

Responsiivisten sovelluskehysten vertailu

Joni Turunen



Tekijä(t) Joni Turunen	
Koulutusohjelma Tietojenkäsittelyn koulutusohjelma	
Opinnäytetyön otsikko Responsiivisten sovelluskehysten vertailu	Sivu- ja liitesivumäärä 34 + 1
Opinnäytetyön otsikko englanniksi Responsive framework comparison	
<p>Opinnäytetyön tarkoituksena on vertailla kolmea responsiivista sovelluskehystä Bootstrapia, Foundationia ja Skeletonia. Vertailu suoritetaan keräämällä kehittäjien kokemuksia kirjallisista lähteistä. Tavoitteena on selvittää sovelluskehysten sopivuutta eri laajuisiin projekteihin ja sovelluskehysten ominaisuuksien kattavuutta. Lisäksi tavoitteena on selvittää sovelluskehysten muokattavuutta omiin tarpeisiin sopivaksi.</p> <p>Teoriaosiossa perehdytään responsiivisuuden käsitteeseen ja käydään läpi myös siihen liittyviä käsitteitä. Responsiivisuudesta esitetään käsite ja sen lisäksi mihin teknologioiden kehitykseen se perustuu. Lisäksi käydään läpi yleisimpiä vaihtoehtoja responsiiviselle tavalle toteuttaa sivu ja niiden teknistä perustaa.</p> <p>Empiria-osassa käydään läpi sovelluskehysten perusominaisuuksia ja esitetään jokaisesta sovelluskehyksestä yleisimmät ominaisuudet ohjelmakoodi esimerkein. Lisäksi kerätään kehittäjien positiivisia ja negatiivisia kokemuksia niistä. Opinnäytetyön menetelmänä toimii kirjaselvitys. Pohdinnassa käydään läpi lähteitä ja työn mahdollisia jatkokehitys mahdollisuuksia.</p> <p>Rajauksena työssä kerätään kokemuksia vain kehittäjiltä, ei lopullisten sivustojen käyttäjiltä. Myöskään sovelluskehittäjien käyttämään käyttöjärjestelmään tai alustaan ei ole otettu kantaa tai siihen millä kielellä palvelinpuolen toteutus on tehty. Työ keskittyy ainoastaan asiakaspuolen sovelluskehysiin.</p> <p>Kehittäjien kokemusten mukaan Bootstrap on laajin ja parhaiten laajennettavissa oleva, mutta myös hyvin raskas sovelluskehys. Foundation on myös laaja ja silläkin pystyy tekemään laajoja kaupallisia projekteja, mutta se edellyttää graafista osaamista. Skeleton on kevein ja soveltuu pieniin projekteihin ja prototyyppeihin, mutta ei laajoihin kaupallisiin sovelluksiin.</p>	
Asiasanat Responsiivisuus, sovelluskehys	

Author(s) Joni Turunen	
Degree Programme Information Technology	
Report/Thesis title Responsive framework comparison	Number of pages and appendix pages 34 + 1
<p>The aim of the thesis is to compare three different responsive frameworks Bootstrap, Foundation and Skeleton. The comparison is made by collecting developer feedback from literary sources. The objective is to reveal how the frameworks suit projects of different size and the how adequate the number of features is. The aim is also to find out how the frameworks can be customized to meet developer's needs.</p> <p>In the theory part, there is an introduction to the concept of responsive design and concepts related to it. Responsive design is introduced as a concept, as well as the technologies and their development that it is based upon. The most common alternatives to responsive design and the technological base for those are presented.</p> <p>In the empirical part the most used and common features from all the selected frameworks are introduced with code-level examples included. Both negative and positive developer experiences are collected. The methodology for the study is a literary analysis. In the concluding part, the main sources are overviewed and potential further study based on the thesis is discussed.</p> <p>The thesis scope is limited to collecting experiences from developers only, not from site users. Neither the operating system nor platform nor what language has been used for the server-side implementation are limited. The thesis is focused solely on client-side frameworks.</p> <p>According to developer experiences, Bootstrap has the most features and can be extended, but it is also very heavy for performance. Foundation is similarly broad and it also can be used in large scale commercial projects, but it requires good graphical skills. Skeleton is the lightest and can be used for small projects and prototypes, but not for large commercial software.</p>	
Key words Responsive, framework	

