

Aksiospro:n mobiilinäkymän kehitys ja responsiivisen suunnittelun tärkeys nykyaikana

Tatu Korpela



Tekijä(t) Tatu Korpela	
Koulutusohjelma Tietojenkäsittelyn koulutus, Ohjelmistotuotanto profiili	
Raportin/Opinnäytetyön nimi Aksiospro:n mobiilinäkymän kehitys	Sivu- ja liitesivumäärä 21
<p>Pääaihe on asiakasyrityksen Aksioksen ohjelmiston Aksios Pro:n mobiilipuolen kehittäminen ja parantaminen, mikä eteni asteittain ja sivu kerrallaan. Ideana oli saada näkymistä symmetrisiä ja keskelle painotettuja, jotta käyttökokemus älypuhelimilla parantuisi. Koska monella elementillä ja osalla oli koodiin kirjattuja tyylejä, selkeämpien ja helpommin hallittavien tyyliluokkien sijaan, oli ohjelmiston ulkoasun kehittäminen hidasta.</p> <p>Työssä aluksi käydään läpi ohjelmistojen suunnittelua mobiilialustoille. Responsiivisella suunnittelulla tarkoitetaan, kuinka ohjelmisto tai verkkosivu suunnitellaan ja toteutetaan eri kokoisille näkymille. Kuinka näkymän tai selaimen eri osat muuttuvat leveyden vaihtuessa.</p> <p>Mobile First suunnittelussa ohjelmiston tai verkkosivun osat ja käyttäjäkokemus suunnitellaan ensisijaisesti sopimaan pienemmille mobiililaitteiden näytöille, ja tätä näkymää käytetään pohjana, kun toiminnallisuutta ja käytettävyyttä laajennetaan toimimaan paremmin suuremmille tietokoneen näytöille.</p>	
Asiasanat Mobile first - Ohjelmiston suunnittelu ensisijaisesti Mobiilialustoille, kuten älypuhelimille. Responsiivisuus - Ohjelmiston osat muuttuvat näkymän koon muuttuessa. Css - tyylimäärittely tiedosto, sisältää määriteltyjä tyylejä Elementti - Ohjelmisto koostuu useista elementeistä, minkä sisällöt ja koot vaihtelevat	

Sisällys

1	Johdanto	1
2	Responsiivinen suunnittelu ja mobile first	2
2.1	Responsiivinen suunnittelu.....	2
2.2	Progressive Advancement ja Graceful Degradation	4
2.3	Mobiilille suunnittelu on erilaista	5
2.4	Haasteet mobiilille suunnittelussa	8
2.5	Haasteet mobiilin käytössä	11
2.6	Miksi responsiivinen suunnittelu ja mobile first ovat nykyään tärkeitä.....	12
2.7	Responsiivisen suunnittelun heikkouksia	14
3	Tuotokset	15
3.1	Aksios Corporationin ohjelmiston responsiivisuuden kehitys.....	15
3.2	Sivujen samankaltaisuuksien korjauksia	17
3.3	Työajanhallinnan-sivun vaatimat muutokset.....	19
3.4	Kalenterin rakenne ja toiminnot mobiilinäkymälle	19
4	Pohdinta.....	20
	Lähteet	22

1 Johdanto

Projektin tarkoituksena on kehittää vanhankoa ja mobiililaitteille huonosti skaalautuvaa ohjelmistoa, täten tuoden ohjelmisto lähemmäs nykyajan mobiiliympäristön mittoja, ja parantaen käyttäjäkokemusta ja toimivuutta mobiililaitteilla. Projektissa selainpohjainen ohjelmisto päivitetään skaalautumaan erikokoisten mobiililaitteiden näytöille.

Ohjelmistoon on aikaisemmin rakennettu ensisijaisesti mobiilille tarkoitettuja sivuja, mitkä ovat käytössä, kun käyttäjä kirjautuu ohjelmistoon sisään mobiililaitteella. Kyseessä ei ole automaattisesti skaalautuvia sivuja, vaan erilliset mobiilille tarkoitettut sivut.

Ohjelmiston kehitystyöllä ja asiakastyytyväisyyden parantumisella olemassa olevien asiakkaitten kohdalla varmistetaan heidän tyytyväisyys ja sitoutuminen käyttämään ohjelmistoa jatkossakin. Lisäksi ohjelmiston ulkonäön responsiivisuuden kehityksellä mahdollistetaan parempi markkinointi uusille asiakkaille mahdollisuudella vedota ohjelmiston käyttömahdollisuuksiin mobiililaitteilla.

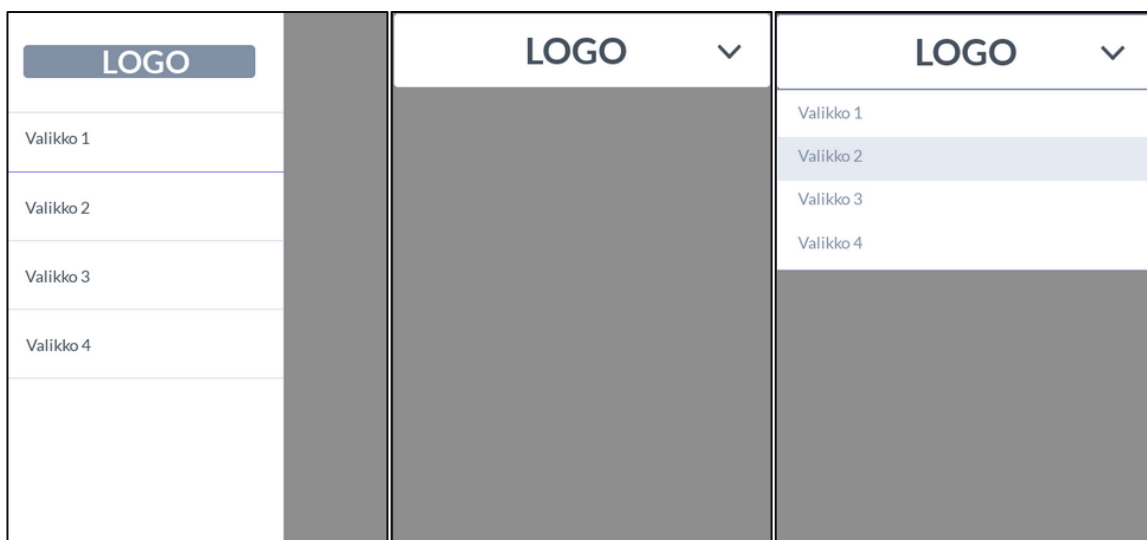
Opinnäytetyössä tullaan käymään läpi Responsiivista ja Mobile First - suunnittelua, sekä miksi nykyään on kannattavaa rakentaa verkkosivut responsiivisiksi.

2 Responsiivinen suunnittelu ja mobile first

Tässä kappaleessa selitän tarkemmin mitä responsiivisella suunnittelulla ja mobile first termeillä tarkoitetaan, ja mitä vaikutuksia niillä on käytännössä niin ohjelmoijille, kuin lopputähtäjille. Kappaleessa käydään myös läpi responsiivisen suunnittelun tuottamia haasteita.

2.1 Responsiivinen suunnittelu

Mitä tarkoitetaan, tai mistä puhutaan, kun mainitaan termi Responsiivinen suunnittelu? Kyseisellä termillä tarkoitetaan sitä, kuinka verkkosivut suunnitellaan ja toteutetaan, niin että verkkosivut skaalautuvat automaattisesti eri kokoisille ruuduille. Responsiivisen suunnittelun tarkoituksena on saada verkkosivu näyttämään hyvältä käyttäjälle riippumatta, katso taanko verkkosivua älypuhelimien pienestä näytöstä, tablettitietokoneen ruudulta, tai pöytä tietokoneen näytöltä. Responsiivisella suunnittelulla pyritään saamaan kaikki näkymät yhtä käyttökelpoisiksi. Verkkosivun eri osat voivat vaihtaa paikkaa, tekstin ja kuvien koot pienenevät tai kasvavat, sivulta on jopa mahdollista piilottaa kokonaisia ei tärkeitä osia. Sivulta selainikkunan koon muuttuessa valikon avoin tyyli (kuva 1) voi muuttua pudotusvalikoksi, eli valikko muuttuu niin, että siitä on näkyvillä vain pieni kuva tai lyhyt otsikko (kuva 2), ja sen aktivoituessa, kuten esimerkiksi valikkoa painetaan, valikko laajenee ja näyttää useampia valittavia osia (kuva 3). Responsiivisella suunnittelulla mahdollistetaan verkkosivun automaattinen mukautuminen eri kokoisille näytöille, ja jopa selaimen koon muuttuessa (Kim Bohyun. 2013).



Kuva 1. Avoin valikko.

Kuva 2. Pudotusvalikko kiinni. Kuva 3. Pudotusvalikko auki

Toimiakseen kunnolla, responsiivisessa suunnittelussa sivun eri elementit, kuten tekstit, kuvat ja painikkeet, asetetaan joustavalle ruudukolle (kuva 4), termiltään ”Flexible Grid”, mikä muuttuu riippuen näkymän koosta (Nielsen & Budiu 2013, 28).

Yksi	Kaksi	Kolme
Neljä	Viisi	Kuusi
Seitsemän	Kahdeksan	Yhdenksän

Kuva 4. Kolmeen riviin ja kolmeen sarakkeeseen jaettu joustava ruudukko

Joustavan ruudukon sarakkeet automaattisesti mitoittaa oman leveytensä vapaana olevan tilan mukaan, eli normaalisti joustavan ruudukon leveys on sama kuin selainikkunan leveys, ja se joustaa automaattisesti selainikkunan leveyden muuttuessa. Ruudukkoa ei ole pakko jakaa tasaisesti saman kokoisiin osiin, vaan jokaisen sarakkeen leveys voidaan määrittellä erikseen (kuva 5).

