi CSS-standardia vuonna 2012. Jo aiemmin oli mahdollista kirjoittaa laitekohtaisia tyylejä ruuduille ja tulostimille, mutta esimerkiksi laitteen näytön leveyttä ei ollut mahdollista tarkastella. Media queryt tarjosivat tämän mahdollisuuden. (W3C 2012.)

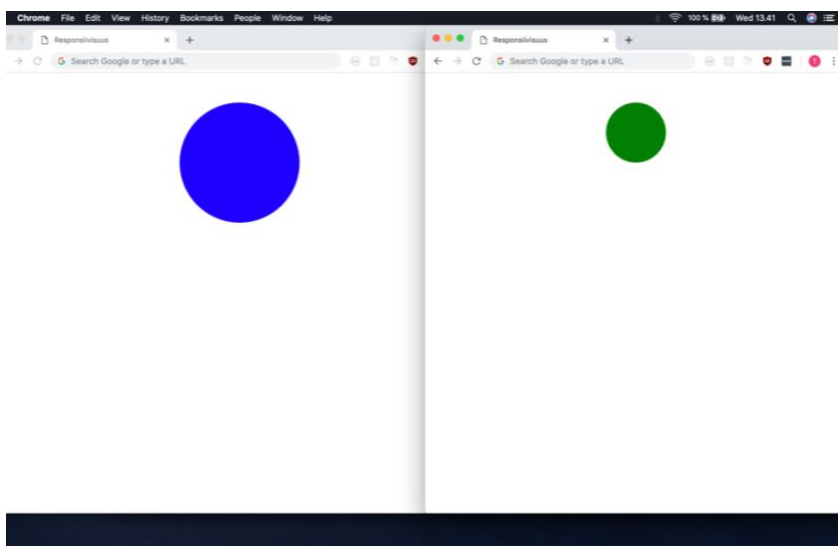
Media queryt kirjoitetaan CSS-tiedostoon ja ne perustuvat *breakpointien* eli murtumispisteiden käyttöön. Murtumispisteenä voidaan käyttää esimerkiksi sivuston

leveyttä kuvaavaa pikselimäärää. Kun murtumispiste saavutetaan, media query aktivoituu ja sen sisältämät tyylimääritykset astuvat voimaan. Tyylimäärityksiä voidaan kirjoittaa useille murtumispisteille, esimerkiksi isolle näytölle, tabletin näytölle sekä puhelimen näytölle voidaan kaikille kirjoittaa omat tyylimäärityksensä. (W3Schools 2019c.)

```
1  .pallo {
2      margin: 50px auto;
3      border-radius: 50%;
4      height: 200px;
5      width: 200px;
6      background: blue;
7  }
8
9  @media screen and (max-width: 768px) {
10     .pallo {
11         background: green;
12         height: 100px;
13         width: 100px;
14     }
15 }
```

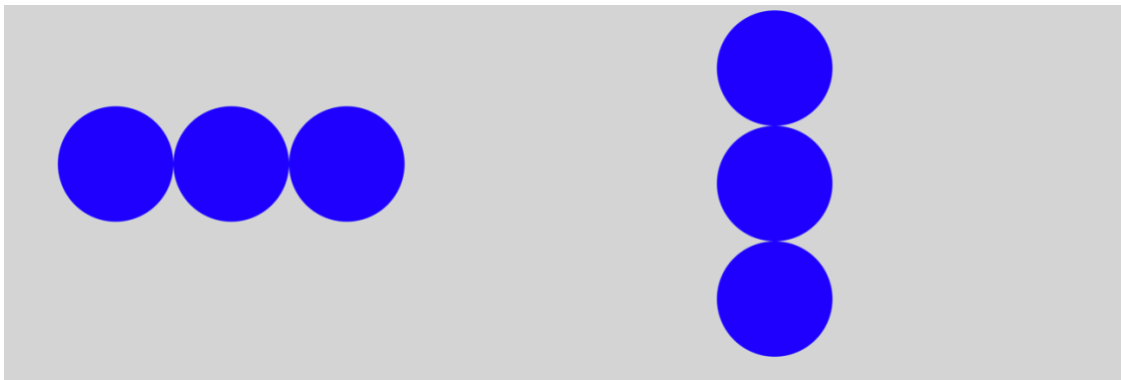
Kuva 1. Esimerkki media queryn käytöstä CSS-tiedostossa

Yllä olevassa kuvassa 1 on esitetty esimerkki media querystä osana CSS-koodia. Murtumispisteenä on käytetty 768 pikselin maksimileveyttä, jonka kohdalla media query aktivoituu ja sen tyylimääritykset tulevat voimaan. Nämä tyylimääritykset pätevät myös kaikkiin alle 768 pikselin näyttöihin. Kuvasta 2 käy ilmi kyseisen media queryn vaikutukset käytännössä. Kun näytön leveys saavuttaa 768 pikselin leveyden, pallon koko pienenee ja väri vaihtuu.



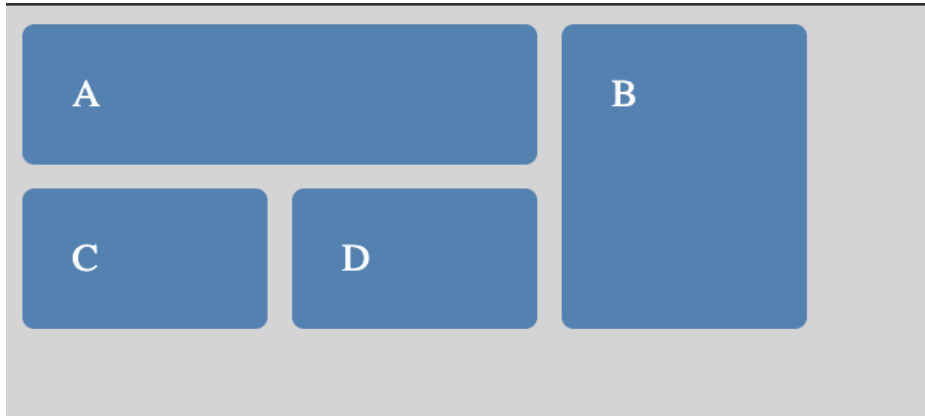
Kuva 2. Media query käytännössä

Flexible Box -tekniikka (tunnetaan myös nimellä Flexbox) mahdollistaa elementtien sijoittelun yksiulotteisesti isäntäelementin (eng. container) sisällä. Isäntäelementille määritellään ominaisuus "display: flex", jolloin sen sisältämistä elementeistä tulee niin kutsuttuja flex-elementtejä. Isäntäelementti osaa jäsentää sen sisältämät elementit tasaisesti saatavilla olevan leveyden ja korkeuden mukaisesti. Elementit jäsennellään yksiulotteisesti vaaka- tai pystyakselin mukaisesti. Kuvassa 3 on esitetty esimerkki vaaka- ja pystyakselin mukaisesta jäsentelystä. Kuvan vasemmassa reunassa olevien elementtien isäntäelementille on annettu määrittäminen "flex-direction: row", jolloin elementit asettuvat vaaka-akselin mukaisesti. Oikeassa reunassa puolestaan käytetään määrittäystä "flex-direction: column", jolloin elementit asettuvat pystyakselin mukaisesti. Responsiiviseen kehitykseen Flexbox-tekniikka soveltuu sen mukautuvuuden vuoksi. (Coyier 2013.)



Kuva 3. Flexbox-elementtien jäsentely vaaka- ja pystyakselin mukaisesti

Grid Layout -tekniikka puolestaan mahdollistaa elementtien sijoittelun kaksiulotteisesti ruudukkorakenteeseen. Isäntäelementille määritellään ominaisuus "display: grid", jonka jälkeen sen sisältämät elementit voidaan asettaa ruudukkorakenteeseen haluttuun järjestykseen vaaka- ja pystyakseleille. Kuvassa 4 on esitetty esimerkki elementtien asetelusta ruudukkorakenteeseen hyödyntäen määrittämiä "grid-column" sekä "grid-row". Näiden määrittäysten avulla elementit saadaan aseteltua ruudukkoon halutuille paikoille ja ne saadaan määritettyä halutun kokoisiksi. Kaksiulotteisuutensa ansiosta Grid Layout mahdollistaa Flexbox-tekniikkaa paremmin suurien ja monimutkaisten asetelmien toteutuksen. (House 2016.)



Kuva 4. Elementtien jäsentely Grid Layout -tekniikalla

2.2 CSS-ohjelmistokehykset

Responsiivisen kehityksen tueksi on myös vuosien saatossa kehittynyt kolmansien osapuolien CSS-ohjelmistokehyksiä. Suosittuja ohjelmistokehityksiä ovat muun muassa Bootstrap sekä Foundation. Molemmat ovat vuonna 2011 julkaistuja avoimen lähdekoodin responsiivisia CSS-ohjelmistokehyksiä. Bootstrapista on sittemmin julkaistu neljä isoa versiopäivitystä ja Foundationista kuusi. Molemmat kirjastot hyödyntävät CSS-tyylimääritelmiä sekä verkkosivustoille dynaamisuutta lisäävän ohjelmointikielen JavaScriptin funktioita, joiden tarkoituksena on helpottaa ja nopeuttaa responsiivisten, mobiilinäyttöille sopivien sivustojen kehitystä. Responsiivisuuden lisäksi kirjastot tarjoavat valmiita käyttöliittymäkomponentteja, kuten nappuloita, taulukkoja, navigointivalikoita sekä lomakkeita käyttöliittymän suunnittelun ja kehityksen tueksi. (Bootstrap 2019; Foundation 2018.)