Sisällys

1	Johdanto	1
2	Responsiivisuus	3
2.1	Responsiivisuus yleisesti	3
2.2	Vaihtoehtoja responsiivisuutta kohtaan	4
2.3	Teknologinen tausta.....	6
2.4	HTML5.....	7
2.5	Natiivi ja hybridi vaihtoehtoina.....	7
2.5.1	Natiivi.....	8
2.5.2	Hybridi.....	8
2.6	Mobile first	9
2.7	CSS3 ja media kysely	10
2.8	Sovelluskehukset	11
3	Tutkimuskysymykset	12
3.1	Menetelmä ja lähteet.....	12
3.2	Tutkimuskysymykset.....	12
4	Kehittäjien kokemukset	14
4.1	Bootstrap	14
4.1.1	Mediakysely	14
4.1.2	Joustava ruudukko.....	14
4.1.3	Typografia.....	15
4.1.4	Painikkeet	16
4.1.5	Lomakkeet	17
4.1.6	Listat	19
4.1.7	Taulukot	19
4.1.8	Positiiviset.....	21
4.1.9	Kritiikkiä	21
4.2	Foundation.....	22
4.2.1	Mediakysely	22
4.2.2	Joustava ruudukko.....	22
4.2.3	Typografia.....	23
4.2.4	Painikkeet	24
4.2.5	Lomakkeet	25
4.2.6	Listat	25
4.2.7	Taulukot	26
4.2.8	Positiiviset.....	27
4.2.9	Kritiikkiä	27
4.3	Skeleton.....	28
4.3.1	Mediakysely	28

4.3.2	Joustava ruudukko	28
4.3.3	Typografia	29
4.3.4	Painikkeet	29
4.3.5	Lomakkeet	30
4.3.6	Listat	30
4.3.7	Taulukot	31
4.3.8	Positiiviset	31
4.3.9	Kritiikkiä	32
5	Pohdinta	33
	Lähteet	35
	Liitteet	40
	Liite 1. Vertailu sovelluskehysten ominaisuuksista	40

1 Johdanto

Opinnäytetyön aiheena oli responsiivisten sovelluskehysten (framework) vertailu. Vertailtaviksi sovelluskehysiksi valittiin kolme eri vaihtoehtoa: Bootstrap, Foundation ja Skeleton. Bootstrap ja Foundation ovat yleisimpiä sovelluskehyskiä ja niissä kattavasti ominaisuuksia myös laajoihin kaupallisiin sovelluksiin. Skeleton on uusi hieman erilaisella taustajatoksella tehty alusta, se on suunniteltu kevyemmäksi vaihtoehdoksi.

Aihe on kiinnostava, sillä responsiivisuus ja responsiivisten sovelluskehysten käyttö on yleistynyt nopeasti. Pöytäkoneiden näytöt ovat kasvaneet, samaan aikaan kun pienemmän näytön omaavat älypuhelimet ja tabletit ovat yleistyneet nopeasti. Tämän seurauksena lähes kaikkien sivustojen tulee mukautua laitteen ja näytön ominaisuuksiin. Responsiiviset sovelluskehukset tarjoavat tähän helposti käyttöön otettavan ja kattavan ratkaisun.

Tutkimusmenetelmänä opinnäytetyöhön valittiin kirjaselvitys. Tärkeimpiä lähteitä ovat englanninkieliset web-julkaisut, jotka keskittyvät graafisen alan uutisiin ja esittelyihin. Lisäksi kehittäjien kokemuksia kerättiin useista blogipäivityksistä, sekä laajemmista vertailevista artikkeleista. Web-julkaisusta eniten opinnäytetyössä nojataan Smashing Magazine ja A List Apart sivustoihin.

Opinnäytetyöllä on kolme tutkimuskysymystä:

- Minkäkokoisiin projekteihin sovelluskehys sopii?
- Kuinka kattavasti sovelluskehyksessä on ominaisuuksia?
- Onko ominaisuuksien muokkaaminen yksilölliseksi helppoa?

Näiden tutkimuskysymysten taustoja ja tavoitteita avataan tarkemmin Tutkimuskysymykset-luvussa. Sovelluskehyksestä on aiemmin tutkittu esimerkiksi teknisempää puolta kuten nopeutta toteutuksessa. Tämä opinnäytetyö keskittyy selvittämään kehittäjien kokemuksia, siitä minkälaiseen projekteihin nämä sovelluskehukset soveltuvat parhaiten.

Työ on jaettu viiteen pääkappaleeseen. Ensin esitellään teoriataustaa responsiivisuudelle, sekä vaihtoehtoja sille. Tämän jälkeen käydään läpi tutkimuskysymykset ja tärkeimmät lähteet. Näitä seuraa sovelluskehysten ominaisuuksien esittely ja kehittäjien kokemukset. Kaikista sovelluskehyksistä on pyritty esittelemään samat asiat, painopiste on ollut yleisimmin käytetyillä perusominaisuuksilla. Laajemmissa sovelluskehyksissä on huomattavan paljon erilaisia ominaisuuksia ja lisää tulee jatkuvasti, mutta näitä on rajattu pois, jotta voidaan vertailla sovelluskehyskiä keskenään mahdollisimman tasapuolisesti. Lopuksi

seuraa Pohdinta-luku, jossa käydään läpi ja avataan tuloksia, sekä pohditaan omaa oppimista ja opinnäytetyöprosessia.

2 Responsiivisuus

Tässä luvussa käydään läpi opinnäytetyön teoriataustaa. Luvussa esitellään responsiivisuuden käsite ja sen taustalla vaikuttavat HTML5:en ja CSS3:en, sekä teknologinen tausta koko ilmiölle. Lisäksi käydään läpi myös vaihtoehtoja responsiivisuudelle.