Yksi		
Kaksi	Kolme	Neljä
	Viisi	

Kuva 5. Kolmeen riviin ja viiteen sarakkeeseen jaettu ruudukko

Kuinka on mahdollista, että sivun tyyliä ja rakennetta voidaan muuttaa niin paljon, vaatiiko se paljonkin, ”kun tämä tapahtuu, niin sitten tämä tapahtuu” - koodia? Ei nykyaikana. Jo vähän aikaa on ollut mahdollista hallita sivun näkymän skaalautumista ja responsiivisuutta pienellä merkinnällä tyylitiedoissa: media, koodissa käytettynä merkinnällä @media. Ethan Marcotten kirjoittamassa artikkelissa selitetään, kuinka media -merkinnän avulla pystytään määrittelemään millä näkymän tai näytön leveyksillä ja korkeuksilla tulee käyttöön halutut tyylit (Ethan Marcotte. 25.5.2010).

Tämän lisäksi media -merkinällä pystytään määrittelemään tarkempia sääntöjä kuin pelkästään laitteen näkymän mittojen mini- tai maksimi- leveyksien ja korkeuksien perus-

teella, mutta monet näistä rajauksista voi olla niin epäkäytännöllisiä tai niillä on tarkka tarkoitus, että ne ovat lähes tarpeettomia tarpeettomia. Media – merkinnän kaikki mahdolliset rajaukset, kuinka kirjoitetaan koodissa ja omakäänteinen selitys:

- width: näkymän, kuten verkkoselaimen ikkunan, leveys.
- height: näkymän, kuten verkkoselaimen ikkunan, korkeus.
- device-width: näytön koko leveys.
- device-height: näytön koko korkeus.
- orientation: onko laite pystyasennossa (portrait), vai vaaka-asennossa (landscape).
- aspect-ratio: näkymän kuvasuhde, eli leveyden suhde korkeuteen.
- device-aspect-ratio: näytön kuvasuhde.
- color: väritarkkuus.
- color-index: laitteeseen tallennettujen värien määrä, pitää olla numero ja ei saa olla negatiivinen numero.
- monochrome: mustavalkoisen näkymän tarkkuuden, pitää olla numero ja ei saa olla negatiivinen numero.
- resolution: resoluutio.
- scan: onko kyseessä ”progressive”, p vai ”interlace”, i televisionäyttö. Esimerkiksi 720p HD TV, p merkitsee ” progressive” kuvan näyttämistä, ja 1080i HD TV, i merkitsee ” interlace” kuvan näyttämistä.
- grid: pohjautuuko laite ruudukoihin vai bitteihin

Näistä rajauksista on osa kuitenkin jäänyt tarpeettomiksi kehityksen takia, ja ovat käytössä vain tukemassa vanhempaa koodia. Vanhoja rajauksia ovat device-width, device-height, sekä device-aspect-ratio. (Frain 2015, 25 - 26)

2.2 Progressive Advancement ja Graceful Degradation

Kun puhutaan responsiivisuudesta, tulee suunnitteluvaiheessa helposti mietittyä mille alustalle sivu tai ohjelmisto on pääasiallisesti tarkoitettu. Vincent Xian kirjoittamassa artikkelin mukaan, on kaksi lähestymistapaa, kuinka kehittää responsiivisia verkkosivuja tai applikaatioita. Progressive Advancement (lyhennettynä PA), ja Graceful Degradation (lyhennettynä GD). Xian mukaan PA suunnittelussa kehitetään ensin versio, missä on vain perustoiminnot ja on ulkonäöltään yksinkertainen, tämä versio mikä on suunniteltu käytettäväksi mobiililaitteissa, ja sen päälle lisätään osia ja toiminnallisuuksia Pc-versiota varten. GD suunnittelu on artikkelin mukaan PAn vastakohta, ensin suunnitellaan ja rakennetaan Pc-versio erilaisilla toiminnoilla, ja osia lähdetään karsimaan mobiili versiota varten. Xian selittää artikkelissa myös miksi mobiiliversio lähtöinen PA tyyli on käytännöllisempi kuin GD, sillä käyttämällä PA lähtöistä kehitystä, kehittäjät pystyvät paremmin keskittymään ohjelman olennaisiin osiin ja toimintoihin ensimmäisenä, varmistaen nopeammin toimivan prototyypin käyttöönoton, ja vasta myöhemmin ohjelmistoon voi kehittää muita vähemmän tärkeitä osia. (Vincent Xia. 21.12.2017)

Olen samaa mieltä artikkelin kanssa PA tyylin olevan käytännöllisempi, ja mielestäni myös seuraa paremmin suunnittelun vaiheita, ensin keskitytään saamaan toimiva ranka, tai esimerkki versio applikaatiosta, ja jälkeinpäin rakennetaan ulkonäkö ja osat mitä ohjelmistossa olisi miellyttävää olla, mutta ei olennaista olla.

2.3 Mobiilille suunnittelu on erilaista

Kun ohjelmistoa suunnitellaan mobiilille, tai olemassa olevasta ohjelmistosta suunnitellaan mobiilille sopivaa versiota, se vaatii hieman erilaista lähestymistapaa kuin tietokoneelle suunniteltaessa näiden laitteitten eroavaisuuksien vuoksi. Tietokonetta pidetään usein pysyvämmällä paikalla, sen käytössä hyödynnetään näppäimistöä ja hiirtä. Mobiililaitteet kulkevat ihmisten mukana, niissä on kosketusnäyttö, ja ne ovat koolta tehoilta pienempiä.

Viitaten Nick Babichin artikkeliin ” 10 Do’s and Dont’s of Mobile App Design”, eli suomennettuna ”10 Tee ja Älä Tee Mobiiliapplikaation Suunnittelussa” on aina ensimmäisenä tunnistettava kohderyhmä, joille mobiiliohjelmisto on tarkoitettu, ja mitä he tarvitsevat. Tutkimalla ja tutustumalla tarkemmin kohderyhmään ja heidän tarpeisiinsa, ohjelmisto voidaan rakentaa kyseiset tarpeet huomioon ottaen, parantaen tulevien käyttäjien tyytyväisyyttä. Babich myös suosittelee, että ennen mobiiliapplikaation suunnittelua, on hyvä idea tutustua kilpailijoiden tarjontaan ja heidän mobiiliapplikaatioiden rakenteeseen. Tätä kautta pystyy olemaan käyttäjän roolissa ja tunnistamaan niitä osia mitkä eivät välttämättä miellytä käyttäjiä, ja paikkaamaan kyseiset ongelmat jo ennen oman mobiiliapplikaation rakentamista. (Nick Babichin 1.2.2018)

Toinen huomioitava osa mobiilille suunniteltaessa on tärkeimpien ominaisuuksien priorisointi. Keskittymällä ohjelmiston rajattuun pääkäyttötarkoitukseen, ja kitkemällä liialliset osat ja toiminnot pois häiritsemästä käyttäjää. Myöskään näkymissä ei saa olla liikaa osia, vaan jokaisen näkymän pitäisi olla varattuna tietylle toiminnolle. Tämä suunnittelu selkeyttää näkymiä, jolloin käyttäjä tietää paremmin mikä näkymän tarkoitus on ja mistä tarvittavat toiminnot löytyvät, ja saa näkymät tuntumaan helpompikäyttöisiltä. (Nick Babich.

1.2.2018) Mitä vähemmän käyttäjä käyttää aikaa haluamansa osan etsimiseen, vaikuttaa se negatiivisesti käyttäjäkokemukseen ja käyttäjän haluun palata takaisin

Babich listaa navigoinnin käytettävyyden selkeyden tärkeimmäksi osaksi kaikissa mobiiliapplikaatioissa. Mikä on ymmärrettävää, sillä mikäli käyttäjä tuntee itsensä eksyneeksi, tai käyttäjä turhautuu hankalan tai epäselvän navigoinnin takia, voi olla mahdollista, että käyttäjä etsii itselleen kilpailijan tarjoaman applikaation. Babich listaa artikkelissaan run-

saasti suosituksia kuinka suunnitella toimiva ja helppo navigaatio. Näihin suosituksiin kuuluu muun muassa reitittämisen arkkitehtuurin järjestäminen tavalla, mikä minimoi tarvittavat toiminnot halutun tiedon löytämiselle. (Nick Babichin 1.2.2018) Välttämällä turhia toimintoja määränpään löytämisessä, käyttäjältä vaaditaan vähemmän aikaa, ja samalla voidaan vähentää riskiä, että käyttäjä eksyy hakiessaan jotain. Tätä voi verrata julkiseen liikenteeseen, kumpaa halutaan tarjota asiakkaalle, reitti määränpäähen minimimäärällä vaihtoja, vai reitti määränpäähen usealla vaihdolla? Molemmissa pääsee perille, mutta toinen vie enemmän aikaa ja vaatii useamman asian huomioimista.