CSS-ohjelmistokehyksiä on myös kritisoitu. Ne sisältävät valtavan määrän ohjelmakoodia, ja niiden sisällyttäminen omalle sivustolle kasvattaa sivuston kokoa huomattavasti, yleensä turhaan. Sivuston suuri koko kasvattaa latausaikoja ja heikentää sivuston suorituskykyä. Tämän vuoksi CSS-ohjelmistokehyksiä käytettäessä tulisi omalle sivustolle sisällyttää vain ne ohjelmistokehyksen määrykset, joita tarvitsee. Lisäksi ohjelmistokehyksen tyylimääryksiä joutuu usein itse ylikirjoittamaan, mikäli sivustolle kaipaa persoonallisempaa tyyliä. Ylikirjoittelut johtavat usein monimutkaisiin tyylimääryksiin ja vaikeuttavat ohjelmakoodin ylläpidettävyyttä. (Albeza 2016.)

CSS-ohjelmistokehykset tarjoavat helpotusta responsiivisuuden sekä käyttöliittymien toteutukseen, mutta ne eivät varsinaisesti tarjoa mitään, mitä itse ei voisi CSS:n avulla tehdä. Responsiivisuuden toteuttaminen Flexbox- tai Grid Layout -tekniikalla on suhteellisen helppoa, ja selaintuki näille tekniikoille alkaa olemaan hyvä. Yksittäiset käyttöliittymäelementit voidaan ohjelmoida itse tai käyttää kolmannen osapuolen, avoimen

lähdekoodin ratkaisuja, jotka eivät ole riippuvaisia valtavasta määrästä ohjelmakoodia. (Albeza 2016.)

2.3 Tuetut ominaisuudet

Responsiivisen sivuston ominaisuuksista puhuttaessa on syytä muistaa, että kyseessä on tavallinen, selaimen avulla luettava Internet-sivusto, joka mukautuu mobiililaitteille sopivaksi. Sen tarjoamat ominaisuudet ovat samat mobiililaitteella ja tietokoneella selattaessa. Selaimesta riippuen JavaScriptin avulla kuitenkin pystytään hyödyntämään erilaisia laiteominaisuuksia.

Device Features	Device Features
 NETWORK TYPE & SPEED ✓	 NETWORK TYPE & SPEED ✗
 ONLINE STATE ✓	 ONLINE STATE ✓
 VIBRATION ✓	 VIBRATION ✗
 BATTERY STATUS ✓	 BATTERY STATUS ✗
 DEVICE MEMORY ✓	 DEVICE MEMORY ✗

Kuva 5. Google Chromen (vasemmalla) ja Safarin (oikealla) tuki erilaisille laiteominaisuuksille (Bar 2019)

Kuvasta 5 nähdään, että erilaisiin laiteominaisuuksiin, kuten värinään sekä akun, verkkoyhteyden ja laitteen muistin tietoihin on mahdollista päästä käsiksi selaimen kautta. Haasteeksi kuitenkin muodostuu näiden ominaisuuksien selaintuki. Kuten kuvasta käy ilmi, Safari tukee näistä ominaisuuksista vain yhtä siinä missä Google Chrome tukee kaikkia. Tämä aiheuttaa ongelmia sivustolle, joka näitä ominaisuuksia hyödyntää, sillä se ei toimi halutun kaltaisesti kaikilla käyttäjillä. Selaintuen parantuessa näiden ominaisuuksien hyödyntäminen sivustoilla tulee kuitenkin varmasti yleistymään.

Laiteominaisuuksien tuen yleistyessä Internet-sivustot tulevat yhä enemmän muistuttamaan mobiilisovelluksia. Progressive Web App eli progressiivinen web-sovellus on termi, jolla tarkoitetaan selaimella luettavaa web-sovellusta, joka on ominaisuuksiltaan verrattavissa erilliseen mobiilisovellukseen. Progressiivisen web-sovelluksen toiminta perustuu resurssien tallentamiseen välimuistiin, jolloin sivusto toimii jopa ilman Internet-yhteyttä. Progressiiviset web-sovellukset kykenevät lähettämään käyttäjälle push-notifikaatioita, eli aloitusruudulle ilmestyviä ilmoituksia. Lisäksi sovellukset voidaan

asentaa mobiililaitteen aloitusnäytölle, jolloin ne saadaan avattua mobiilisovelluksen tavoin ilman latausaikoja. (Petrelius 2017.)

2.4 Hyödyt ja haasteet

Responsiivinen sivusto mobiiliratkaisuna on kustannustehokas. Kehitystekniikat ovat samoja tietokoneen näytölle ja mobiililaitteille kehitettäessä, eikä sivustoa tarvitse suunnitella ja kehittää eri käyttöjärjestelmille. Ylläpidettävyys on helppoa, kun ylläpidettävänä on vain yksi sivusto.

Sivustoihin on helppo julkaista päivityksiä, joiden avulla voidaan lisätä sivustoille uusia ominaisuuksia sekä yleisesti parantaa sivuston toimintaa. Käyttäjän itse ei tarvitse ladata päivityksiä, vaan sivustolla vierailtaessa sivusto on aina päivitettyä uusimpaan ja ajankohtaisimpaan versioon.

Sivuston sisällön mukauttaminen eri kokoisille näytöille voi olla suunnittelumielessä haasteellista. Pienellä näytöllä sisältöä mahtuu näytettäväksi huomattavasti tietokoneen näyttöön verrattuna vähemmän. Mobile first -suunnittelumallilla pyritään torjumaan tätä haastetta aloittamalla sivuston suunnittelu mobiiliversiosta ja siirtyen sitten isompien näyttöjen suunnitteluun. Mobile first -mallilla pystytään tarjoamaan mobiilikäyttäjille paras käyttäjäkokemus ja sivuston sisällön mukauttaminen suuremmille näytöille on helpompaa kuin isolle ruudulle mahtuvan suunnittelun mukauttaminen pienille näytöille. (ZURB 2017.)

Oman haasteensa responsiivisten sivustojen saumattomuudelle tuovat eri selaimet ja niiden tuetut ominaisuudet. HTML, CSS ja JavaScript ovat standardisoituja teknologioita, joihin aktiivisesti kehitetään uusia ominaisuuksia. Nämä ominaisuudet julkaistaan virallisten ehdotusten muodossa, jotka sisältävät tiettyjä ohjeita siitä, kuinka ne tulee toteuttaa. Selainkehittäjien vastuulla on toteuttaa tuki ehdotetuille ominaisuuksille ohjeita hyödyntäen. Tästä seuraa se, että selaimet eivät ole keskenään samanlaisia, sillä eri selainkehittäjät toteuttavat oman versionsa ominaisuuksista. Sama ominaisuus voi toimia hiukan eri tavoin kahdessa eri selaimessa, tai toisesta selaimesta voi puuttua kokonaan tuki ominaisuudelle, jota toinen selain tukee. (McPeak 2018.)

Esimerkiksi Grid Layout -ominaisuus on tuettu kokonaan tai osittain suosituimpien selainten uusimmissa versioissa, mutta vanhemmista versioista tuki ominaisuudelle puuttuu (Caniuse.com 2019). Kaikki käyttäjät eivät välttämättä käytä aina uusinta selainversiota, ja selaintuen puuttuessa sivusto ei toimi halutulla tavalla.

3 Mobiilisovellukset

Natiivilla mobiilisovelluksella tarkoitetaan tietyssä käyttöjärjestelmässä toimivaa mobiilisovellusta. Suosituimmat käyttöjärjestelmät mobiililaitteilla ovat Googlen kehittämä Android ja Applen kehittämä iOS, joiden yhteenlaskettu markkinaosuus mobiilikäyttöjärjestelmistä on yli 97% (Statcounter 2019).

Mobiilisovellukset eivät ole selainriippuvaisia, vaan ne toimivat itsenäisesti ja pääosin muista sovelluksista riippumatta. Sovellukset ladataan ja asennetaan useimmiten käyttöjärjestelmän omasta sovelluskaupasta, jotka ovat Androidilla Google Play ja iOS:llä App Store. Sovelluksia voidaan jakaa myös muilla tavoin, mutta sovelluskaupat ovat käyttöjärjestelmien viralliset jakelukanavat, joista löytyviin sovelluksiin voi luottaa. Sovelluskauppojen avulla käyttäjät voivat sovelluksia helpommin myös löytää sekä jättää niille arvosteluja. Sovelluskaupasta ladattavat sovellukset ovat joko ilmaisia tai maksullisia.

3.1 Toteutustekniikat natiivisovelluksille

Natiivin mobiilisovelluksen kehittämisessä käytetään teknologioita käyttöjärjestelmäkohtaisesti. Natiivi Android-sovellus ei toimi iOS-käyttöjärjestelmässä ja päinvastoin.