2.1 Responsiivisuus yleisesti

Responsiivinen web-suunnittelu on lähestymistapa jossa sivuston ulkoasu ja kehitys reagoivat käyttäjän toimintaan ja käyttöympäristöön näytön koon, alustan (platform) ja laitteen asennon (orientation) perusteella (kuva 1). Responsiivisuus käytäntönä koostuu sekoituksesta joustavia ruudukkoja (grid) ja kuvia, sekä CSS media kyselyiden (media query) hyvästä käytöstä. Verkkosivustoa teknisen toteutus tulisi vastata käyttäjän laitteen ominaisuuksiin, jolloin ei tarvitse tehdä erillistä versiota jokaiselle alustalle ja laitteelle. (Knight 2011.)

De Graeven mukaan responsiivisuuden kolme keskeisintä teknistä ominaisuutta ovat:

- Mediakyselyt ja niiden kuuntelijat.
- Joustava ruudukko joka käyttää suhteellisia kokoja.
- Joustava media ja kuvat, dynaamisen koon muokkauksen tai CSS:n avulla.



Kuva 1. Responsiivisesti mukautuva sivu eri laitteilla (Vinaora 2014)

Käsitteenä Responsiivinen web-suunnittelu ilmestyi ensimmäisen kerran vuonna 2010 Ethan Marcotten kirjoittamasta A list apart-verkkojulkaisussa ilmestyneessä artikkelissa. Se on yleistynyt nopeasti ja sitä käytetään hieman eri tavalla eri yhteyksissä, nykyään yleisesti käytetty termi on responsiivisuus. Johtuen käsitteen laajasta käytöstä, sen merkitys ei ole täysin yksiselitteinen. Käsitettä tullaan käyttämään laajasti eri yhteyksissä vielä jatkossakin, joten sen merkitys tulee laajenemaan ja muuttumaan. (Danger Gardner, 2014.)

Keskeinen käsite responsiivisuudessa on mukautuminen käyttäjän tarpeisiin ja laitteen ominaisuuksiin. Käyttäjän tarpeisiin mukautumisessa ei huomioida ainoastaan mahtuuko sisältö laitteen näytölle. On tärkeää miettiä mikä sisältö on käyttäjälle tarpeellisinta ja esittää oleellisin sisältö ensin. (De Graeve 2011.)

Tablettien ja älypuhelimien määrä on kasvanut huomattavasti viime vuosina. Kaikille alustoille, joilla sivustoja selataan, oman applikaation suunnitteleminen on liian työläs prosessi. On selkeämpää suunnitella yksi sivusto, joka mukautuu erilaisten laitteiden ominaisuuksiin kuin ylläpitää useita eri versioita. (Cashmore 2012.)

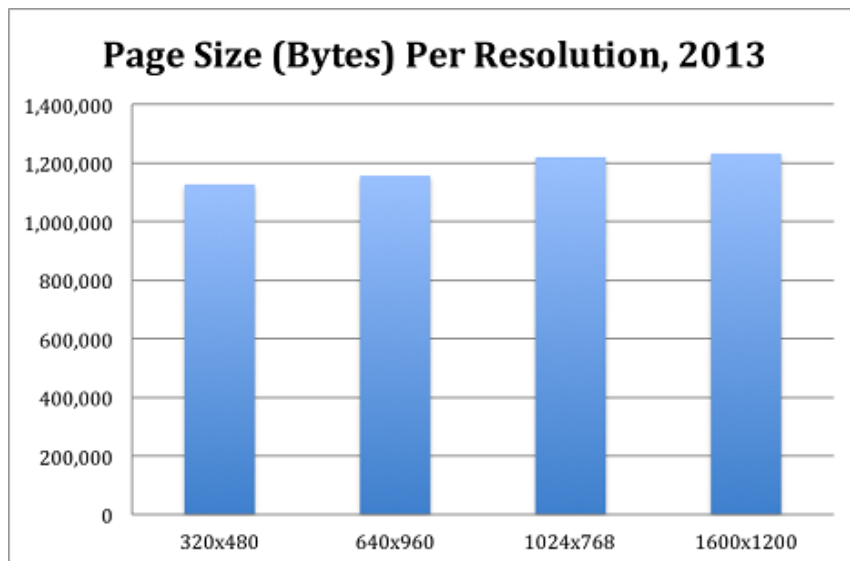
Responsiivinen suunnittelu soveltuu hyvin HTML5:en kanssa, joka parantaa web suunnittelun joustavuutta. Se mahdollistaa web-kehittäjien luoda graafisia, interaktiivisia ja monipuolisia sivustoja, sillä HTML5 tekee multimedian ja graafisen sisällön käsittelystä helpompaa millä tahansa laitteella. (Guertler 2012, 44.)