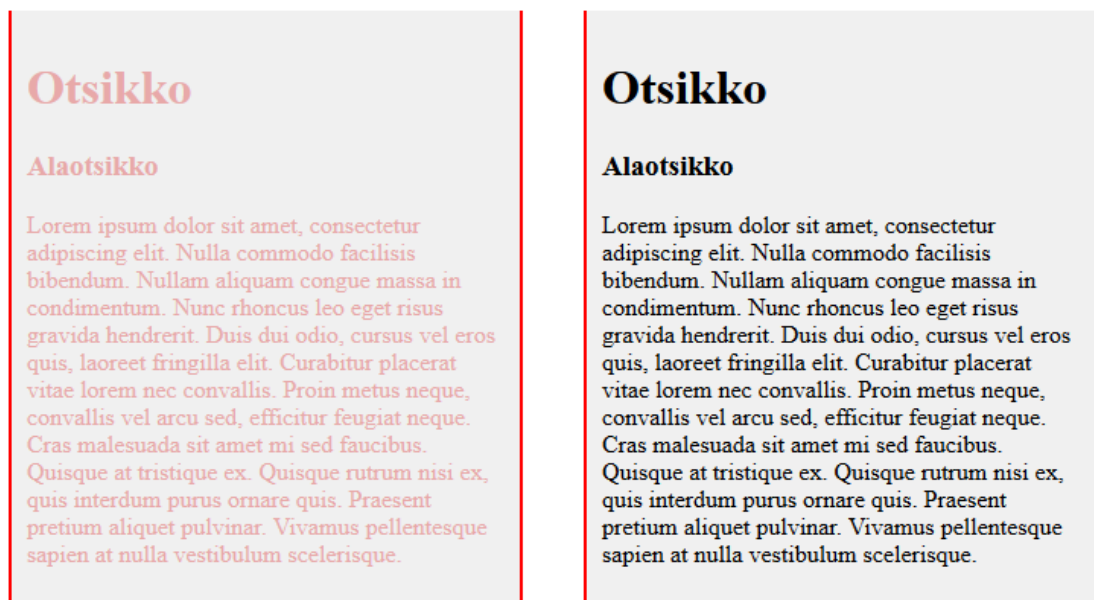
Selkeän reittiarkkitehtuurin lisäksi Babich listaa kuinka tärkeitä on kertoa käyttäjälle missä käyttäjä on millä tahansa hetkellä. Kertomalla käyttäjälle missä tämä on applikaation sisällä selkeällä tavalla, vähennetään eksymisen riskiä. Muita Babichin suosituksia on pitämällä navigoinnin toiminnot yhdenmukaisina. (Nick Babichin 1.2.2018) Muuttamalla toimintoja tai asettelua, joihin käyttäjä on tottunut kesken kaiken, saa aikaan vain hämmennystä käyttäjissä, ja yllättävä muutos ei myöskään tarjoa hyvää kuvaa applikaatiosta.

Koska suunnitellessa mobiililaitteelle, suunnitellaan pienelle kosketusnäytölle. Babichin mukaan, pienet painikkeet jo pienillä näytöillä, on yleinen ongelma mikä häiritsee applikaatioiden käytettävyyttä ja turhauttaa käyttäjiä. (Nick Babichin 1.2.2018) Ratkaisu ongelmaan on selvä ja yksinkertainen, painikkeet on suunniteltava riittävän suuriksi, että niiden painaminen ei ole ongelmallista käyttäjälle, mutta samalla on muistettava, että painikkeet eivät saisi olla liian isoja, milloin ne veisivät liikaa tilaa. Esimerkkinä pienissä painikkeissa voi käyttää sattuman varaisia verkkosivuja, joissa on ponnahdusikkuna mainoksia. Ponnahdusikkuna mainoksissa on useasti pienen pieni sulkemispainike, mitä on vaikea painaa, ja mikäli suunnittelussa applikaatioissa on käytössä samankokoisia painikkeita, käyttäjät eivät varmasti katso hyvällä, jos he joutuvat jatkuvasti yrittämään pienen painikkeen painamista.

Pieni näyttö vaikuttaa painikkeiden käyttämisen lisäksi myös tekstin luettavuuteen. Babich listaa artikkelissaan, kuinka suuri osa applikaatioista on tekstiä, ja tämän takia lukukelpoinen teksti on tärkeä osa applikaatiota. Babich listaa kolme asiaa, mitkä ovat tärkeitä huomioida lukukelpoisen tekstin aikaansaamiseksi, käytettävän fontin tyyli, fontin koko, sekä tekstin taustan ja itse tekstin värien kontrasti. (Nick Babichin 1.2.2018) Fontin tyyliä harvemmin tarvitsee muuttaa, sillä usein tarjolla oleva oletusfontti on yleiseen käyttöön kelpaava, ja fontin tyyliä harvemmin tarvitsee muuttaa. Mikäli applikaatiossa halutaan oletusfonteista poikkeavia fonttityylejä, on muistettava, kuinka hyvin teksti on luettavissa pienemmiltä näytöiltä.

Näyttöjen koko vaikuttaa suoraan sopivan fontin koon valitsemiselle kahdelta osalta. Ensimmäisenä on muistettava, että pieneltä näytöltä on vaikea lukea pientä tekstiä, ellei haluta käyttäjien siristelevän silmiään tai pitävän mobiililaitetta naamassa kiinni. Pienen tekstin lukeminen pieneltä näytöltä on hyvä keino ärsyttää silmiä ja lukijoita, ja saa applikaation tuntumaan epämiellyttävältä. Siksi on hyvä valita riittävän suuri fontti, mutta koko tuo esiin toisen huomiota vaativan osa, teksti ei saa olla liian suurta. Mikäli applikaatioon valitaan liian suuri fonttikoko, tekstistä tulee vaikeampi lukuista, koska mobiililaitteiden pienille näytöille mahtuu rajattu määrä. Valitsemalla liian suuri fonttikoko, teksti vie enemmän tilaa näytöllä, ja pitkien tekstien lukeminen muuttuu työläämmäksi, kun käyttäjä joutuu liikuttamaan näkymää enemmän. Artikkelissa Babich suosittelee käyttämään vähintään fonttikokoa 11, koska sen kokoista tekstiä on vaivatonta lukea normaalitilanteissa ilman näkymän suurentamista (Nick Babichin 1.2.2018).

Kolmantena huomioitavana asiana Babich listaa tekstin ja taustan välisen kontrastin. Babichin mukaan, pitämällä tekstin ja taustan värien eroavaisuus riittävän suurina, tekstiä on helpompi lukea, kun teksti ei sulaudu taustaan (Nick Babichin 1.2.2018). Valkoiselta pohjalta on helpompi lukea mustaa tekstiä, kuin vaalean harmaata tekstiä (Kuva 6). Poikkeavissa tilanteissa, kuten painikkeissa tai tyylitellyissä näkymissä, voi käyttää väriympyrää apuna valitessa riittäviä kontrasteja tekstille ja taustalle.



Kuva 6. Pienen ja suuren kontrastin ero

Artikkelissaan Babichin suosituksiin kuuluu interaktiivisten osien tarjoama palaute, tällä Babich tarkoittaa, esimerkiksi kun käyttäjä painaa applikaatiossa painiketta, on tärkeätä

näyttää käyttäjälle selkeällä visuaalisella tavalla, että nappia on painettu tai jotain on tapahtumassa. Babichin mukaan, tarjoamalla visuaalinen varmistun, että käyttäjän toiminto on tapahtunut, varmistetaan käyttäjälle tämän interaktion tapahtuminen tai aktivoituminen, jolloin käyttäjä ei joudu arvaamaan rekisteröitykö tapahtuma. Babich jatkaa selittämällä, kuinka välittämällä tieto käyttäjälle interaktion aktivoitumisesta tai sen latautumisesta, käyttäjä on tietoinen interaktion tilasta, eikä joudu arvaamaan onko tapahtuma keskeytynyt vai päättynyt onnistuneesti. (Nick Babichin 1.2.2018)

Babich suosituksiin mitä mobiiliapplikaatioita suunnitellessa kannattaa tehdä, listautuu kirjoittamisen minimointi, sen hitauden ja virhealttiuden takia. Babich suosittelee pitämään lomakkeiden koot niin pieninä kuin mahdollista ja karsimaan tarpeettomat kentät, sekä automatisoimaan niitä osia mitkä on mahdollista täyttää ennalta, esimerkiksi päiväys tai kellonaika. (Nick Babichin 1.2.2018) Rajoittamalla käyttäjiltä vaadittavien tietojen syöttöjen määrä minimiin, annetaan käyttäjille tunne käytön vaivattomuudesta ja sulavuudesta. Lisäksi on hyvä muistaa, että mobiiliversio on usein rakennettu tarjoamaan nopeasti saatavaa vastausta ongelmaan, tämän takia mobiiliapplikaatio on hyvä rakentaa tavalla mikä vaatii käyttäjältä mahdollisimman vähän toimintoja.

Babichin listaamiin suosituksiin kuuluu, kuinka on hyvä idea pyrkiä yhdistämään applikaation eri versiot, kuten tietokone ja mobiili, toisiinsa, koska tämä tarjoaa käyttäjälle yhtenäisen applikaation. Keinoja yhtenäisen applikaation tarjoamiselle on esimerkiksi hakutietojen tai suosituksien säilyttäminen laitteelta toiselle. (Nick Babichin 1.2.2018)

Babichin artikkelissa viimeisenä suosituksena on asia minkä pitäisi olla itsestäänselvyys jo ennen ensimmäisen koodirivin kirjoittamista: Aina testaa käyttöliittymä (Nick Babichin 1.2.2018). Testausta ei voi olla painottamatta liikaa, vaikka kuinka hyvältä käyttöliittymä suunniteltaessa voi vaikuttaa, ja vaikka kuinka hyvin applikaatio on kirjoitettu, ikinä ei voi tiedä täydellä varmuudella miltä applikaatio näyttää, miten se toimii, miltä sen käyttö tuntuu. Testatessa voi törmätä ongelmiin tai asioihin mitä ei ole huomioitu, ja mitkä voi vaikuttaa negatiivisesti loppukäyttäjän mielipiteeseen. Testaamalla applikaatiota testikäyttäjillä auttaa löytämään mahdollisia ongelmia, tai kuulemaan testaajien mielipiteitä käyttöliittymästä. Näillä tiedoilla varmistetaan, että julkaistu applikaatio toimii ilman jatkuvia ongelmia, ja sen suunniteltu toimimaan oikein loppukäyttäjän käsissä.