Android-sovellusten pääasiallinen kehityskieli on Java, ja toinen suosittu kehityskieli on Kotlin. Kuvassa 6 on esitelty yksinkertaisen esimerkin avulla Javan ja Kotlinin syntaksia. Android-sovellusten lähdekoodi muutetaan Android Software Development Kitin (SDK) avulla .apk-päätteiseksi tiedostoksi, jonka avulla sovellusta voidaan jakaa ja asentaa Android-laitteisiin. (Android Developers 2019a.)

```
1 public class HelloWorld {
2     public static void main(String[] args) {
3         System.out.println("Hello world!");
4     }
5 }
1 fun main(args: Array<String>) {
2     println("Hello World!");
3 }
4
5
```

Kuva 6. Yksinkertainen "Hello world" -esimerkki Javalla (vasemmalla) ja Kotlinilla (oikealla)

Android-sovellusten käyttöliittymä määritellään XML-tiedostojen avulla. XML (Extensible Markup Language) on rakenteellinen kuvauskieli, jonka avulla säilötään ja siirretään tietoa (W3Schools 2019d). Sovelluksen käyttöliittymän määrittely XML-tiedostoissa erottaa sen

ohjelmakoodista, joka hallinnoi sovelluksen logiikkaa ja parantaa ohjelmakoodin luettavuutta. Lisäksi XML-tiedostojen avulla voidaan määritellä käyttöliittymiä eri kokoisille näytöille. (Android Developers 2019b.)

Android-sovelluksia voidaan kehittää monissa eri kehitysympäristöissä. Suosittu vaihtoehto on Googlen kehittämä, Android-sovellusten virallinen kehitysympäristö Android Studio. Android Studio tarjoaa monia työkaluja kehityksen ja testauksen helpottamiseksi, kuten laite-emulaattorin, jonka avulla sovellusta voi testata emuloidulla Android-laitteella. (Android Developers 2019c.)

Applen iOS-käyttöjärjestelmälle sovellusten kehitystä varten tarvitaan Applen Mac-tietokone ja kehitysympäristöksi Applen kehittämä Xcode-kehitysympäristö. Lisäksi liittyminen Apple Developer Program -kehittäjäohjelmaan vaaditaan, jotta sovelluksia pystyy julkaisemaan App Storessa. (Grummitt 2017.)

iOS-sovellusten kehitys tapahtuu pääsääntöisesti Applen itsensä kehittämällä Swift-ohjelmointikielellä. Swift julkaistiin vuonna 2014 vaihtoehdoksi aiemmin iOS-sovellusten kehityksessä käytetylle Objective-C -ohjelmointikielelle. Swift tarjoaa parannuksia ohjelmakoodin turvallisuuteen vahvemman tyyppityksen ja virheidenhallinnan avulla. Lisäksi Apple on optimoinut Swiftin toimimaan mahdollisimman tehokkaasti ja nopeasti. iOS-sovellusten kehitys on kuitenkin edelleen mahdollista myös Objective C -ohjelmointikielellä. (Grummitt 2017.)

Kuvassa 7 on esitetty yksinkertaisen esimerkin avulla Swiftin ja Objective-C:n syntaksia. Yksinkertaisenkin esimerkin avulla nähdään, että Swiftin avulla pyritään yksinkertaisempaan syntaksiin, joka muistuttaa enemmän suosittuja ohjelmointikieliä, kuten JavaScriptiä ja Pythonia.

```
let hello = "Hello world"
print(hello)
1 NSString *hello = @"Hello World";
2 NSLog(@"%@", hello);
3
```

Kuva 7. "Hello world" -esimerkki Swiftillä (vasemmalla) ja Objective-C:llä (oikealla)

3.2 Muut toteutustekniikat

Sovelluksia on myös mahdollista kehittää web-tekniikoita (HTML, CSS, JavaScript) hyödyntäen. Vaihtoehtoisia mobiilisovellusten toteutustapoja ovat hybridisovellukset sekä Facebookin kehittämä React Native -kirjasto.

Hybridisovelluksella tarkoitetaan mobiilisovellusta, joka on ikään kuin natiivisovelluksen ja web-sovelluksen yhdistelmä. Hybridisovellukset ohjelmoidaan web-tekniikoilla, mutta perinteisen selaimen sijaan ne toimivat käyttöjärjestelmien omilla selainkomponenteilla, jotka ovat Android-laitteilla WebView ja iOS-laitteilla WKWebView. (Griffith 2019.)

Hybridisovelluksien kehittämistä varten tarvitaan ohjelmisto, joka kääntää ohjelmakoodin toimimaan mobiililaitteiden selainkomponentteja hyödyntäen. Suositettu vaihtoehto on esimerkiksi Apache Cordova. Ohjelmiston avulla päästään hyödyntämään myös erilaisia liitännäisiä, jotka tarjoavat pääsyn mobiililaitteen laiteominaisuuksiin, kuten kameraan. (Griffith 2019.)

Hybridisovellusten etu on siinä, että sama ohjelmakoodi kääntyy sekä Android- että iOS-laitteille. Yhden mobiilisovelluksen kehittäminen web-tekniikoilla on helpompaa ja nopeampaa kuin natiivisovellusten kehittäminen molemmille alustoille. Natiivisovellukset kuitenkin tarjoavat paremmin optimoidun lopputuloksen, sillä ne pääsevät hyödyntämään alustakohtaisia käyttöliittymäelementtejä, tuki laiteominaisuuksille on parempi ja natiivisovelluksissa käytetyt ohjelmointikielät ovat JavaScript-kieltä paremmin optimoituja toimimaan omilla alustoillaan. (Griffith 2019.)

React Native on Reactiin perustuva mobiilisovellusten kehitystä varten kehitetty kirjasto. React on Facebookin kehittämä ja ylläpitämä JavaScript-kirjasto, joka on tarkoitettu dynaamisten käyttöliittymien luomiseen. Reactin avulla käyttöliittymäelementit voidaan helposti jakaa erillisiksi, uudelleenkäytettäviksi komponenteiksi. React-komponenttien luomisessa käytetään yleensä JSX-syntaksia, joka muistuttaa HTML-syntaksia. (React 2019.)

React Native hyödyntää komponenttimoduuleja, jotka kääntyvät sovelluskohtaisesti niitä vastaaviksi natiivikomponenteiksi, mutta sovellusten ohjelmointi tapahtuu JavaScriptillä. Kuvassa 8 on esitetty React Nativen syntaksia, jossa käytetään <View> sekä <Text> -moduuleja tekstin tulostukseen. Hybridisovellusten tavoin React Native -sovellukset kääntyvät sekä Android- että iOS-käyttöjärjestelmille ilman tarvetta käyttöjärjestelmäkohtaisille sovelluksille. (React Native 2019a.)

```

1  class Hello extends React.Component {
2      render() {
3          return (
4              <View>
5                  <Text>Hello World!</Text>
6              </View>
7          );
8      }
9  }

```

Kuva 8. React Nativen syntaksia

Hybridisovelluksista poiketen React Nativella kirjoitetut mobiilisovellukset eivät käytä käyttöjärjestelmän selainkomponentteja sovellusten ajamiseen. Näin React Native pyrkii natiivisovellusten kaltaisesti optimoituun suorituskykyyn. Aina se ei kuitenkaan ole mahdollista, vaan React Native saattaa hiukan hävitä suorituskyvyssä natiivisovelluksille ja sovellusten optimointi saattaa vaatia enemmän työtä. (React Native 2019b.)

React Native -sovellusten käyttöliittymät voidaan luoda CSS-kielestä tutulla Flexbox-tekniikalla. Siinä missä Android- ja iOS-sovelluksia kehitettäessä käyttöliittymiä joudutaan luomaan eri kokoisille näytöille erikseen, React Nativessa Flexbox pitää huolen siitä, että käyttöliittymä on aina responsiivinen ja näytön kokoon sopeutuva. (React Native 2019c.)

Käyttöliittymien luomisen helpottamiseksi React Nativessa voidaan käyttää myös erilaisia avoimen lähdekoodin käyttöliittymäkirjastoja. Suosittuja vaihtoehtoja ovat esimerkiksi NativeBase sekä React Native Elements. Käyttöliittymäkirjastojen tarkoituksena on tarjota valmiita käyttöliittymäelementtejä ja vähentää tarvetta Flexbox-tyylimääritelmien luonnille. Ne pyrkivät myös takaamaan käyttöliittymäelementtien toimivuuden sekä Android- että iOS-käyttöjärjestelmillä ilman tarvetta käyttöjärjestelmäkohtaisille tyylimäärittelyille. (NativeBase 2019; React Native Elements 2019.)

Vaikka React Native sisältääkin suuren määrän valmiita komponenttimoduuleja, joita sovelluksessa voi käyttää, saattaa siitä joskus puuttua haluttu komponentti. React Nativessa voidaankin hyödyntää myös käyttöjärjestelmäkohtaista natiivia ohjelmakoodia esimerkiksi Javalla tai Swiftillä sen sijaan, että komponentit jouduttaisiin ohjelmoimaan JavaScriptillä itse. (React Native 2019d.)

3.3 Tuetut ominaisuudet

Mobiilisovellusten suurin etu on siinä, että ne pääsevät helposti hyödyntämään erilaisia laiteominaisuuksia, joiden avulla käyttäjäkokemusta voidaan parantaa ja sovelluksiin

saadaan laaja-alaista sisältöä. Laiteominaisuuksien käyttöä mobiilisovelluksissa ei rajoita web-sivustojen tapaan selaintuki tai muukaan tuki. Natiivisovelluksissa laiteominaisuuksien tuki on rakennettu käyttöjärjestelmään, ja hybridisovelluksissa laiteominaisuuksiin päästään käsiksi erilaisten liitännäisten avulla.