Ei voida olettaa että käyttäjän ei tarvitse päästä käsiksi kaikkeen sisältöön, koska he ovat mobiililaitteella. On mahdollisesti tarpeen muuttaa fontteja ja vuorovaikutuselementtejä sivustolla mukautumaan kosketusnäytölliseen käyttöliittymään. Sivustojen julkaisijoille responsiivisuus tarjoaa yksinkertaisen tavan saavuttaa käyttäjät eri laitteilla. Käyttäjille se mahdollistaa hyvän käyttökokemuksen useilla eri alustoilla. (De Graeve 2011; Cashmore 2012.)

2.2 Vaihtoehtoja responsiivisuutta kohtaan

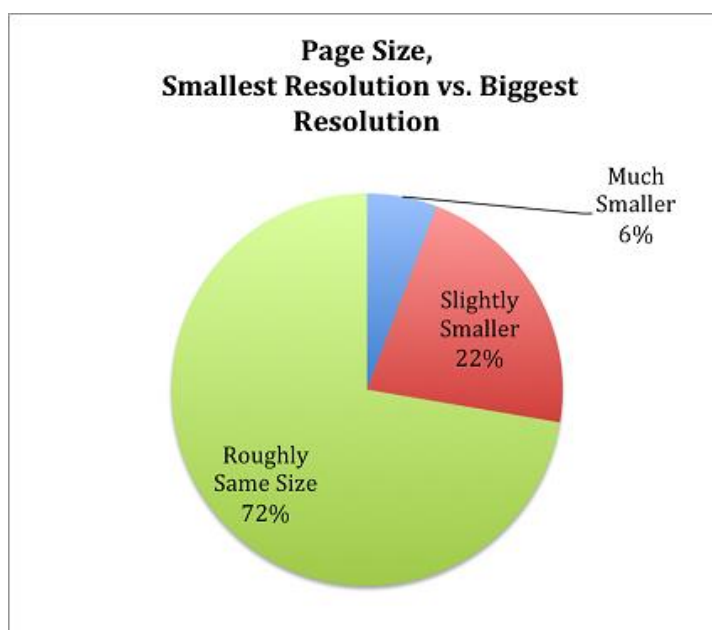
Responsiivisuutta pidetään joskus ratkaisuna joka kattaa kaikki laitteet samalla kertaa, mutta siihen liittyy myös riskejä. Yksi keskeinen riski on sivustojen nopeus, mobiilikäyttäjille nopeus on hyvin tärkeä tekijä. Mobiilisivuston ensimmäisen näkymän lataus saisi kestää korkeintaan sekunnin. (Firtman 2014.)

Responsiivinen suunnittelu sisältää yleensä saman HTML-dokumentin lataamisen kaikille laitteille ja media kyselyiden käyttämisen eri CSS-tiedostojen ja kuvien lataamiseen. Nykyajan pöytäkoneille suunnitellut sivustot ovat hyvin laajoja ja voivat sisältää suuria kuvia, tällaisten sivustojen lataaminen mobiiliyhteyksillä kestää liian kauan. (Firtman 2014.)



Kuva 2. Sivun koko eri resoluutioilla (Podjarny 2013)

Sisällön optimointi laitteen mukaan on harvinaista. Pienillä näytöille suunnitellut sivustojen versiot eivät ole juurikaan pienempiä kuin suuremmille resoluutioille (kuva 2). Pienet sivut ovat keskimäärin vain 7 prosenttia pienempiä kuin suuremmat sivut. Valtaosa sivuista on suurin piirtein yhtä suuria riippumatta laitteen koosta (kuva 3). Huomattavasti pienempiä sivustoista on vain murto-osa. (Podjarny 2013.)



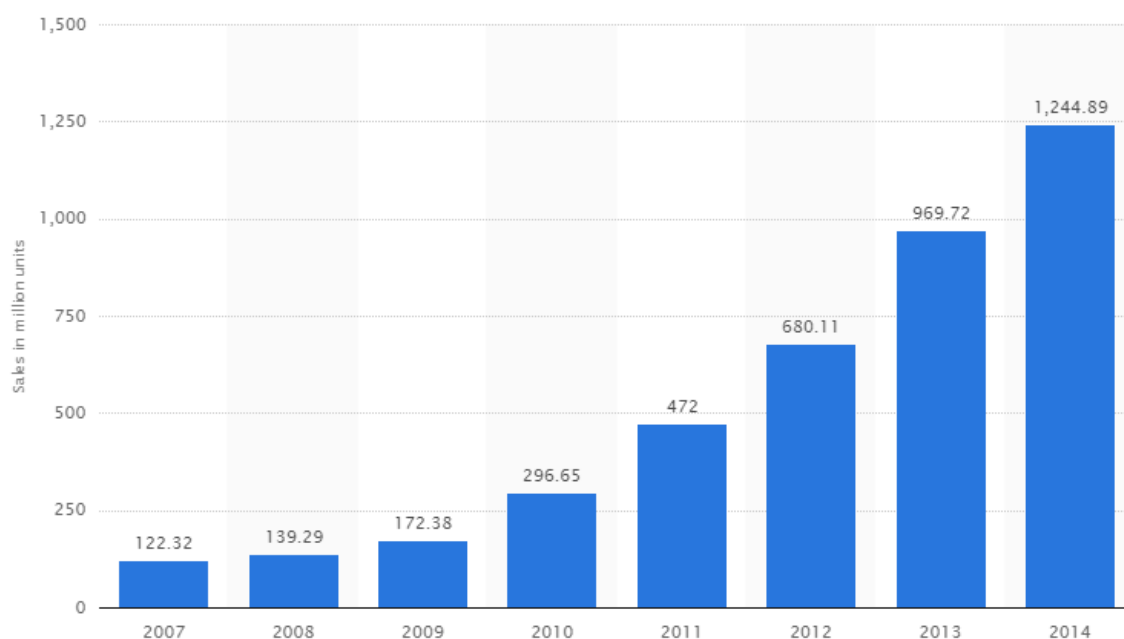
Kuva 3. Pienen ja suuren resoluution sivun koon ero (Podjarny 2013)