2.4 Haasteet mobiilille suunnittelussa

Vaikka responsiivinen suunnittelu keventää ohjelmiston koodaajien työkuormaa, se ei yksin vapauta ohjelmoijaa kaikista ongelmista ja vastuista. Kun ohjelmiston mobiiliversiota

suunnitellaan, on edelleen tärkeätä muistaa älypuhelinien tuomat rajoituksen ja suunnitella mobiiliversio niiden perusteella.

Nielsenin ja Budiun mukaan mobiilille suunnittelussa on kaksi kohtaa mitkä vaikuttavat huomattavasti ohjelmistojen mobiiliversioiden suunnitteluun. Ensimmäisenä on mobiiliversion näkymän ja navigoinnin tulee olla rakennettu yhteen tavalla mikä varmistaa haettavien asioiden helpon löytämisen. Toiseksi on huomioitava kuinka suunnitella mobiiliversio pienen näkymän ja hitaamman internetyhteyden ympärille. Nielsen ja Budiun jakavat ongelmat mihin käyttäjät eittämättä tulevat törmäämään neljään osaan. (Nielsen & Budiun 2013, 50-51)

Ensimmäisenä ongelmana tuodaan esille, on mobiililaitteiden näytön pienuus verrattuna pöytätietokoneiden näyttöihin. Mobiililaitteiden näytön pieni koko rajoittaa kuinka paljon käyttäjä näkee sivusta kerralla, ja tämä pakottaa käyttäjän muistamaan pienempiä asioita enemmän ja useammin, varsinkin jos kyseessä on suurempi sivu. Lisäksi mobiililaitteiden näytön pieni koko rajoittaa eri osien kokoa ja kuinka paljon niitä voi olla kerralla näkyvillä, ilman että ne tekisivät näkymästä sekavan tai liian turvonneen tuntuisen. Mobiililaitteiden näytön pieni koko vaatii mobiiliversion suunnittelun huomioivan mitkä osat ovat tärkeimpiä sujuvan käytön kannalta, kuinka paljon ja minkälaista informaatiota käyttäjälle halutaan tarjota. Mobiililaitteista kerralla saatavan informaation huomioimisen lisäksi on tärkeätä tunnustaa kuinka paljon tai vähän tilaa kyseinen informaatio vie. Mikäli näkymä on liian tiiviisti pakattu täyteen, tämä hankaloittaa käyttäjälle relevantin osan löytämistä tai tekstin seuraamista. Liian harvaan levitettynä syntyy turhaa tilaa, mikä pakottaa käyttäjän muistamaan paljon pieniä osia kokonaisuudesta, kuten aikaisemmin mainittiin. Siksi suunnittelussa on huomioitava tärkeimmät ja oleellisimmat kohdat, ja rakentaa näkymä niiden ympärille. (Nielsen & Budiun 2013, 50-51)

Toisena ongelmana Nielsen ja Budiun listaavat mobiiliversion painikkeiden käytön ja kirjoittamisen vaikeuden. Koska mobiiliversion ei ole tietokoneiden hiirtä tai näppäimistöä, niin erinäisten painikkeiden, tekstikenttien ja tekstissä esiintyvien hyperlinkkien painaminen voi olla vaikeata tai joissain tapauksissa, varsinkin kun pieniä painikkeita on useampi vierekkäin, väärän painikkeen painaminen sormella on mahdollista hiiren osoittimen puuttuessa. Myöskään sivun vertikaalista liikuttaminen ilman hiiressä olevaa rullaa, tai näppäimistöissä olevia painikkeita, on hankalampaa, ja voi aiheuttaa ongelmia, kuten käyttäjä vahingossa painaisi painiketta yrittäessään liikuttaa näkymää vertikaalisesti. Mobiililaitteilla tekstin kirjoittaminenkin on huomattavasti hankalampaa pienien näppäimien ja kosketusnäyttöjen takia. Nämä ongelmat lisäävät kirjoitusvirheiden todennäköisyyttä. (Nielsen & Budiun 2013, 51)

Kolmantena ongelmana Nielsen ja Budiu listaavat latausnopeuden. Vaikka nykyaikana mobiililaitteiden verkkonopeus on kasvanut sitten vuoden 2013, verkkonopeus puhelimissa on pienempi kuin tietokoneissa mitkä on yhdistetty modeemeihin. Mobiiliversio latautuu hitaammin kuin tietokoneversio, ja näkymän latautumiseen kuluva aika, vaikka se kyse olisikin muutamista sekunneista, voi olla turhauttava käyttäjälle. Neljäntenä ja viimeisenä ongelmana Nielsen ja Budiu listaavat ne mobiiliversiot, mitkä ei ole optimoitu toimimaan kunnolla mobiililaitteissa, näihin voi kuulua esimerkkeinä versiot mitkä eivät skaa laudu sopimaan pienemmille näytöille, versiot missä näkymän rakenne on epäystävällinen käyttäjälle, tai ylipäättänsä suunnittelussa ei ole huomioitu mobiiliversiota ollenkaan. (Nielsen & Budiu 2013, 51)

Nielsenin ja Budiun esittämistä kolmesta ongelmasta, kaksi on enemmän yhteydessä toisiinsa, mobiililaitteitten näytön pienuus verrattuna pöytätietokoneiden näyttöihin, ja mobiiliversioiden painikkeiden käyttö ja kirjoittamisen vaikeus, sillä näytön koko ja yhdellä kertaa näkyvä tila vaikuttavat käytettävien painikkeiden ja tekstin kokoon. Mikäli painikkeet ja teksti ovat suurempi kokoisina helpottamaan käyttöä ja lukemista, niin näytön koon takia käyttäjä näkee vähemmän yhdellä kerralla ja joutuu täten muistamaan enemmän. Mikäli käyttäjän halutaan puolestaan näkevän enemmän yhdellä kertaa, johtaa se usein painikkeitten käytön vaikeutumiseen ja tekstin lukeminen hankaloituu, kun käyttäjä joutuu keskittymään pienen tekstin lukemiseen. Nämä kaksi ongelmaa yhdessä vaativat mobiiliversion suunnittelussa runsaasti huomiota, miten tarjota käyttäjälle tärkeimmät ja oleellimmat asiat, ilman vaivan aiheuttamista käyttäjälle.

Kolmatta ongelmaa, eli ohjelmiston latausnopeutta, on vaikeampi hallita suunnittelussa, koska se on riippuvaisempi käyttäjän mobiililaitteen internet yhteyden nopeudesta ja muista mahdollisista ulkoisista tekijöistä, mitkä vaikuttavat mobiililaitteen latausnopeuteen. Palvelun tarjoajalla voi olla ongelmia yhteyden kanssa, käyttäjän ympäristö voi häiritä latausnopeuksia, esimerkiksi tunneleissa, tai käyttäjän puhelimessa on ongelma mikä hidastaa latausnopeuksia. Mutta mobiiliversion suunnittelussa on mahdollista parantaa latausnopeutta, tai enemmänkin kyse on, että mobiiliversiossa on vähemmän ladattavaa, ja tämän takia on nopeammin ladattavissa. Yksi suurimmista latausaikaa syöivistä osista mobiiliversioissa on kuvat. Kuvien kanssa on tärkeätä rajoittaa niiden kokoa mobiiliversioissa, on tarpeetonta ladata pienen älypuhelimien 360x640 pikselin tarkkuuden näytölle kuvaa, minkä laatu ja koko on tarkoitettu 1920x1080 pikselin tarkalle tietokoneen näytölle, kuvan lataaminen veisi huomattavan määrän aikaa mobiililaitteen verkkoyhteydellä, varsinkin jos samalla sivulla olisi useampi saman suuruista kuvaa.

Nielsen ja Budiu huomauttavat kirjassa, kuinka suuri määrä kuvia hidastaa latausnopeuksia, mutta myös mainitsevat kuvien vetoavan käyttäjiin. He myös esittävät yhdeksi keinoksi vähentää kuormitusta, on pelkistää käyttöliittymää ja vähentää sivulatauksia. (Nielsen & Budiu 2013, 80)

2.5 Haasteet mobiilin käytössä

Mobiilin käytössä on ongelmia, mitkä ovat häiriötekijöitä käyttäjälle. Kaikki ongelmat eivät kuitenkaan ole välttämättä sidoksissa ohjelmiston mobiiliversion käytettävyyteen, vaan ovat itse puhelimen aiheuttamia rajoituksia. Yksi näistä ongelmista on lukemisen ymmärtämisen vaikeutuminen mobiililaitteilla.