Erilaisia laiteominaisuuksia on suuri määrä. Sovellukset voivat hyödyntää esimerkiksi kameraa, mikrofonia, Bluetoothia sekä kontakti- ja sijaintitietoja. Ne voivat lähettää käyttäjän kotinäytölle ilmestyviä ilmoituksia eli push-notifikaatioita. Near Field Communication (NFC) -teknologian avulla mobiilisovelluksilla voi jopa maksaa ostoksia maksukorttien lähimaksuominaisuuden tavoin. Teknologian kehittyessä myös mobiilisovellukset kehittyvät entisestään sitä mukaan, kun uusia ominaisuuksia mobiililaitteisiin kehitetään ja käyttöjärjestelmät ne omaksuvat.

Mobiilisovellusten, erityisesti natiivisovellusten, hyvä suorituskyky mahdollistaa web-sivustoja paremmin suorituskyvyltään vaativien palvelujen kehityksen. Tämän vuoksi suosittuja mobiilisovelluksia ovatkin erilaiset mobiilipelit. Peleissä käytetään usein paljon animaatioita ja tyylliteltyjä grafiikoita, ja ne vaativat hyvän suorituskyvyn pystyäkseen tarjoamaan käyttäjälle mieluisan kokemuksen. Vaikka pelejä voidaan kehittää myös web-selaimille HTML5-teknologian avulla, on natiivisti tietyille käyttöjärjestelmälle kehitetty peli parhaiten optimoitu tarjoamaan vaaditun suorituskyvyn. (Chauhan 2018.)

3.4 Hyödyt ja haasteet

Mobiilisovellukset pystyvät tarjoamaan monipuolisen käyttäjäkokemuksen mobiilikäyttäjille. Monipuolisten laiteominaisuuksien hyödyntäminen mahdollistaa laajan kirjon toimintoja, joita sovelluksissa voi tehdä tai mihin niitä voi hyödyntää.

Sovellusten suorituskyky on usein web-sivustoja parempi, ja mobiilisovelluksien kehityksessä ei ole samankaltaisia rajoitteita kuin web-kehityksessä selaintuen aiheuttamat rajoitteet. Sovelluksia on helppo julkaista käyttöjärjestelmien sovelluskaupoissa, ja niiden kautta käyttäjät saavat helposti sovelluksia ladattua sekä myös päivitettyä. Päivitysten avulla sovellusten toimintoja voi parannella ja päivittää sekä lisätä sovellukseen uusia ominaisuuksia. Käyttäjän tulee kuitenkin itse ladata päivitykset sovelluskaupasta, ja mikäli käyttäjä ei ole sallinut automaattisia päivityksiä eikä aktiivisesti tarkasta päivityksiä itse, voi haasteeksi muodostua se, että käyttäjä käyttää vanhentunutta sovellusversiota, joka voi olla ominaisuuksiltaan hyvinkin jäljessä uusinta versiota ja jopa haavoittuvainen tietoturvaongelmille (Osakwe 2018).

Vaikka natiivisovellukset ovatkin usein suorituskyvyltään ja ominaisuuksiltaan parhaita vaihtoehtoja, muodostuu haasteeksi niiden toteutustekniikat. Kehitys tapahtuu molemmille käyttöjärjestelmille omilla ohjelmointikielillään, jotka poikkeavat toisistaan hyvinkin paljon. Optimaalisen mobiilisovelluksen kehittäminen sekä Android- että iOS-käyttöjärjestelmille vaatii kokonaan oman sovelluksen kehittämistä molemmille käyttöjärjestelmille. Toiselle käyttöjärjestelmälle kehitetty sovellus ei toimi toisessa. Ohjelmakoodia ei siis voi kierrättää, jonka vuoksi sovellusten kehitys ja ylläpito on aikaa vievää sekä kallista. (Y Media Labs 2019.)

JavaScriptiin perustuvat mobiilisovellusten toteutusvaihtoehdot pyrkivät helpottamaan kehitystä. Yhden sovelluksen kehittäminen ja yhden koodikannan kääntäminen molemmissa käyttöjärjestelmissä toimivaksi sovellukseksi on houkutteleva vaihtoehto. Lisäksi helpotusta kehitykseen tuo web-tekniikoiden käyttö, jolloin sekä web-sivusto että mobiilisovellus voidaan kehittää samoilla tekniikoilla. Haasteena hybridisovelluksissa on kuitenkin suorituskyky, sillä käyttöjärjestelmien selainkomponenteissa toimiva sovellus ei pysty tarjoamaan natiivin sovelluksen kaltaista suorituskykyä. (Y Media Labs 2019.)

4 Käyttäjätutkimus

Osana opinnäytetyötä toteutettiin käyttäjätutkimus. Tutkimuksen tavoitteena oli selvittää käyttäjien mielipiteitä ja mieltymyksiä mobiilipalveluiden käyttöön liittyen sekä tutkia suhtautumisia mobiilipalveluiden ratkaisutapoihin liittyen.

Tässä kappaleessa käydään läpi tutkimusmenetelmä sekä tutkimuksen tulokset.

4.1 Tutkimusmenetelmän kuvaus

Tutkimuksen empiirinen osa toteutettiin kvantitatiivisena eli määrällisenä kyselytutkimuksena. Määrällisen tutkimuksen tavoitteena on tuottaa luvuiksi muunnettua tietoa, jonka pohjalta tutkittavaa ilmiötä pyritään ymmärtämään. Määrällisessä kyselytutkimuksessa suositaan monivalintakysymyksiä, joiden avulla vastaukset voidaan helposti kategorisoida ja näin saadaan numeerista tietoa siitä, kuinka suuri osa vastaajista on vastannut tietyn vaihtoehdon. (McLeod 2017.)

Kysely suunnattiin mobiilipalveluiden käyttäjille. Mobiililaitteiden suosion ja arkipäiväisyyden johdosta kyselyyn oli kenen tahansa helppo vastata omakohtaisten kokemusten pohjalta. Määrällinen kyselytutkimus valikoitui menetelmäksi, koska tavoitteena oli saada ensisijaisesti luvuiksi muunnettua tietoa mobiilikäyttäjien mieltymyksistä mobiilipalveluiden käyttöön liittyen. Kyselyssä suosittiin monivalintakysymyksiä kyselyn määrällisen luonteen vuoksi, mutta kyselyssä hyödynnettiin myös avoimia kysymyksiä. Monivalintakysymysten avulla saatiin helposti numeerista tietoa tutkimalla, mitä vaihtoehtoja vastaajat ovat valinneet. Lisäksi avoimiin kysymyksiin kertyi paljon samankaltaisia vastauksia, jolloin nekin saatiin luokiteltua samankaltaisista vastauksista koostuneisiin luokkiin.

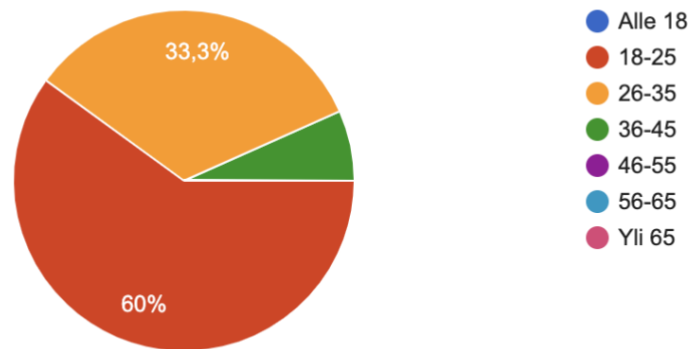
Kysely koostui kolmesta osiosta, ja tavoitteena oli pitää kysely lyhyenä, jotta siihen vastaaminen oli nopeaa ja mutkatonta. Ensimmäisessä osiossa kysyttiin taustatietoina vastaajan ikää ja ammattia tai koulutusta. Toisessa osiossa esitettiin muutamia kysymyksiä mobiilipalveluiden käyttöön liittyen. Kysymysten tavoitteena oli saada tietoa käyttäjien tottumuksista mobiilisovelluksien lataus- ja päivitysaktiivisuuteen liittyen sekä selvittää, mitä ominaisuuksia käyttäjät pitävät tärkeinä mobiilipalveluissa. Viimeisessä osiossa esitettiin esimerkkejä digitaalisista palveluista, joiden kohdalla vastaajan tuli pohtia, käyttääkö palvelua mieluummin mobiilisovelluksena vai responsiivisena sivustona. Viimeisen osion tarkoituksena oli selvittää, erottuuko toinen ratkaisutavoista selvästi

edukseen käyttäjien mielestä vai vaikuttaako palvelun tyyppi mieltymyksiin. Kyselyn kysymykset löytyvät liitteestä 1.

Kysely toteutettiin Google Forms -työkalua hyödyntäen. Vastauksia kerättiin viikon ajan aikavälillä 01.04. – 07.04.2019. Linkkiä kyselyyn jaettiin WhatsApp- ja Telegram-pikaviestintäkanavien välityksellä tuttavapiirille, josta linkki lähti leviämään eteenpäin. Kyselyyn saatiin 30 vastausta.

4.2 Tutkimuksen tulokset

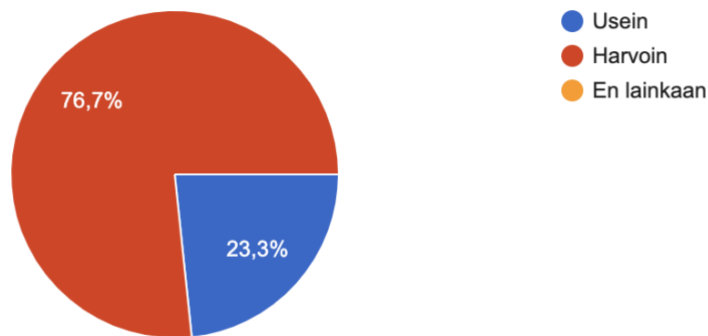
Ensimmäisenä kysymyksenä kysyttiin taustatietona vastaajan ikäluokkaa. Kyselyyn vastanneista henkilöistä suurin osa (60%) kuului ikäluokkaan 18-25. Kolmasosa vastaajista kuului ikäluokkaan 26-35, ja 2 vastaajaa kuului ikäluokkaan 36-45. (Kuvio 1.)