Toinen kritiikin aihe on monimutkaisuus. Responsiiviset sivut ovat luonnostaan monimutkaisia kehittää sillä ne pyrkivät tukemaan montaa erilaista alustaa ja resoluutiota ilman optimointia jollekin tietylle laitteelle tai laitetypille. Hyvän toteutuksen tekeminen on mahdollista, mutta edellyttää monimutkaista javascript-ohjelmointiskriptejä. Suoraan älypuhelimella tarkoitetun sivuston tekeminen on yksinkertaisempaa ja lopputuloksesta tulee myös nopeampi. (Chan 2013.)

Kolmas kritiikin aihe on käyttökokemus(UX). Mobiili on täysin erilainen käyttökokemus kuin pöytäkone, joten vain yhden nettisivun tekeminen vaikka se olisi responsiivinen saattaa haitata käyttökokemusta molemmilla alustoilla. Pyrkimys täyttää molempien kohdeyhmien käyttäjien tarpeet saattaa johtaa siihen ettei kummankaan käyttökokemus ole parhain mahdollinen. (Ghazarian 2014.)

2.3 Teknologinen tausta

Responsiivinen suunnittelun taustalla on mobiili-internet ja erikokoisten laitteiden nopea yleistyminen. Älypuhelimet ovat yleistyneet nopeasti muutaman viime vuoden aikana ja myynti ylitti vuonna 2014 miljardin kappaleen rajan (kuva 4). Älypuhelimet ja tabletit ovat niin yleisiä, että ne täytyy ottaa huomioon sivustoja suunnitellessa. Toisaalta ei ole enää mahdollista suunnitella jokaiselle laitetypille ja näytölle omaa versiotaan. (Knight 2011; Gartner 2015.)



Kuva 4. Älypuhelimien myynti loppukäyttäjille (miljoonia kappaleita) (Statista 2015)

2.4 HTML5

HTML5 on viides tarkastettu laitos ja uusin versio HTML-standardista. HTML5:en kehitys oli noin 15 vuotta kestävä, pitkä ja monimutkainen prosessi, virallinen standardi julkaistiin joulukuussa 2014. Se sisältää monipuolisen media-tuen, mutta myös enemmän tukea web applikaatioiden luomiseen jotka voivat olla vuorovaikutuksessa käyttäjän ja hänen tietojensa kanssa. (Bright 2014; Mozilla Developer Network 2015b.)

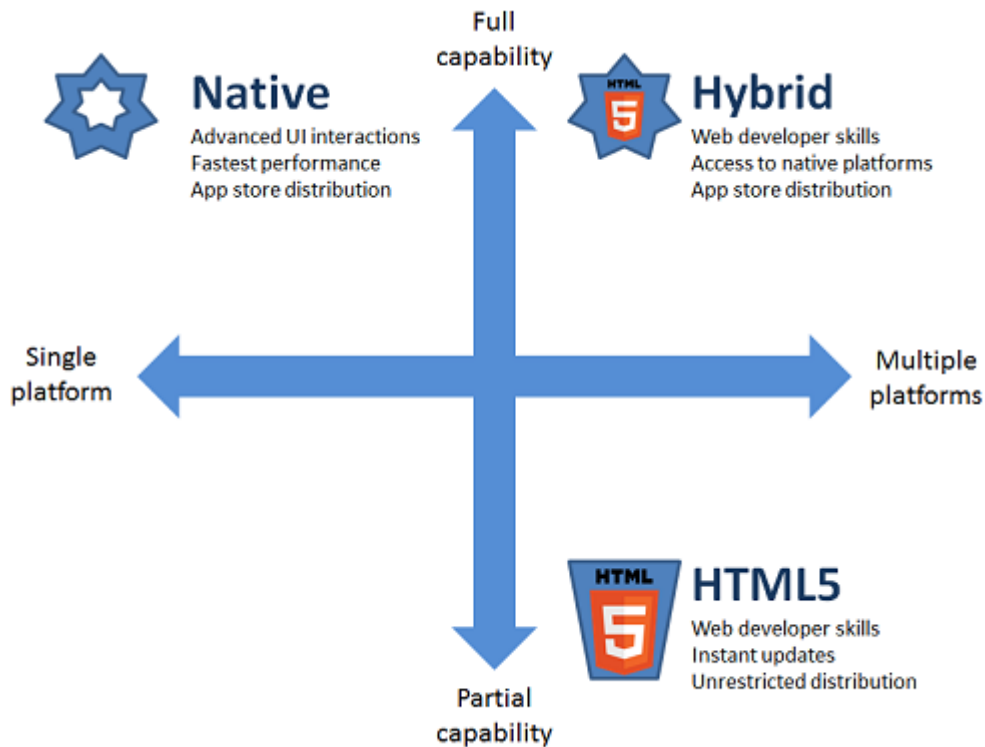
HTML4:än vuonna 1997 tapahtuneen julkaisun ja HTML 4.1:en vuonna 1999 tapahtuneen julkaisun jälkeen W3C (World Wide Web Consortium) ryhtyi kehittämään XML pohjaista XHTML:ää. XHTML sisälsi lukuisia ominaisuuksia XML-kielestä kuten tarkemman validoinnin syntaksille. Lisäksi XHTML:än tarkoituksena oli olla modulaarinen joka mahdollistaisi useiden tarkempien alistandardien julkaisun. (Bright 2014; W3C 2014.)