R.I. Singh ja hänen virkatoverinsa teettivät Albertan yliopistossa kokeen, minkä tulokset osoittivat, että ihmiset ymmärtävät hankalasti käsiteltävän tiedon huomattavasti heikommin, kun informaatio luetaan pienestä näytöstä, kuten älypuhelisten näytöstä, verrattuna pöytä tietokoneen näyttöön. Testissä R.I. Singh ja hänen virkatoverinsa käyttivät kymmenen eri verkkosivun tietosuojakäytäntöjä. Nielsen & Budiu 2013, 102-107)

Kuinka tutkimuksessa päädyttiin siihen, että informaation lukeminen pienemmältä näytöltä on hankalaa, verrattuna saman tiedon lukemiseen suuremmalta näytöltä? Nielsen ja Budiu huomauttavat kuinka paljon eri häiriötekijöitä kokeen suorituksessa poistettiin ja siitä huolimatta lukeminen mobiilinäytöltä oli testin mukaan jopa kaksi kertaa hankalampaa kuin pöytä tietokoneen näytöltä. Puuttuviksi häiriötekijöiksi Nielsen ja Budiu mainitsevat kuinka luettava osa oli suoraan saatavilla ja luettavissa, mikä poisti navigointiin ja käyttöliittymään liittyvät ongelmat. Toiseksi puuttuvaksi häiriötekijäksi mainitaan ympäröivien virikkeitten puute, sillä testi suoritettiin suljetussa ympäristössä, mistä puuttui yleiset virikkeet mitä tulee vastaan julkisissa paikoissa liikkeessä puhelimen kanssa, kuten melu ja visuaaliset ärsykkeet. Näiden tietojen perusteella kokeessa päädyttiin näytön pienen koon aiheuttavan vaikeuksia tekstejä lukiessa. Nielsen ja Budiu listaavat kaksi syytä miksi näytön koolla on niin suuri vaikutus siihen, kuinka hyvin lukija ymmärtää lukemansa. Ensimmäiseksi syyksi listataan, kuinka vähän tekstiä mobiililaitteiden näytöille mahtuu kerralla, ja tämä estää käyttäjää lukemasta suurempia määriä kerralla ja pakottaa muistamaan ne asiat mitkä eivät ole enää näkyvillä. Toiseksi syyksi Nielsen ja Budiu mainitsevat, kuinka lukija joutuu lukiessa siirtämään näkymää jatkuvasti, ja joutuessaan tarkastamaan aikaisempaa osaa nopea vilkaisu ei riitä, vaan lukija täytyy siirtämään koko näkymää hakiesaan tiettyä kohtaa nopean vilkaisun sijaan. Tämän lisäksi Nielsen ja Budiu listaavat liikkuvalla näkymälle kolme ongelmaa:

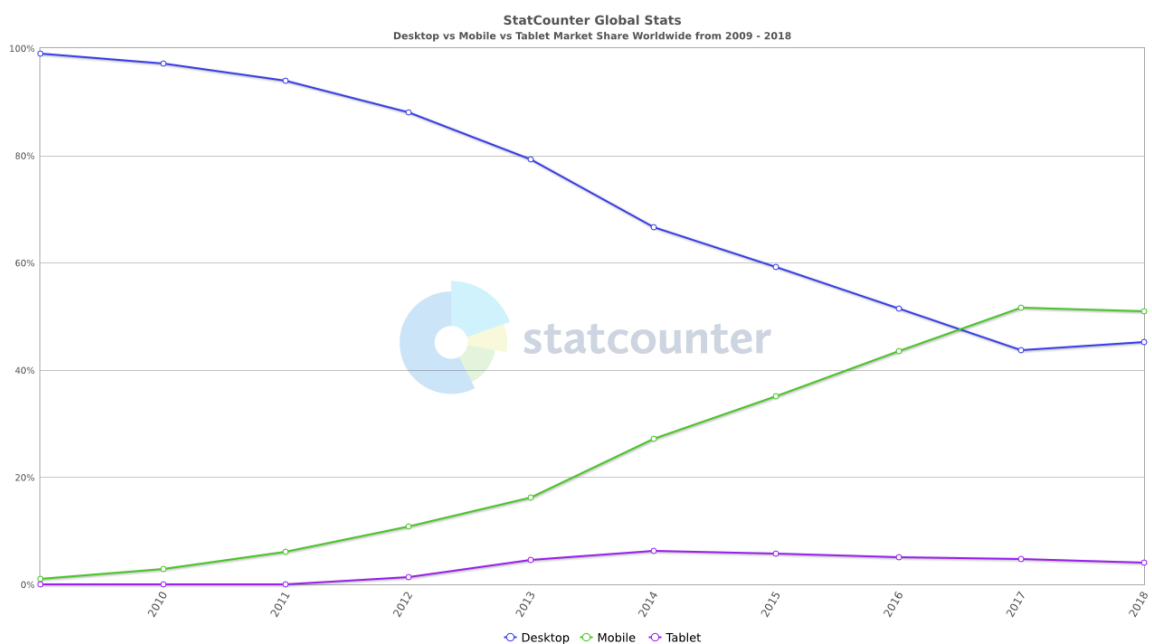
- Se vaatii enemmän aikaa, täten häiriten muistia.

- Se jakaa lukijan huomion, kun tämä joutuu etsimään sitä osaa missä haluttu tieto sijaitsee
- Se vaatii lukijaa etsimään uudelleen kohta mikä vaati aikaisemman tiedon hakemista

Näiden syiden takia on tärkeää varmistaa, että mobiiliversio on optimoitu välttämään tarpeetonta liikutusta ja toimintojen hakemista, ja poistamalla mahdolliset tarpeettomat ja mahdollisesti häiriötä aiheuttavat osat. Mobiililla osien selkeys ja lukemisen helppous ovat välttämättömiä osia toimivan mobiilinäkymän luomisessa, tekstin tulee olla tiivis, helposti luettavissa ja tarvittaessa hyödyntää luettelomerkintöjä, sekä osien jakamista selkeillä reunoilla, mitkä erottavat mahdollisesti eri osat toisistaan. (Nielsen & Budiu 2013, 102-107)

2.6 Miksi responsiivinen suunnittelu ja mobile first ovat nykyään tärkeitä

Ajat ovat muuttuneet siitä, kun rakennettiin verkkosivu sopimaan muutaman eri kokoisen tietokoneen näytölle. Ennen verkkosivujen näkymät koodattiin pikselintarkoiksi ja tämän seurauksesta verkkosivut saattoivat olla hyvinkin hankalia käyttää tai niiden tyyli näytti rikinäiseltä, mikäli niitä katsottiin näytöltä, minkä kokoa ei sivun koodauksessa ollut otettu huomioon (Kim Bohyun. 2013). Kuten kuvasta 7 nähdään, puhelinten käyttö verkon selaamiseen alkoi kasvamaan nopealla vauhdilla vuodesta 2009, ja tätä huomattavaa kasvua voi pitää yhtenä tekijänä miksi pikselintarkkojen verkkosivujen kehittäminen alkoi vaihtua joustavampaan responsiiviseen suunnitteluun. Kasvaneen puhelinten verkkoselauksen lisäksi eri mallisten puhelinten näyttöjen koot kannustivat myös suunnittelemaan automaattisesti mukautuvia verkkosivuja pikselintarkkojen sivujen tilalle (Kim Bohyun. 2013).



Kuva 7. Verkkoselainten käyttö eri laitteilla maailmassa väliltä 2009-2018 (Statcounter 2019)

Poissiirtyminen pikselintarkoista verkkosivuista oli omasta näkökulmastani muutakin hyötyä, kuin käyttäjäkokemuksen parantaminen automaattisesti näkymän kokoon sopeutuvalla suunnittelulla, hylkäämällä pikselintarkat näkymät ja vaihtamalla responsiiviseen suunnitteluun, todennäköisesti samalla vapautui työtunteja koodaajille, sillä heidän työtaakkansa pieneni, koska heidän ei tarvitse kuluttaa turhaa aikaa tarkkaan aseteltuun usealle eri koolle, vaan koodaajat pystyvät rajaamaan muutoksia helpommin ja vähemmällä vaivalla. Lisäksi mikäli erikokoisille näytöille tarkoitetut näkymät olivat erillisinä osina, responsiivisen suunnittelun avulla vähentyisi myös kaikkien erikokoisten sivujen vaatima ylläpito ja yhtenäisyys.

Näkemyksiä tukee Rempelin ja Bridgesin tutkimus, missä he toteavat kuinka nopeasti responsiivinen suunnittelu otettiin käyttöön monipuolisuuden ja erityisesti sen käytännöllisyyden takia. Responsiivisen suunnittelun avulla ei ole tarpeellista rakentaa jokaiselle eri laitteelle omaa sivustoa, tai ei tarvitse erikseen rakentaa siirtosivua, millä käyttäjä siirrettäisiin sen hetkisen laitteen optimoidulle sivulle. (Rempel & Bridges, 2013)

Mobile First - suunnittelu on lyhyesti selitettynä juurikin nimensä veroinen, web-sivun osat ja käyttöliittymä suunnitellaan ensin pienelle puhelimen näytölle sopivaksi, ja sitä käytetään lähtöpisteenä, kun sivuston rakennetta kehitetään sopimaan suuremmille näytöille. Mobile First suunnittelu käyttääkin aikaisemmin mainittua Progressive Advancement - tyyliä.

Kuten kuvassa 7 nähdään, verkkoselainten käyttö älypuhelimilla on kasvanut runsaasti viimeisen kymmenen vuoden aikana, vaikkakin puhelimen käyttö internetin selaamiseen on vasta viimeisen kuuden vuoden aikana kasvanut tietokoneen rinnalle käytön määrissä. Oman käyttökokemuksen ja arvion kautta, oletan kasvun mitä todennäköisesti johtuvan älypuhelinomistajien määrän kasvusta, suuremmista älypuhelin näytöistä, paremmin toimivista ja varmemmista langattomista verkoista, sekä älypuhelin prosessointikyvyt ja tehot ovat kasvaneet niin suuriksi ja voimakkaiksi, että puhelinta voidaan rinnastaa jo pieneen tietokoneeseen mikä kulkee helposti ihmisten taskussa.

Eri yritykset ovat oletettavasti huomioineet tämän kasvun ja tuntunevat nykyään odottavan, että jokaisella on oma älypuhelin, kun katsotaan, kuinka moni eri palvelu on siirtynyt joko kokonaan digitaaliseksi tai on automatisoitu. Esimerkiksi pankkitunnukset voidaan katsoa puhelimen applikaatiosta, ja yhä harvempi pankki lähettää paperisia tunnuslukuja asiakkaille. Minusta tämä kertoo, kuinka isoksi osaksi älypuhelimet ovat tulleet elämäämme.