Kuvio 1. Vastaajien ikäjakauma

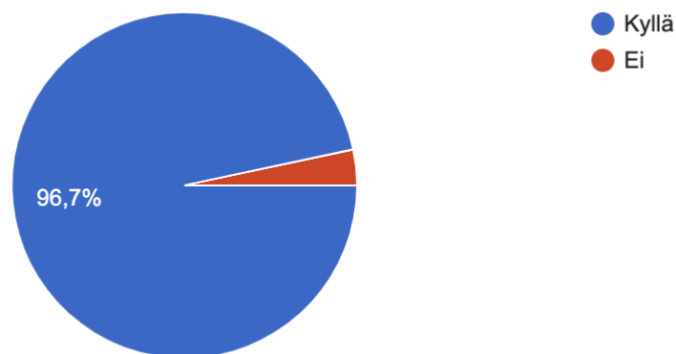
Toisena taustatietona kysyttiin vastaajan ammattia tai koulutusala. Kysymys esitettiin avoimena tekstikysymyksenä, jotta vastaajalla oli vapaus vastata kysymykseen itse haluamalla tarkkuudella. Kyselyyn vastanneiden joukossa oli sekä eri alojen opiskelijoita että ammattiin valmistuneita.

Taustatietojen jälkeen siirryttiin toiseen osioon, jossa esitettiin kysymyksiä mobiilipalveluiden käyttöön liittyen. Ensimmäisenä kysymyksenä kysyttiin mobiilisovellusten latausaktiivisuudesta. Valtaosa (77%) vastaajista lataa mobiilisovelluksia sovelluskaupasta harvoin. Tämän voi tulkita tarkoittavan käyttäjien suosivan sovellusten lataamista vain silloin, kun kokee sovellukselle olevan tarvetta. Loput vastaajista (23%) vastasivat lataavansa sovelluksia usein. (Kuvio 2.)



Kuvio 2. Mobiilisovellusten latausaktiivisuus

Yhtä vastaajaa lukuun ottamatta kaikki vastaajat vastasivat lataavansa aktiivisesti päivityksiä sovelluskaupoista asennettuihin sovelluksiin. Kysymyksessä pyydettiin vastaamaan vaihtoehto ”Kyllä”, mikäli käyttäjä on sallinut mobiililaitteen asetuksista automaattisten sovelluspäivitysten asentamisen. Automaattiset päivitykset ovat mobiililaitteissa oletusasetuksena päällä, ja vastauksista voidaan päätellä käyttäjien pitävän automaattisia päivityksiä päällä. (Kuvio 3.)



Kuvio 3. Vastausprosentit sovellusten päivittämiseen liittyen

Mobiilisovellusten asennus- ja päivitystottumuksiin liittyvien kysymysten jälkeen kysyttiin ominaisuuksista, joita pidetään mobiilipalveluissa tärkeinä. Vastauksessa sai valita vaihtoehdoista useamman sekä halutessaan kirjoittaa vapaasti myös muita ominaisuuksia, joita vaihtoehtoihin ei valmiiksi oltu listattuna. 100% vastaajista valitsi tärkeäksi ominaisuudeksi mobiilipalvelun käytettävyyden, johon kuuluvat esimerkiksi käyttöliittymän selkeys, toimivuus ja palvelun suorituskyky. Seuraavaksi eniten ääniä sai turvallisuus, jota piti tärkeänä 77% vastaajista. Offline-ominaisuutta, eli palvelun toimimista ilman Internet-yhteyttä piti tärkeänä 30% vastaajista. Laitteen kotinäytölle

ilmestyviä ilmoituksia eli push-notifikaatioita tärkeinä piti pienempi määrä, 13% vastaajista. Lisäksi yksi vastaaja ilmoitti kokevansa tärkeäksi sen, ettei palvelu saisi viedä hirveästi tilaa mobiililaitteen muistista.

Seuraavana kysymyksenä kysyttiin käyttäjien kokemia haasteita liittyen mobiililaitteilla verkkosivustojen selailuun. Kysymyksen tavoitteena oli saada käsitystä siitä, kuinka hyvin verkkosivustojen responsiivisuus ja käytettävyys mobiililaitteilla käyttäjien mielestä toteutuu. Kysymys esitettiin avoimena tekstikysymyksenä, jolloin vastaajalla oli vapaus vastata kysymykseen kuten tahtoi. Valtaosa vastaajista mainitsi haasteena verkkosivustojen skaalautuvuuden eli sisällön mukautumisen mobiililaitteille sopivaksi. Tämä kieli siitä, että verkkosivustojen responsiivisuudessa on vielä paljon puutteita. Yksittäiset palvelut voivat olla hyvinkin mobiililaitteille mukautuvia ja käyttäjäystävällisiä, mutta monilla sivustoilla on vielä ongelmia responsiivisuuden saralla.

Toinen suosittu vastaus verkkosivustojen haasteisiin liittyen oli se, että vaikka sivusto mobiililaitteille mukautuisikin, niin se karsii samalla sisältöä ja toimintoja liian rajusti johtaen käytettävyyden heikentymiseen. Esimerkkeinä mainittiin sivuston teksteihin, linkkeihin ja painikkeisiin liittyvät ongelmat, kuten liian pieni teksti ja hankalasti painettavat linkit ja painikkeet. Osa vastaajista kertoi, että jotkut sivustot karsivat kokonaisia toimintoja mobiiliversioissa tai näyttävät niin erilaiselta, että toimintoja on hankala löytää. Lisäksi kävi ilmi sivustojen hidastelua ja kankeutta mobiililaitteilla selatessa. Kaiken kaikkiaan valtaosa vastaajista vastasi kokeneensa haasteita mobiililaitteilla verkkosivustoja selatessa, ja haasteet liittyivät vahvasti juurikin käytettävyyteen. 3 vastaajaa ilmoitti, että haasteita tai toimimattomuutta ei ole ollut.

Seuraavaksi siirryttiin kyselyn viimeiseen osioon. Osiossa esitettiin esimerkkejä erilaisista digitaalisista palveluista, ja vastaajan tuli pohtia, käyttääkö palvelua mieluummin mobiilisovelluksena vai verkkosivustona mobiililaitteella. Osion tavoitteena oli selvittää, onko jompikumpi mobiiliratkaisuista selvästi enemmän käyttäjien mieleen vai vaikuttaako palvelun tyyppi mieltymyksiin. Selkeyden vuoksi vaihtoehtoissa käytettiin termiä mobiilisivusto responsiivisen sivuston sijaan, jotta vastaajan ei tarvitse tuntea responsiivisuuden käsitettä voidakseen vastata kysymykseen.

Ensimmäisenä palveluna esitettiin pikaviestintäpalvelu. Pikaviestintäpalvelulla tarkoitetaan Internet-yhteyden avulla toimivaa reaaliaikaista viestintäpalvelua, jonka kautta käyttäjät voivat lähettää tekstiviestejä sekä mediaa, kuten kuvia ja ääniä (Gil 2019). Vastaajista 100% valitsi käyttävänsä palvelua mieluummin mobiilisovelluksena (kuvio 4).



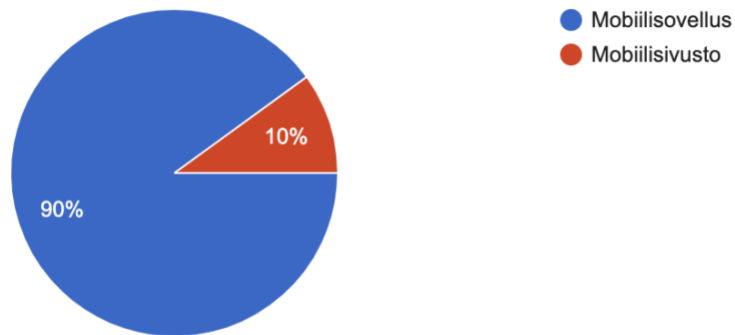
Kuvio 4. Mobiilisovellus vai mobiilisivusto: pikaviestintäpalvelu

Seuraavana esimerkkinä esitettiin esimerkiksi Spotify-musiikkipalvelun kaltainen suoratoistopalvelu, jonka avulla käyttäjä saa nauttia laajasta valikoimasta viihdettä kuukausimaksua vastaan. Vastaajista jälleen 100% valitsi käyttävänsä suoratoistopalvelua mieluiten mobiilisovelluksena (kuvio 5).