XHTML ei kuitenkaan ollut kovin yhteensopiva kehittäjien käytännön tarpeiden kannalta. Useat web-sivut eivät noudata standardeja ja monet kehittäjät kokivatkin että W3C:en kehityssuunta ei ollut oikea. HTML:stä haluttiin parempi alusta applikaatioille ja älypuhelimien ja tablettien kehityksen ja yleistymisen myötä tästä tuli nopeasti hyvin keskeinen tarve. (Bright 2014.)

Ryhmä yrityksiä (mm. Apple, Mozilla ja Opera) perustivat vuonna 2004 oman erillisen HTML:n kehitykseen tähänneen ryhmän nimeltä WHATWG (Web Hypertext Application Technology Working Group). WHATWG teki suuren osan HTML5:en alkuperäisestä kehitystyöstä, W3C otti WHATWG:n HTML5-luonnoksen lähtökohdakseen vuonna 2007. W3C ja WHATWG ovat tulleet yhteisymmärrykseen HTML5:en ydinominaisuuksista ja toiminnallisuuksista. Molemmilla tahoilla on silti yhä oma käsityksensä web standardien kehityksestä ja sen metodeista, sekä paljon vaikutusvaltaa ja taustatukea alan eri tahoilta. Tämä (Bright 2014.)

2.5 Natiivi ja hybridi vaihtoehtoina

Mobiilisovellukset voidaan jakaa karkeasti kolmeen ryhmään: natiivi, hybridi ja web soveluksiin (kuva 5). Natiivi-sovellus mahdollistaa laitteen ja käyttöjärjestelmän kaikkien ominaisuuksien käyttämisen. Natiivi-sovellus on tehokkain vaihtoehto, sillä se mahdollistaa optimoinnin jokaiselle alustalle erikseen. Web-sovelluksen rakentaminen mahdollistaa alustariippumattomuuden, joka voi huomattavasti pudottaa kehityskustannuksia ja kehitykseen kuluvaa aikaa. Hybridi-sovelluksen tarkoituksena on yhdistää hyviä puolia molemmista käyttämällä yhteistä ohjelmakoodia. (Rudolph 2014.)



Kuva 5. Vertailu natiivin, hybridin ja HTML5:en ominaisuuksista (Korf & Oksman 2012)

2.5.1 Natiivi

Natiivin sovelluksen etuja ovat monimutkaisten puhelimien ominaisuudet kuten kameran, skannauksen, GPS:än käyttö. Lisäksi natiivi-sovelluksen yksilöiminen ja räätälöiminen kyseiselle laitteelle ja alustalle on mahdollista hyvin laajasti. Yksilöllisen käyttöliittymän ja käyttöliittymäkomponenttien rakentaminen on myös natiivin sovelluksen hyvä puoli. Käyttökokemus ja käyttäjän toimintaan reagoiminen on natiivilla sovelluksella mahdollista tehdä paremmin ja nopeammaksi kuin muilla lähestymistavoilla. (O'Dell 2013.)