On mahdollista arvioida, että juurikin tämä puhelinten yleistymisen ja käytettävyys ovat jokuakin pakottaneet web-suunnittelun muuttamaan ja huomioimaan eri kokoiset näytöt suunnittelussa. Mikäli verkkosivut ovat hankala käyttää puhelimen tai tabletin kosketusnäytöllä, siinä riskeerataan mahdollinen asiakas, koska tämä voi lykätä sivustolla käymistä tietokoneella, unohtaa sivuston tai löytää kilpailevan yrittäjän sivut.

2.7 Responsiivisen suunnittelun heikkouksia

Responsiivisen suunnittelun kätevyys muuttaa automaattisesti näkymän osien tyyliä ja paikkaa automaattisesti näkymän koon muuttuessa soveltuakseen paremmin eri kokoisille näytöille ilman käyttökokemuksen heikentymistä tai ilman tarpeettoman vaivan aiheuttamista käyttäjälle. Kuinka monipuolisilla ja halutessa tarkoilla säännöillä pystytään rajaamaan halutut muuttumispisteet. Sekä responsiivisen suunnittelun vallankumouksellinen saapuminen ja kehittyminen monipuoliseksi vastaamaan muuttuvaa ympäristöä puhelinten käytön kasvu verkon selaamisessa. Nämä saavat responsiivisen kehityksen kuulostamaan täydelliseltä vastaukselta millä ei ole mitään heikkouksia, mutta responsiivisella suunnittelullakin on omat heikkouksensa.

Yksi suuri asia mitä responsiivisessa suunnittelussa tarvitaan, on ymmärrys käyttäjistä ja mitä heille halutaan tarjota. On huomioitava laitteiden eri koot, ja ulkoasu on rakennettava huomioimaan pienimmissä näkymissä tavalla mikä ei olisi epäselvä ja sotkuinen. Tärkeimmät toiminnot tulisi olla selvästi näkyvillä, ja niiden tulisi olla helposti käytettäviä. Ulkoasun suunnittelun vaatimusten lisäksi responsiivisessa suunnittelussa on yksi suuri heikkous: Responsiivisen verkkosivun lataaminen kestää kauemmin, koska sen sijaan, että latautuisi yksittäiselle laitteelle optimoitu sivusto, latautuu responsiivisessa sivussa verkkosivun kaikki ominaisuudet, vaikka niitä ei kyseisellä laitteella tarvitsisikaan (Rempel, Hannah Gascho & Bridges, Laurie. 2013).

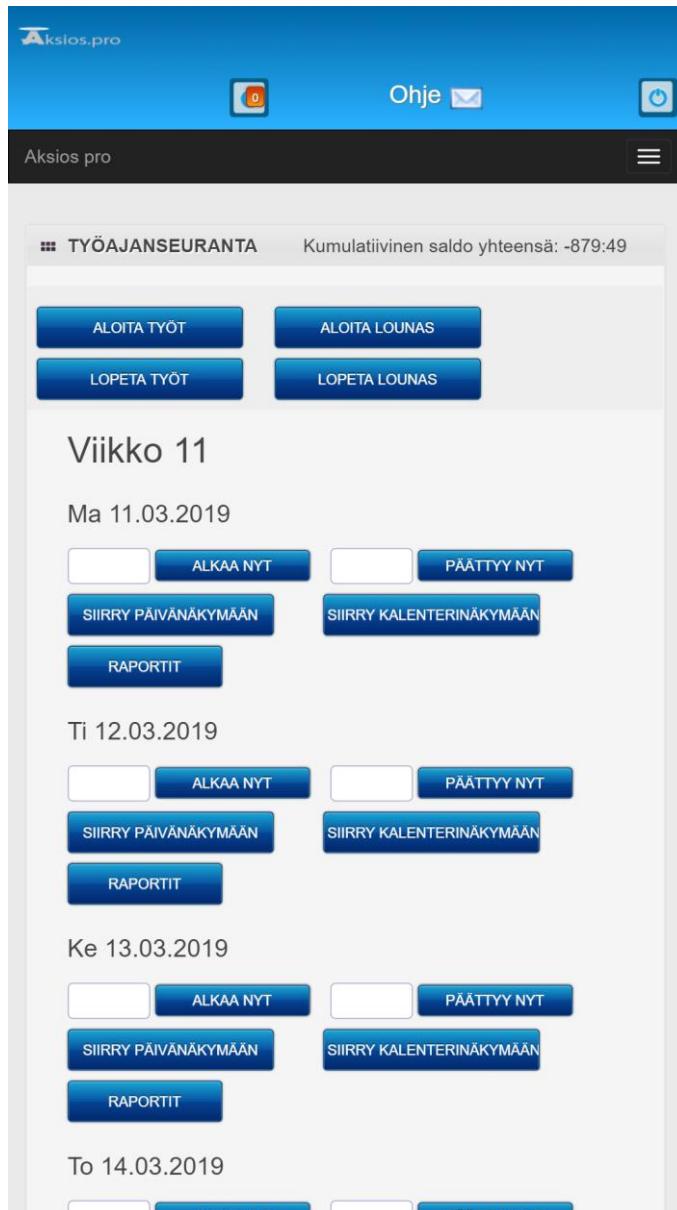
3 Tuotokset

Aksios Corporation on vuonna 2003 perustettu pieni suomalainen yritys, mikä tarjoaa organisaation johtamisen koulutuksen lisäksi heidän itse rakentamaa palveluiden johtamisjärjestelmää Aksios.prota. Aksios Corporationin palveluiden johtamisjärjestelmä Aksios.pro on kuudentoista osan modulaarinen ohjelmisto, ja tämä osa käsittelee opinnäytetyön tekijän tehtävää kehittämään kyseisen ohjelmiston mobiilinäkymän responsiivisuutta, ja käyttöliittymän toimivuutta.

Kuten aikaisemmin johdannossa mainittiin, Aksios.pro on vanha ohjelmisto ja ohjelmiston mobiilinäkymät eivät ole responsiivisia, selaimen näkymän koon mukaan skaalautuvia sivuja, vaan ohjelmistolla on erilliset näkymät mobiililaitteille ja tietokoneille. Vaikka sivuilla on omat mobiilille tarkoitetut näkymät, oli ohjelmiston mobiilipuolella silti ongelmia skaalautumisen kanssa.

3.1 Aksios Corporationin ohjelmiston responsiivisuuden kehitys

Ohjelmiston kehitys alkoi hitaasti, sillä ohjelmisto on rakennettu php:lla, ohjelmointikielellä, johon minulla ei ollut vielä silloin juurikaan kokemusta, mutta en antanut sen häiritä. Aloitin työn tutustumalla ohjelmiston mobiilinäkymään Huawei Honor 9 Lite näytön näkymästä. Ensimmäisenä näkyvänä ongelmana minkä huomasin mobiilisivuilla, oli sisällön koko: kaikki elementit ja fontit olivat pieniä puhelimen näytölle. Toisena ongelmana mobiilinäkymässä oli eri elementtien, kuten painikkeiden ja tekstilaatikoiden sommittelu. Kuvasta 8 voi nähdä ongelmat sommittelussa ja elementtien koossa.



Kuva 8. Mobiili Työajanseuranta ennen korjauksia

Kävin ohjelmiston eri sivut läpi yksi kerrallaan, ja samalla kirjasin ylös eri ongelmakohtia, joita pyrkisin korjaamaan. Korjausta ja hienosäätöä vaativat olivat muun muassa aiemmin mainitut sisällön pieni koko puhelimen näytöllä, elementtien sommittelu sulavammaksi ja kosketusnäytölle sopivammaksi, kuten painikkeiden ja fonttien koon suurentaminen käytön helpottamiseksi. Lisäksi niillä sivuilla, joilla oli popup-tyyppisiä elementtejä, eli sivulle ilmestyvä ikkuna tai elementti, minun piti saada kyseiset elementit pysymään näkymän sisällä.

Kun olin omasta mielestäni tutustunut riittävästi ohjelmiston, niin mobiili ulkoasuun kuin pc version, ulkoasuun, ja saanut parempaa ymmärrystä millainen ohjelmiston yleisilme oli,

sekä millaista ilmettä aikaisemman mobiiliversion ulkoasulla pyrittiin tuomaan esille, ymmärsin paremmin millaista tyyliä kannattaisi lähteä tavoittelemaan, jotta ulkoasujen uudet tyylit pysyisivät yhdenmukaisina.

Aloitin ohjelmiston ulkoasun työstämisen aluksi kokeilemalla koodiin tulevia mahdollisia muutoksia Chrome-verkkoselaimen konsolin tyylihallinnalla, mikäli muutokset paransivat ulkoasua, kirjasin muutoksen tiedot ylös ja lisäsin ne koodiin.