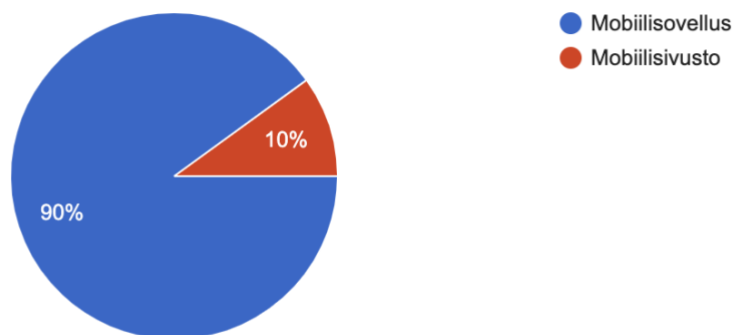
Kuvio 5. Mobiilisovellus vai mobiilisivusto: suoratoistopalvelu

Kolmantena esimerkkinä esitettiin pankkipalvelu. Pankkiasioiden hoitaminen on digitalisoitunut hurjasti ja pankkipalveluiden toimiminen ja käytettävyys mobiililaitteilla on tärkeää. Osa pankeista tarjoaakin pankkiasioiden hoitoon tarkoitettuja mobiilisovelluksia, jotka tunnetaan myös nimellä mobiilipankki. Kyselyyn vastanneista 90% suosii pankkiasioiden hoitamista mobiilisovelluksen avulla, ja 10% suosii verkkosivustoa (kuvio 6).



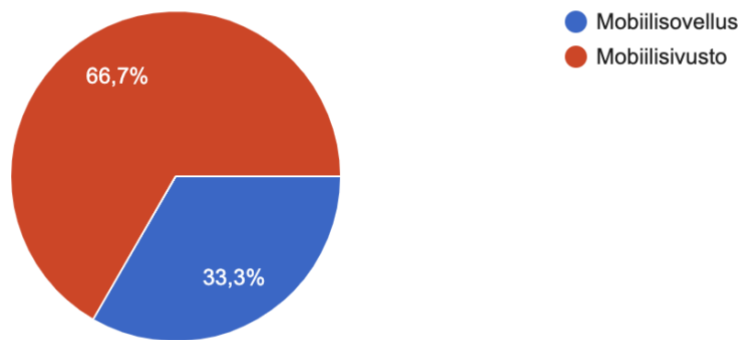
Kuvio 6. Mobiilisovellus vai mobiilisivusto: pankkipalvelu

Pankkipalvelun jälkeen esimerkkinä oli sosiaalisen median kanava. Sosiaalisen median kanavia on suuri määrä ja ne ovat muodostuneet osaksi ihmisten arkea. Suosittuja sosiaalisen median kanavia ovat muun muassa Twitter, Instagram ja Facebook. Jälleen 90% vastaajista suosii sosiaalisen median palveluja mobiilisovelluksina, ja 10% mobiilisivustoina (kuvio 7).



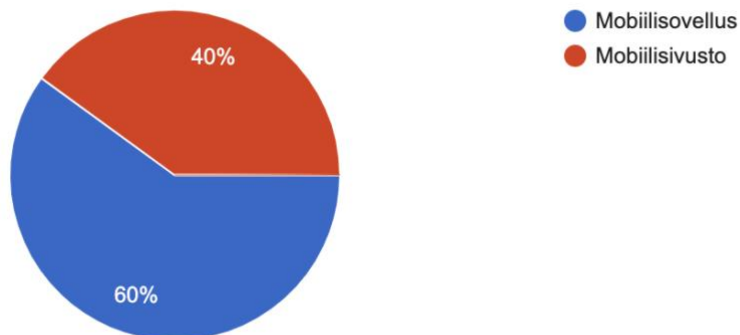
Kuvio 7. Mobiilisovellus vai mobiilisivusto: sosiaalisen median kanava

Seuraavan esimerkin kohdalla hajontaa tapahtui enemmän. Esimerkkinä oli verkkokauppa, ja 67% vastaajista valitsi käyttävänsä verkkokauppaa mieluiten mobiilisivustona. 33% vastaajista suosii verkkokauppapalvelua mobiilisovelluksena. Syynä mobiilisivustojen suosimiseen verkkokauppojen kohdalla voisi olla se, että verkkokaupat ovat usein toiminnoiltaan laaja-alaisia, siirtymisiä näkymästä toiseen tapahtuu paljon ja lisäksi tilausten tekeminen ja maksaminen saatetaan kokea turvallisemmaksi verkkosivuston kautta. (Kuvio 8.)



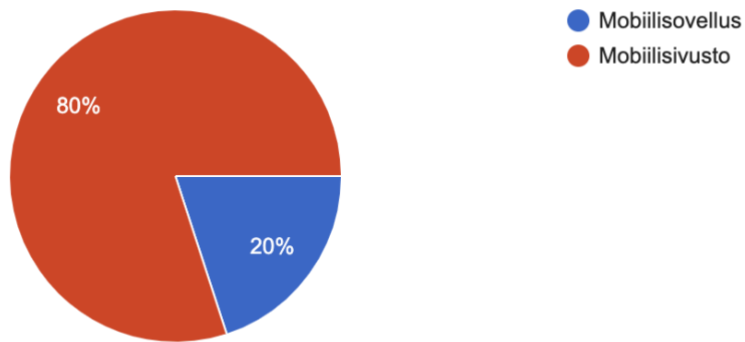
Kuvio 8. Mobiilisovellus vai mobiilisivusto: verkkokauppa

Seuraavana esimerkkinä esitettiin uutispalvelu. Suositut uutispalvelut, kuten Helsingin Sanomat, Yle ja Iltalehti tarjoavat omat mobiilisovelluksensa, mutta ovat myös panostaneet verkkosivustojen responsiivisuuteen. Uutispalvelu on palveluna melko kevyt, keskittyen pääsääntöisesti tekstiin ja kuviin, välillä myös videoihin. Kyselyyn vastanneista 60% suosii uutispalveluiden käyttöä mobiilisovelluksen avulla, ja 40% mobiilisivuston kautta (kuvio 9).



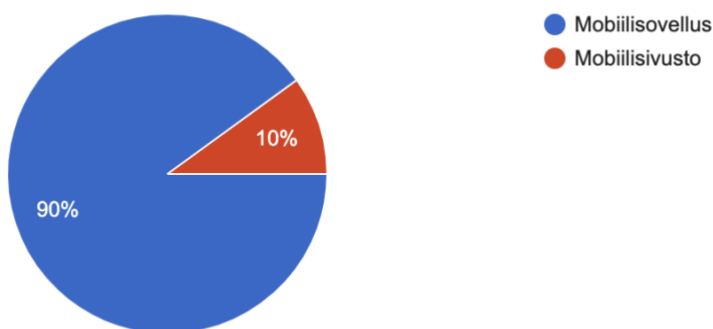
Kuvio 9. Mobiilisovellus vai mobiilisivusto: uutispalvelu

Uutispalvelun jälkeen esimerkkinä esitettiin keskustelufoorumi. Keskustelufoorumit ovat esimerkiksi tiettyyn aihealueeseen keskittyviä keskustelukanavia, joissa käyttäjät voivat keskustella anonyymisti tai nimimerkkejä hyödyntäen. 80% kyselyyn vastanneista suosivat keskustelufoorumeja mobiilisivustoina ja 20% mobiilisovelluksena (kuvio 10). Keskustelufoorumit ovat perinteisesti melko yksinkertaisia palveluita, joiden käyttäminen responsiivisena sivustona on sujuvaa, ja tästä syystä suurin osa käyttäjistä ei välttämättä koe tarvetta erilliselle mobiilisovellukselle.



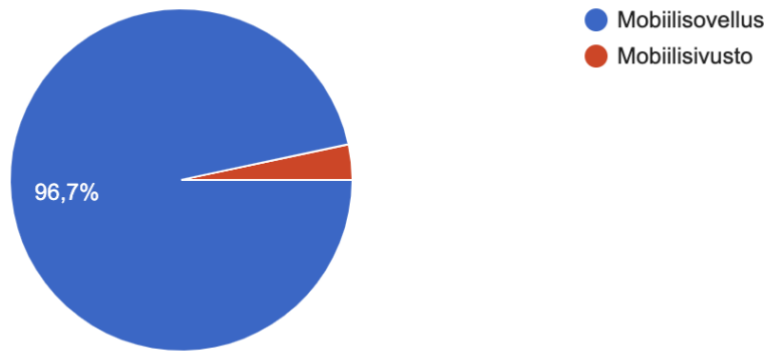
Kuvio 10. Mobiilisovellus vai mobiilisivusto: keskustelufoorumi

Toiseksi viimeisenä esimerkkinä oli karttapalvelu. Karttapalvelun avulla käyttäjä voi tutkia karttaa, saada tietoa omasta sijainnistaan sekä lähellä olevista palveluista ja navigoida paikasta toiseen. Vastaajista 90% suosii karttapalveluita mobiilisovelluksena ja 10% sivustona (kuvio 11).



Kuvio 11. Mobiilisovellus vai mobiilisivusto: karttapalvelu

Viimeisenä esimerkkinä esitettiin mobiililaitteella pelattavat pelit. Pelit ovat usein melko raskaita ja hyvää suorituskykyä vaativia palveluita, ja ne ovatkin usein suosituimpien mobiilisovellusten joukossa. 97% vastaajista suosikin pelejä mobiilisovelluksina, ja vain 1 vastaaja vastasi suosivansa pelejä mobiilisivuston kautta (kuvio 12).



Kuvio 12. Mobiilisovellus vai mobiilisivusto: pelit

5 Johtopäätökset ja pohdinta

Tässä kappaleessa peilataan tutkimuksen tuloksia teoriataustaan ja käydään läpi johtopäätöksiä sekä pohditaan tutkimuksen luotettavuutta, jatkokehitysmahdollisuuksia sekä opinnäytetyöprosessia.