2.5.2 Hybridi

Hybridi-sovelluksessa HTML, CSS ja Javascript pyörivät sisäänrakennetussa selaimessa (WebView) natiivi-sovelluksen sisällä. Hybridi-kehityksen hyötyjä ovat saman koodin käyttäminen usealla alustalla, näin kuluu vähemmän aikaa ja kustannuksia kehittämiseen (kuva 6). Lisäksi web-kehittäjät voivat käyttää osaamistaan web-sovelluksien tekniikoista ja hyödyntää responsiivisuudesta tuttuja käyttöliittymä elementtejä ja osaamista. Web-sovelluksista poiketen hybridi-sovellukset voivat käyttää joitakin laitteen ja käyttöjärjestelmän ominaisuuksia. Hybridi-sovelluksilla on lisäksi hyvät offline-toiminnallisuudet, ne eivät edellytä verkkoyhteyttä kaikkiin toiminnallisuuksiin oikein rakennettuina. Lisäksi sovelluksia voidaan myydä ja levittää suosittujen sovellusmyymälöiden avulla (App Store ja Google Play). (Rudolph 2014.)

BENEFITS

- Developer can use existing web skills
- One code base for multiple platforms
- Reduced development time and cost
- Easily design for various form factors (including tablets) using responsive web design
- Access to some device and operating system features
- Advanced offline capabilities
- Increased visibility because the app can be distributed natively (via app stores) and to mobile browsers (via search engines)

DRAWBACKS

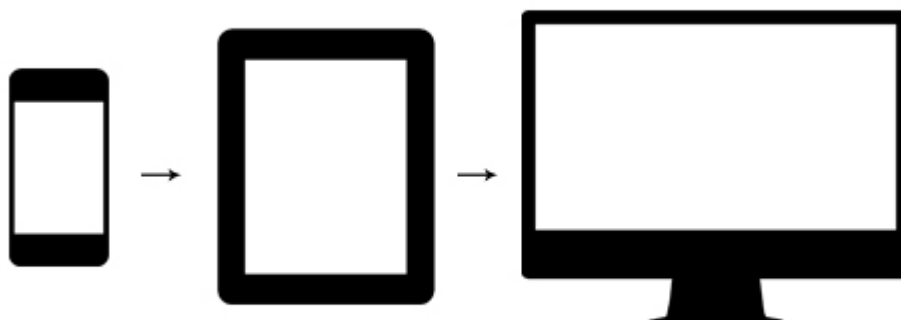
- Performance issues for certain types of apps (ones relying on complex native functionality or heavy transitions, such as 3D games)
- Increased time and effort required to mimic a native UI and feel
- Not all device and operating system features supported
- Risk of being rejected by Apple if app does not feel native enough (for example, a simple website)

Kuva 6. Hybridi sovelluksen edut ja haitat (Rudolph 2014)

Haittapuolina hybridi-sovelluksilla on tehokkuus ja suorituskyky ongelmat tietyn tyyppisillä raskailla sovellustyypeillä, sellaisilla jotka nojaavat vahvasti raskaisiin natiivi-operaatioihin kuten 3D-pelit. Lisäksi käyttöliittymän saaminen muistuttamaan natiivia sovellusta verkkosivun sijaan vaatii paljon kehitystyötä. Keskeinen haittapuoli on myös, että vain osa laitteiden ja käyttöjärjestelmien omista ja niille yksilöllisistä ominaisuuksista on käytettävissä. Haittapuolien määrän vuoksi tuleekin miettiä tarkasti, mikä on sovelluksen lähtökohta ja tavoite sillä hybridi lähestymistapa soveltuu huomattavasti paremmin tietyn tyyliin ohjelmiin. (Rudolph 2014; Lehdonvirta & Korpela 2013, 20-21.)

2.6 Mobile first

Perinteisesti suurin osa web-kehittäjistä on lähestynyt pöytäkoneille suunniteltua versiota ensin, jättäen mobiiliversion toissijaiseksi tavoitteeksi. Jopa responsiivisen suunnittelun yleistyessä monet aloittavat suurelle näytölle suunnittelulla ja etenevät tästä pienempiin versioihin. Mobile first kääntää tämän ajattelun päinvastoin (kuva 6). (Johnson 2013.)



Kuva 7. Kuinka mobile first suunnittelu etenee (Johnson 2013)

Käytetään joustavaa ruudukkoa, joustavia kuvia ja mediakyselyitä jotta saadaan asetteluun toimimaan hyvin riippumatta laitteen ominaisuuksista. Mobile first ajattelu on laajempi kuin pelkkä näytön kokoon mukautuminen. Suunniteltaessa mobiilille tulee ottaa huomioon myös mikä sisältö on tärkeintä ja kuinka sivun nopeus optimoidaan. (Frost 2012.)

2.7 CSS3 ja media kysely

CSS3 on viimeisin standardi CSS:lle ja se on taaksepäin yhteensopiva aiempien CSS-versioiden kanssa. CSS3 lisää useita uusia ominaisuuksia aiempiin versioihin, responsiivisuuden kannalta näistä tärkeä on mediakysely. (Mozilla Developer Network 2015a; W3Schools 2015b.)