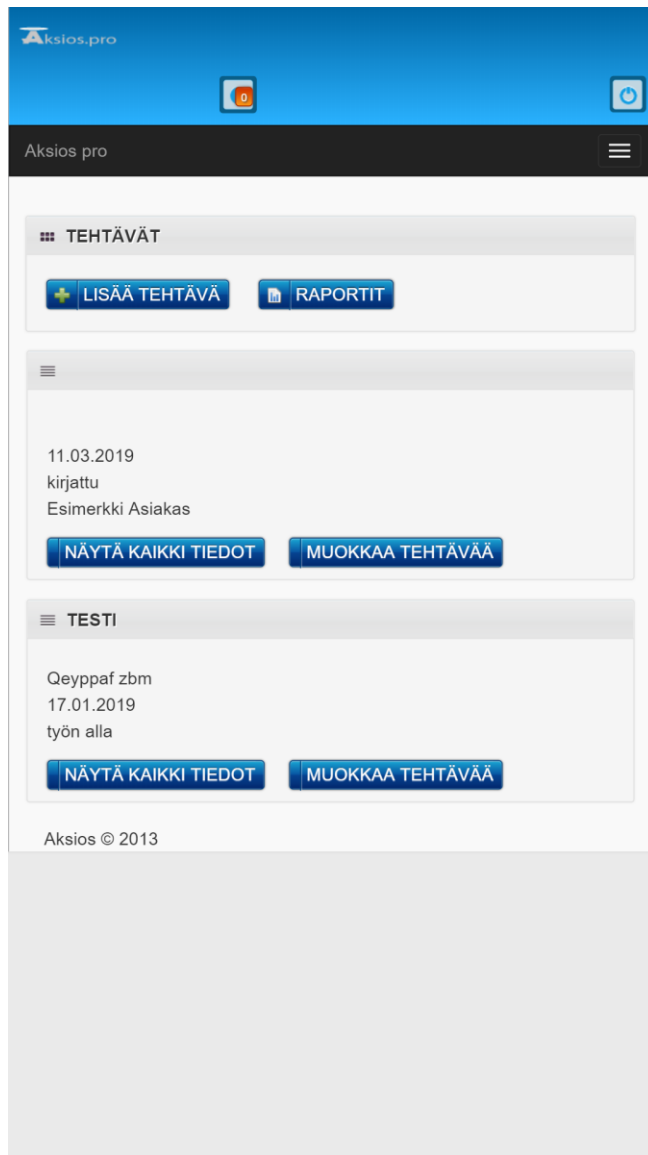
Alussa kuitenkin huomasin ongelmia tyylien korjaamisen kanssa, sillä paljon ohjelmiston elementtien tyyleistä ei ollut kirjoitettu tyylitiedostoksi, vaan oli kirjoitettu osaksi itse elementtiä, ja elementtiin kirjoitetut tyylit ovat korkeampana prioriteettina kuin tyylitiedostoon kirjoitetut tyylit. Ratkaisin ongelman yksinkertaisesti poistamalla elementtiin kirjoitettu tyyli, ja luomalla uuden tyylitiedoston mistä elementit ottaisivat kirjatut tyylit. Pitämällä tyylit kirjattuna css-tiedostossa, on tyylien hallinta helpompaa, kun samaa tyyliä ei tarvitse kirjoittaa uudestaan ja uudestaan jokaiseen elementtiin, ja välttään mahdollisilta virheiltilä uudelleen kirjoittaessa tyyliä elementtiin

3.2 Sivujen samankaltaisuuksien korjauksia

Ensimmäisenä tehtävänä muutin jokaiselta mobiiliversion sivulta sivun leveyden sataan prosenttiin, näin kaikki näkymät olisivat yhdeltä osin yhdenmukaiset. Syy miksi joillakin sivuilla oli suurempi leveys, oli todennäköisesti, koska aikaisemmassa versiossa haluttiin välttää sivun sivuttaista liikettä, kun näkymässä oli liian leveitä elementtejä. Tämä korjaus toikin esille seuraavan korjausta vaativan osan, kaikki sivut eivät leveyden muutoksen jälkeen mahtuneet ruudun näkymään, vaan levisivät reunan yli, aiheuttaen juuri aikaisemmin mainittua sivuttaista liikettä. Jätin liian leveät elementit korjattavaksi, kun etenisin kyseiselle sivulle. Jokaisessa sivussa oli usein jotain sivu kohtaista, mikä aiheutti ongelman leveyden kanssa.

Aloittaessani yksittäisten sivujen korjaamisen, pyrin saamaan näkymästä yksinkertaista ja symmetristä. Ensimmäinen sivu ja sen osat, mitä aloin muokkaamaan kuntoon oli tehtävät. Kuten kuvasta 9 ilmenee, sivun korjattaviin ongelmiin kuului liian leveät elementit, symmetrisyyden puute ja pieni koko painikkeissa ja fontissa, sekä popup-elementin koko ja ilmestymispaikka. Ensimmäinen korjattava osa tehtävät-sivulla, mitä aloin työstämään, oli elementtien ja niiden sisällön koon suurentaminen. Koska kyseessä oli pienelle kosketusnäytölle tarkoitettu näkymä, pyrin sisällön kokoa suurentamalla helpottamaan painikkeiden käyttöä ja tekstin lukemista. Koko muutosten jälkeen aloitin sommittelun parantelun. Tarkoituksena oli saada näkymästä symmetrinen ja keskelle painottuva, jotta sivua voisi käyttää helpommin yhdelläkin kädellä. Pyrin luettelomaiseen tyyliin rivittämällä objektit ja tekstirivit omille tasoille, ja keskittämällä ne näkymän keskelle saadakseni sulavamman

ylhäältä-alas liikkeen. Halusin välttää tässä tapauksessa useampia objekteja samalla rivillä, jotta tekstit ja painikkeet pystyisi pitämään riittävän suurikokoisina niiden lukemisen ja käyttämisen helpottamiseksi. Mikäli samalla rivillä olisi ollut useampia objekteja, painikkeet olisivat voineet olla liian pieniä ja hankaloittaneet niiden käyttöä, ja tekstien pitäminen omilla riveillä jakoi ne selkeiksi, eikä aiheuttanut riskiä, että vahingossa lukisi tiedot väärin.



Kuva 9. Mobiili Tehtävät sivu ennen korjauksia

Popup-elementin korjaus oli seuraavana vaiheena, elementti ilmestyi liian sivulle, leikaten sen osittain pois näkymästä. Lisäksi popup-elementin sisällön sommittelu ei sopinut yhteen sivun kanssa. Ensin annoin popup-elementille kolme responsiivista pistettä, joissa elementin maksimileveys vaihtui, pisteet oli tarkoitettu puhelimille, tableteille ja tietokoneille. Tällä halusin varmistaa, että elementti ei olisi näkymää leveämpi, ja olisi käytettävissä useammalla eri kokoisella laitteella. Tämän jälkeen etsin koodin missä määriteltiin elementin ilmestymispaikka. Etsinnöissä ei mennyt kauaa, ja aloitin testaamisen. Halusin

popup-elementtien tulevan pienemmissä näkymissä mahdollisimman keskelle, mutta suuremmilla näytöillä sivummalle, jotta se ei olisi liikaa tiellä.

3.3 Työajanhallinnan-sivun vaatimat muutokset

Toinen sivukokonaisuus, työajan hallinta, sisälsi edellisen kanssa samoja ongelmia; fonttien ja painikkeiden pienuutta, sekä elementtien epäsymmetrisyyttä, kuten aikaisemmasta kuvasta 9 voi nähdä. Työajanhallinnan-sivulla aloitin korjaukset samassa järjestyksessä kuin edellisessä tehtävät-sivulla, suurensin painikkeiden kokoa. Työajan hallinta -sivulla muutin neljä tärkeintä painiketta, työajan ja lounaan aloitus ja lopetus painikkeet, selkeämmiksi muuttamalla kyseisten painikkeiden kokoa huomattavasti suuremmiksi. Kun painikkeiden koot tuntuivat hyviltä pienellä puhelimen näytöllä, siirryin elementtien asetteluun. Kuten edellisellä sivulla, pyrin mahdollisimman symmetriseen ulkoasuun, jotta käyttö yhdellä kädellä olisi helppoa ja toimivaa. Ajatuksena oli asetella objektit yhden tai kahden pareina samalle tasolle, jotta interaktiiviset objektit, kuten painikkeet ja lyhyet tekstikentät, pystyisi pitämään riittävän suurina helpottamaan niiden käyttöä, ja jakamaan objekteja saman aiheisiin lohkoihin, kuten esimerkiksi pitämään spesifisen työpäivän työajan aloituksen kellonajan samalla rivillä kuin työpäivän aloitus painikkeen.

3.4 Kalenterin rakenne ja toiminnot mobiilinäkymälle

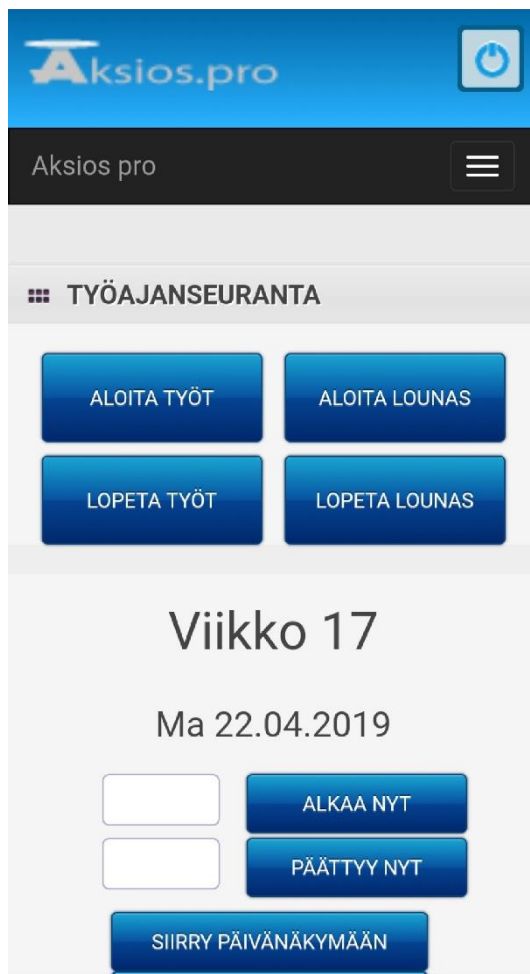
Kolmas sivukokonaisuus, mikä vaati isoja korjauksia, oli kalenteri. Kalenterinäkylässä korjausta vaativat olivat elementtien symmetrinen sommittelu ja kalenterissa esiintyvän tekstin sovittaminen kalenterin pienien osien sisälle, kuten päivämäärät päivänäkylässä. Ensimmäisenä asiana korjasin näkymän leveyden. Kalenteri sisälsi elementtejä, joilla oli määritelty minimileveys pikseleillä, mikä johti pienellä puhelimen näytöllä liian leveisiin elementteihin, aiheuttaen sivuttaista liikettä. Ongelma ratkesi helposti, sillä minimikoko ei osoittautunut tarpeelliseksi. Poistin minimikoon ja muutin kalenterielementin koon skaalautumaan isäntäelementin leveyden mukaan ja käyttämään prosentteja. Lisäämällä kalenterille vielä minimikorkeus pikseleillä, oli mahdollista käyttää kalenteria, sillä leveys oli suhteessa korkeuteen.