5.1 Johtopäätökset

Tutkimuksen tavoitteena oli selvittää, miten responsiivinen sivusto ja erillinen applikaatio soveltuvat omalta osaltaan mobiiliratkaisuiksi sekä selvittää, suosivatko käyttäjät selvästi jompaakumpaa menetelmää digitaalisten palveluiden käytössä. Teoriaosuuden kautta tutustuttiin molempiin ratkaisutapoihin toteutustekniikoiden osalta, tuettujen ominaisuuksien kautta sekä pohdittiin hyötyjä ja haasteita. Kyselytutkimuksen avulla kartoitettiin käyttäjien mielipiteitä mobiiliratkaisuista sekä niihin liittyvistä mieltymyksistä. Ratkaisutavat eroavat toisistaan suuresti, mutta yhtäläisyyksiäkin löytyy.

Molemmat ratkaisutavat soveltuvat mobiiliratkaisuiksi digitaalisille palveluille. Modernit verkkoselaimet tukevat yhä enenevässä määrin tekniikoita, joiden avulla responsiivisten, mobiililaitteen ominaisuuksia hyödyntävien sivustojen kehittäminen onnistuu mutkattomasti. Tutkimuksessa kävi kuitenkin ilmi, että responsiivisiin sivustoihin liittyy suuri määrä haasteita. Suurimmat haasteet liittyvät juurikin responsiivisuuteen itsessään, eli sivuston sisällön mukautumiseen mobiililaitteen näytölle sopivaksi. Toimintorikkaan sivuston mukauttaminen pienelle näytölle on haastavaa, mutta toisaalta myös toimintojen rajoittaminen ja piilottaminen mobiilinäkymässä on turhauttavaa. Tätä turhautumista saataisiin torjuttua suunnittelemalla verkkopalvelun toiminnot mobile first -suunnittelumallilla. Sen avulla palvelun toiminnot ja käyttäjäkokemus saataisiin huomioitua riittävän laajasti mobiililaitteilla käyttämisen osalta.

Kyselytutkimuksen tulokset osoittavat, että mobiilisovellukset ovat käyttäjien suosima tapa käyttää digitaalisia palveluja. Kyselyssä esitetyistä esimerkkipalveluista valtaosan kohdalla mobiilisovellukset osoittautuivat käyttäjien suosimaksi käyttötavaksi. Mielipiteitä enemmän jakaneet esimerkkipalvelut olivat verkkokauppa, keskustelufoorumi ja uutispalvelu, joista kahta ensimmäistä käytetään mieluiten responsiivisena verkkosivustona.

Mobiilisovelluksia on saatavilla tarpeeseen kuin tarpeeseen, niin viihteeseen kuin erilaisten asioiden hoitamiseen tarkoitettuja. Sovellukset kykenevät tarjoamaan responsiivista sivustoa paremmin parhaan mahdollisen käyttäjäkokemuksen

mobiililaitteilla, sillä ne ovat alusta alkaen suunniteltu nimenomaan mobiilikäyttäjille. Mobiilisovelluksen toimintoja ja käyttöliittymää ei tarvitse mukauttaa suurelle määrälle eri kokoisia näyttöjä, ja näin sovellukset saadaan suunniteltua juurikin mobiililaitteiden näyttökokoon sopivaksi. Lisäksi laiteominaisuuksien laaja kirjo, johon mobiilisovellukset helposti pääsevät käsiksi, parantaa ja näppäröittää käyttäjäkokemusta entisestään.

Digitaalisen palvelun tyyppin tai toimialan vaikutus mobiiliratkaisun valintaan oli yksi tutkimuskysymyksistä, ja tuloksista käy ilmi, että jonkin verran vaikutusta on. Käyttäjät eivät koe halua tai tarvetta ladata erillistä mobiilisovellusta kaikkien asioiden hoitamista varten, vaan suosivat välillä myös verkkosivustoja. Verkkokaupat, uutispalvelut ja keskustelufoorumit usein huomioivat mobiilikäyttäjät hyvin ja niiden kehityksessä on kiinnitetty huomiota responsiivisuuteen ja toimivuuteen pienelläkin näytöllä käytettäessä. Mikäli käyttäjät ovat tyytyväisiä palvelun toimivuuteen ja käytettävyyteen verkkosivustona käytettäessä, ei tarvetta erilliselle mobiilisovellukselle välttämättä koeta.

Käytettävyys on mobiilipalvelun tärkein ominaisuus. Palvelun käyttöliittymän selkeys, toimintojen löydettävyys, nopea suorituskyky ja viiveetön reagointi käyttäjän painalluksiin vaikuttavat kaikki käytettävyyteen. Heikko käytettävyys johtaa käyttäjän turhautumiseen ja saa tämän hakemaan ratkaisua toisesta palvelusta. Mobiilisovelluksissa hyvä käytettävyys toteutuu juurikin edellä mainituista syistä – mobiilisovellukset ovat kehitetty varta vasten mobiililaitteille käyttöliittymän suunnittelusta alkaen aina käyttöjärjestelmäkohtaisesti optimoituun suorituskykyyn asti. Mobiilisovellusten kehitys vaatii hieman enemmän työtä ja niiden toteutustekniikat, erityisesti natiivisovellusten, toteutustekniikat vaativat erilaista ja laajempaa osaamista responsiivisten sivustojen kehitykseen verrattuna, mutta niihin käytetty aika ja vaiva palkitaan paremmalla käytettävyydellä, käyttäjäkokemuksella ja käyttäjätyytyväisyydellä.

On kuitenkin myös syytä pohtia mobiilipalveluiden tulevaisuutta. Progressiiviset web-sovellukset kykenevät jo nyt tarjoamaan mobiilisovelluksen kaltaisia käyttäjäkokemuksia. Nopeat latausajat, offline-toiminnallisuudet, push-ilmoitukset sekä mahdollisuus lisätä palvelu mobiililaitteen kotinäytölle ovat kaikki mobiilisovelluksille tyypillisiä ominaisuuksia, joita myös progressiiviset web-sovellukset hyödyntävät. Voidaan siis sanoa, että progressiiviset web-sovellukset pystyvät tarjoamaan mobiilisovellusten kaltaisen käytettävyyden ilman tarvetta käyttöjärjestelmäkohtaisille versioille. Modernit verkkoselaimet tukevat jo nyt laajasti niiden kehityksen mahdollistavia ominaisuuksia, ja tuki on hyvää vauhtia yleistymässä. Progressiivisten web-sovellusten kehityksen yleistyessä ja niihin kohdistuvan yleisen tietämyksen laajentuessa niistä on

mobiilisovellusten haastajaksi. Tällä hetkellä mobiilisovellus on paras vaihtoehto mobiiliratkaisuksi, mutta muutaman vuoden kuluttua tilanne voi olla jo toinen.

5.2 Pohdinta

Tutkimus toteutettiin määrällisenä kyselytutkimuksena, jonka tavoitteena oli tuottaa ensisijaisesti numeerista tietoa käyttäjien kokemuksista ja mielipiteistä mobiiliratkaisuihin liittyen. Pyrin pitämään kyselyn yksinkertaisena ja lyhyenä, jotta siihen vastaaminen olisi mahdollisimman mutkatonta ja helppoa. Näiden asioiden osalta tutkimus mielestäni onnistui ja sain tuloksista irti haluttua tietoa.

Tutkimuksen luotettavuutta pohtiessa on kuitenkin syytä huomioida kyselyyn saatujen vastausten suhteellisen pieni määrä. Määrällinen tutkimus olisi hyötynyt suuremmasta otantamäärästä tutkimuksen tuloksia arvioitaessa ja johtopäätöksiä tehdessä. Tutkimuksen suunnitteluun ja vastausten keräämiseen olisi voinut käyttää enemmän aikaa ja sitä olisi voinut jakaa laaja-alaisemmin vastaajien tavoittamiseksi.

Tutkimuksen pienestä otantamäärästä huolimatta pidän tutkimuksen tuloksia ja johtopäätöksiä melko luotettavina. Mobiilisovellukset mahdollistavat toimivan ja sujuvan käyttäjäkokemuksen tällä hetkellä responsiivisia sivustoja paremmin ja se näkyy käyttäjien mieltymyksissä. En usko, että suuremmalla otantamäärällä tulokset ja johtopäätökset olisivat olleet merkittävästi erilaisia.

Jatkokehitys ja -tutkimusideoita pohdittaessa voidaan nostaa esiin progressiiviset web-sovellukset ja niihin liittyvät tietämykset ja mielipiteet. Itse koen progressiivisten web-sovellusten nousevan muutaman vuoden kuluessa mobiilisovellusten haastajaksi, kun niihin liittyvä tietämys paranee ja niiden kehitys yleistyy. Olisi mielenkiintoista tutkia esimerkiksi sitä, kuinka laajasti progressiivisten web-sovellukselle tyypillisiä ominaisuuksia hyödynnetään verkkopalveluiden kehityksessä tällä hetkellä ja huomioidaanko progressiiviset web-sovellukset varteenotettavana vaihtoehtona mobiilisovelluksille.

Opinnäytetyöprosessi on opettanut itselleni erityisesti oman ajan hallintaa sekä itsenäisen, pitkäkestoisemman projektin työstämiseen liittyviä taitoja. Huolellinen suunnittelu ja opinnäytetyöhön liittyvien tehtävien jakaminen viikkotasoihin kokonaisuuksiin auttoi selkeyttämään kokonaiskuvaa ja auttoi hahmottamaan projektin aikataulua ja myös pysymään siinä.

Työtä tehdessä omakohtainen kokemus ja tietämys responsiivisten sivustojen sekä mobiilisovellusten kehittämisestä osoittautui avuksi. Teoriaosuutta kirjoittaessa tiesin usein, mitä haluan kirjoittaa. Työ kuitenkin opetti uutta erityisesti mobiilisovelluksiin liittyvistä asioista. Lähteitä aiheesta löytyy suuri määrä, joista pyrin poimimaan itseni mielestä luotettavat ja ajan tasalla olevat.

Kaiken kaikkiaan koen opinnäytetyöprosessin onnistuneeksi. Työ eteni aikataulussa suunnitellun mukaisesti ja työn tavoitteisiin päästiin. Ammatillinen osaamiseni ja ymmärrykseni mobiiliratkaisuista kasvoi työtä tehdessä, eikä suurempia ongelmia työn tekemisessä esiintynyt.

Lähteet

Albeza, B. 2016. You might not need a CSS framework. Luettavissa: <https://hacks.mozilla.org/2016/04/you-might-not-need-a-css-framework/>. Luettu: 04.04.2019.

Android Developers 2019a. Application Fundamentals. Luettavissa: <https://developer.android.com/guide/components/fundamentals>. Luettu: 13.03.2019.

Android Developers 2019b. Layouts. Luettavissa: <https://developer.android.com/guide/topics/ui/declaring-layout>. Luettu: 13.03.2019.

Android Developers 2019c. Meet Android Studio. Luettavissa: <https://developer.android.com/studio/intro>. Luettu: 14.03.2019.

Bar, A. 2019. What Web Can Do Today. Luettavissa: <https://whatwebcando.today/>. Luettu: 08.03.2019.

Bootstrap 2019. About. Luettavissa: <https://getbootstrap.com/docs/4.3/about/overview/>. Luettu: 07.03.2019.

Caniuse.com 2019. CSS Grid Layout (Level 1). Luettavissa: <https://caniuse.com/#search=grid>. Luettu: 11.03.2019.

Chauhan, A. 2018. Browser-Based Gaming Vs Native Apps: What's Your Preference? Luettavissa: <https://www.mobileappdaily.com/2018/01/31/browser-based-gaming-vs-native-apps>. Luettu: 20.03.2019.

Coyier, C. 2013. A Complete Guide to Flexbox. Luettavissa: <https://css-tricks.com/snippets/css/a-guide-to-flexbox/>. Luettu: 06.03.2019.

Enge, E. 2018. Mobile vs Desktop Usage in 2018: Mobile takes the lead. Luettavissa: <https://www.stonetemple.com/mobile-vs-desktop-usage-study/>. Luettu: 01.03.2019.

Foundation 2018. A Yeti's Trek. Luettavissa: <https://foundation.zurb.com/showcase/about.html>. Luettu: 07.03.2019.

Gil, P. 2019. The Full Rundown of Instant Messaging or 'I.M.'. Luettavissa: <https://www.lifewire.com/what-is-instant-messaging-2483319>. Luettu: 09.04.2019.

Grauschopf, S. 2019. Internet Browsers: A Layman's Guide to How They Work. Luettavissa: <https://www.thebalanceeveryday.com/what-is-internet-browser-892819>. Luettu: 05.04.2019.

Griffith, C. 2019. What is Hybrid App Development? Luettavissa: <https://ionicframework.com/resources/articles/what-is-hybrid-app-development>. Luettu: 15.03.2019.

Grummitt, C. 2017. iOS Development with Swift. Manning Publications. New York.

House, C. 2016. A Complete Guide to Grid. Luettavissa: <https://css-tricks.com/snippets/css/complete-guide-grid/>. Luettu: 07.03.2019.

Marcotte, E. 2010. Responsive Web Design. Luettavissa: <https://alistapart.com/article/responsive-web-design>. Luettu: 04.03.2019.

McLeod, S. 2017. What's the difference between qualitative and quantitative research? Luettavissa: <https://www.simplypsychology.org/qualitative-quantitative.html>. Luettu: 04.04.2019.

McPeak, A. 2018. A Brief History of Web Browsers and How They Work. Luettavissa: <https://crossbrowstesting.com/blog/test-automation/history-of-web-browsers/>. Luettu: 11.03.2019.

NativeBase 2019. NativeBase. Luettavissa: <https://nativebase.io/>. Luettu: 05.04.2019.

Osakwe, M. 2018. Why It's Important to Update Your Apps, Software and Devices. Luettavissa: <https://www.nextadvisor.com/blog/why-its-important-to-update-your-apps-software-and-devices/>. Luettu: 05.04.2019.

Petreljus, M. 2017. PWA: Suuri harppaus web-sovelluksille. Luettavissa: <https://koodiystava.fi/pwa-suuri-harppaus-web-sovelluksille-170d35bb8d7e>. Luettu: 10.03.2019.

React 2019. React – A JavaScript library for building user interfaces. Luettavissa: <https://reactjs.org/>. Luettu: 16.03.2019.

React Native 2019a. React Native – Build native mobile apps using JavaScript and React. Luettavissa: <https://facebook.github.io/react-native/>. Luettu: 16.03.2019.

React Native 2019b. Performance. Luettavissa: <https://facebook.github.io/react-native/docs/performance>. Luettu: 19.03.2019.

React Native 2019c. Layout with Flexbox. Luettavissa: <https://facebook.github.io/react-native/docs/flexbox>. Luettu: 19.03.2019.

React Native 2019d. Native Modules. Luettavissa: <https://facebook.github.io/react-native/docs/native-modules-android>. Luettu: 05.04.2019.

React Native Elements 2019. Getting Started. Luettavissa: https://react-native-training.github.io/react-native-elements/docs/getting_started.html. Luettu: 05.04.2019.

Smashing Magazine 2011. Responsive Web Design – What It Is And How To Use It. Luettavissa: <https://www.smashingmagazine.com/2011/01/guidelines-for-responsive-web-design/>. Luettu: 04.03.2019.

Statcounter 2019. Mobile Operating System Market Share Worldwide Feb 2018 – Feb 2019. Luettavissa: <http://gs.statcounter.com/os-market-share/mobile/worldwide>. Luettu: 13.03.2019.

Usability.gov 2019. User Interface Elements. Luettavissa: <https://www.usability.gov/how-to-and-tools/methods/user-interface-elements.html>. Luettu: 05.04.2019.

W3Schools 2019a. CSS Introduction. Luettavissa: <https://www.w3schools.com/css/default.asp>. Luettu: 05.04.2019.

W3Schools 2019b. HTML Introduction. Luettavissa: https://www.w3schools.com/html/html_intro.asp. Luettu: 05.04.2019.

W3Schools 2019c. Responsive Web Design – Media Queries. Luettavissa: https://www.w3schools.com/css/css_rwd_mediaqueries.asp. Luettu: 05.03.2019.

W3Schools 2019d. XML Introduction. Luettavissa:

https://www.w3schools.com/xml/xml_what_is.asp. Luettu: 05.04.2019.

W3C 2012. Media Queries. Luettavissa: <https://www.w3.org/TR/2012/REC-css3-mediaqueries-20120619/>. Luettu: 06.03.2019.

Y Media Labs 2019. Hybrid vs Native Mobile Apps – The Answer is Clear. Luettavissa: <https://ymedialabs.com/hybrid-vs-native-mobile-apps-the-answer-is-clear>. Luettu: 16.04.2019.

ZURB 2017. Mobile First. Luettavissa: <https://zurb.com/word/mobile-first>. Luettu: 11.03.2019.

Liitteet

Liite 1. Kyselytutkimus

Taustatiedot

Ikä *

Alle 18

18-25

26-35

36-45

46-55

56-65

Yli 65

Ammatti tai koulutusala *

Oma vastauksesi _____

Kysymyksiä

Kuinka usein lataat mobiilisovelluksia sovelluskaupasta? *

Usein

Harvoin

En lainkaan

Lataatko päivityksiä asentamiisi sovelluksiin? (Valitse vaihtoehto "Kyllä", jos käytät automaattisia päivityksiä) *

Kyllä

Ei

Mitä seuraavista ominaisuuksista pidät tärkeinä mobiilipalveluissa? *

Käytettävyys (käyttöliittymän selkeys, toimivuus, suorituskyky)

Offline-ominaisuudet (toimivuus ilman verkkoyhteyttä)

Push-ilmoitukset (laitteen kotinäytölle ilmestyvät ilmoitukset)

Turvallisuus

Muu: _____

Oletko kohdannut verkkosivustoilla haasteita tai toimimattomuutta mobiililaitteella selatessasi? Jos olet, niin minkälaisia? *

Oma vastauksesi _____

Valinnat

Osiossa on esitetty esimerkkejä eri tyyppisistä digitaalisista palveluista. Pohdi, kummalla tavalla käyttäisit palvelua mieluiten (mobiilisivusto vai mobiilisovellus).

Pikaviestintäpalvelu (esim. WhatsApp) *

- Mobiilisovellus
- Mobiilisivusto

Suoratoistopalvelu (esim. Spotify) *

- Mobiilisovellus
- Mobiilisivusto

Pankkipalvelu *

- Mobiilisovellus
- Mobiilisivusto

Sosiaalisen median kanava (esim. Twitter) *

- Mobiilisovellus
- Mobiilisivusto

Verkkokauppa *

- Mobiilisovellus
- Mobiilisivusto

Uutispalvelu *

- Mobiilisovellus
- Mobiilisivusto

Keskustelufoorumi *

- Mobiilisovellus
- Mobiilisivusto

Karttapalvelu *

- Mobiilisovellus
- Mobiilisivusto

Pelit *

- Mobiilisovellus
- Mobiilisivusto