Seuraavaksi aloitin päivänäkymän sisällön sovittamisen. Ongelmana oli saada päivien lyhenne ja päivämäärä mahtumaan pieneen tilaan. Ratkaisin lopulta ongelman lisäämällä päivämääräelementtiin tyylin, mikä leikkaa tekstin ja asettaa sen jatkumaan seuraavalta riviltä, kun elementin reuna tulee vastaan. Tällä tyyllillä sain päivämäärän mahtumaan elementin sisälle, ilman tekstin menemistä piiloon tai leviämistä oman elementin ulkopuolelle.

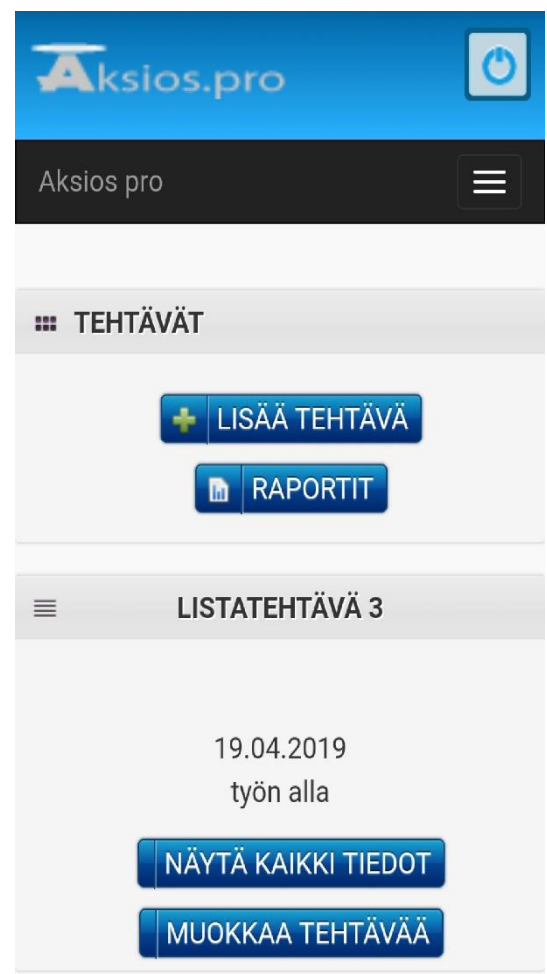
Loput jäljelle jääneet osat olivat tyyleiltään, niin samankaltaisia, että ne sai korjattua samoilla muutoksilla. Järjestämällä otsikot ja niiden syöttökentät listaksi, sai sisällön mahdollisimman symmetriseksi.

4 Pohdinta

Kuten kuvista 10 ja 11 voi nähdä kuinka lopulliset näkymät ovat huomattavasti symmetrisempiä aikaisempiin näkyymiin verrattuna, sekä kuinka elementtien sisältö ja painikkeet ovat suurempia, ja keskitetty keskemälle näkymää helpottamaan painikkeiden käytön kanssa.



Kuva 10. Valmis Työajanseuranta-sivu



Kuva 11. Valmis Tehtävät-sivu

Sijoittamalla sovelluksen painikkeet mobiilinäkymässä keskelle näkymää, pyrin tavoittelemaan siistiä ja järjestelmällistä käyttöliittymää missä käyttäjän ei tarvitse etsiä haettavaa toimintoa, vaan pystyy löytämään haluamansa osan mahdollisimman vaivattomasti seuraamalla sivua ylhäältä alas. Monet sovelluksen osat, kuten painikkeet, kentät ja lyhyet tekstit, tulivat lopuksi aseteltua olemaan yksin tai kaksin samalla tasolla. Tällä asettelulla

mahdollistin objektien kokojen pysyvän riittävän suurina useissa erikokoisissa mobiilinäkymissä, jolloin käyttö pysyisi helppona, ja järjestelmällisyys selkeänä, kuten kuvassa 10 painikkeet ”ALOITA TYÖT” ja ”ALOITA LOUNAS”, sekä ”LOPETA TYÖT” ja ”LOPETA LOUNAS” ovat pareina samoilla tasoilla. Mikäli painikkeet olisivat jokainen omalla rivillä, se olisi vienyt mielestäni tilaa turhaan ja pakottanut käyttäjää hakemaan painiketta listasta. Pitämällä aatteellisesti samat objektit samalla rivillä parina, pysyy näkymä selkeämpänä käyttää ja vaatii vähemmän hakemista käyttäjältä. Mikäli samalle riville olisi laittanut useampia objekteja, niin tällöin kyseiset osat olisivat voineet olla niin pieniä, että niiden käyttö, kuten oikean painikkeen painaminen, olisi ollut hankalaa käyttäjälle. Kuvassa 11 puolestaan näkyy enemmän luettelomainen järjestely. Tarkoituksena kuvan 11 tyyllisissä sivuissa oli pitää osat selkeinä, sillä kyse oli enemmän tekstimuotoisista objekteista. Tärkeät tiedot ovat selkeämmin näkyvillä keskellä sivua, ja teksti on riittävän suuri luettavaksi pienemmälläkin näytöllä.

Ohjelmiston mobiiliversion ulkoasun kehittämisessä oli omat hankaluutensa. Monessa elementissä oli elementtiin itseensä kirjattuja tyylejä, minkä takia kyseiset osat piti etsiä ja antaa elementeille tyyli luokka mistä vastaavat elementit ottaisivat tyyliänsä. Kun elementti tai sen tyyli toistuu useasti, kuten painikkeet, tyylin kirjaaminen elementit sisälle aiheuttavat turhaa työtä ja lisää riskiä virheisiin. Lisäksi ylläpitäminen ja uudistaminen ovat työlämpiä, kun jokainen elementti pitää käydä läpi ja korjata, mikäli elementissä olisi vain tyyli luokkia, pystyttäisiin kaikki muutokset tekemään muuttamalla muutamaa koodin pätkää. Ongelmat ilmenevät myös, kun rakennetaan responsiivisia osia, sillä elementtiin kirjatut tyyli ovat korkeampana prioriteettinä, kuin tyyli luokkaan kirjatut, eli elementti ei muutu.

Aksio Corporation:in ohjelmiston mobiilinäkymän kehityksen aikana opin ymmärtämään php-ohjelmointia. Aikaisempi php-ohjelmoinnin osaaminen oli vain luettua, en ollut aikaisemmin käyttänyt php:ta ohjelmoinnissa. Mobiilinäkymän kehityksen aikana tunsin kuitenkin oppivani nopeasti php-ohjelmointikieltä käytännön tekemisen kautta, ja osaavan tuen avulla taitoni kehittyivät. Mobiilinäkymän kehitys tarjosi minulle myös käytännön kokemusta työmaailman ohjelmoinnista.

Lähteet

Ben Frain. 2015. Responsive Web Design with HTML5 and CSS3. Packt. Birmingham.

Ethan Marcotte. 25.5.2010. Responsive Web Design. Luettavissa: <https://alistapart.com/article/responsive-web-design>. Luettu: 27.3.2019.

Jakob Nielsen & Raluca Budiu. 2013. Mobile Usability. New Riders. Berkeley

Kim Bohyun. 2013. Chapter 4: Responsive Web Design, Discoverability, and Mobile Challenge. Library Technology Reports, 49, 6, s. 29+. Luettavissa: <https://www.questia.com/read/1G1-353319774/chapter-4-responsive-web-design-discoverability>. Luettu: 11.9.2019.

Nick Babich. 1.2.2018. 10 Do's and Dont's of Mobile App Design. Luettavissa: <https://xd.adobe.com/ideas/principles/app-design/10-dos-donts-mobile-app-design/>. Luettu: 4.3.2020

Rempel, Hannah Gascho & Bridges, Laurie. 2013. That Was Then, This Is Now: Replacing the Mobile-Optimized Site with Responsive Design. Information Technology and Libraries. 32, 4, s. 8+. Luettavissa: <https://www.questia.com/read/1G1-357591771/that-was-then-this-is-now-replacing-the-mobile-optimized>. Luettu: 11.9.2019.

Statcounter 2019. StatCounter Global Stats. Luettavissa: <http://gs.statcounter.com/platform-market-share/desktop-mobile-tablet/worldwide/#yearly-2009-2018>. Luettu: 30.3.2019.

Vincent Xia. 21.12.2017. What is Mobile First Design? Why It's Important & How To Make It? Luettavissa: <https://medium.com/@Vincentxia77/what-is-mobile-first-design-why-its-important-how-to-make-it-7d3cf2e29d00>. Luettu: 27.3.2019.