Mediakyselyjä käytetään määrittämään erilaisia tyylisääntöjä erilaisille media- ja laitetyypeille. Mediakyselyt tarkastavat laitteen ominaisuuksia ja niitä voidaan käyttää tarkistamaan monia asioita kuten: selaimen ikkunan leveyttä ja korkeutta, laitteen näytön leveyttä ja korkeutta, laitteen näytön suuntaa (orientation) ja resoluutiota. (W3Schools 2014a.)

```
@media (min-width: 700px) { ... }
```

Kuva 8. Mediakysely esimerkki (Mozilla Developer Network 2015c)

Mediakyselyt mahdollistavat CSS:n soveltamisen silloin kun tietyt ehdot täyttyvät. On mahdollista kirjoittaa esimerkiksi mediakysely, joka soveltaa tiettyjä CSS-sääntöjä jos se-

lain on vähintään tietyn levyinen. Esimerkissä (kuva 8) sovelletaan tiettyjä tyylitietoja jos selain on vähintään 700 pikseliä leveä. (Pettit 2014a.)

2.8 Sovelluskehukset

Sovelluskehys on alusta, jonka avulla web-käyttöliittymä rakennetaan. Sovelluskehukset koostuvat valmiista komponenteista, joita kehittäjät voivat käyttää työskennellessään web-projektien parissa. Komponentteja voidaan myös muokata jonkin verran, mutta se sisältää omat riskinsä siitä kuinka komponentit toimivat jatkossa muiden sovelluskehysten osien kanssa. (Wilczynski 2014.)

Sovelluskehukset koostuvat kolmesta osasta: HTML, CSS ja Javascript. HTML määrittelee sivun rakennetta ja kuinka se on järjestetty sivulla. CSS sisältää sivuston tyyliohjeet, kuinka sivu visuaalisesti muotoillaan. Javascript-koodia käytetään sisällyttämään dynaamisia elementtejä ja toiminnallisuutta. (Wilczynski 2014.)

3 Tutkimuskysymykset

Tässä luvussa käydään läpi tutkimuskysymykset, opinnäytetyöhön valittu menetelmä. Lisäksi yleisellä tasolla käydään läpi tärkeimpiä lähteitä ja mihin niitä on opinnäytetyössä käytetty.

3.1 Menetelmä ja lähteet

Opinnäytetyön toteutuksessa käytettiin menetelmänä kirjallisuuskatsausta. Aiheena responsiivisuus on suhteellisen uusi ja ajankohtainen aihe. Aihe sekä siihen liittyvät käsitteet muuttuvat ja tarkentuvat jatkuvasti, ja niitä käsitellään paljon englanninkielisissä web-pohjaisissa julkaisuissa. Kirjallisuuskatsauksella on saatavissa ajantasaista ja kattavaa tietoa sovelluskehysten ominaisuuksista.

Tärkeimpinä lähteinä työn teoriataustan puolelta toimivat verkkojulkaisut Smashing Magazine ja A List Apart, joissa molemmissa on kattavia ja laajoja artikkeleita responsiivisuudesta ja siihen liittyvistä käsitteistä. Nämä lehdet on suunniteltu ammattilaisia silmällä pitäen ja artikkelit ovat teknisesti hyvin perusteltuja. Työn kieliasu ja käsitteiden suomenokset perustuvat kirjaan HTML5 sovellusalustana, kirja sisältää laadukkaat yleisesitykset myös monista käsitteistä ja näin ollen sitä myös käytetään teorian lähteenä.

Varsinainen tutkimus ja siihen käytetyt lähteinä toimivat blogijulkaisut, joissa pyritään yleensä esittämään sovelluskehysten hyviä ja huonoja puolia vertailemalla niitä keskenään. Blogipostauksista tärkeimpiä ovat Vermilion, jonka vertailussa on kattava taulukko vertailtavien kehysten ominaisuuksista josta saa hyvän yleiskuvan. Tärkeimpiä kirjallisia lähteitä ovat Design Modo, Hamilton ja Pettit, joissa esitetään mielipiteitä ja kokemuksia puolesta ja vastaan.

3.2 Tutkimuskysymykset

Minkäkokoiisiin projekteihin sovelluskehys sopii?

Kuinka kattavasti sovelluskehyksessä on ominaisuuksia?

Onko ominaisuuksien muokkaaminen yksilölliseksi helppoa?

Keskeinen tavoite opinnäytetyössä oli saada lisää tietoa siitä, kuinka isoihin projekteihin sovelluskehukset sopivat. Tämän aiheen kannalta tärkeä on ensimmäinen tutkimuskysymys ”Minkäkokoiisiin projekteihin sovelluskehys sopii?”, joka yleisellä tasolla kartoittaa kehittäjien kokemuksia tästä ominaisuudesta.

Toinen keskeinen tavoite oli selvittää sovelluskehysten mukana tulevia ominaisuuksia. Tämän selvittämiseen tähtää toinen tutkimuskysymys ”Kuinka kattavasti sovelluskehystessä on ominaisuuksia?”. Toinen kysymys liittyy ensimmäiseen, sillä se mahdollistaa tietää tarkemmin kuinka sovelluskehys soveltuu laajempiin projekteihin. Ominaisuuksien määrä on kriittinen asia sovelluskehysten valintaa mietittäessä. Sillä laajaan kaupalliseen projektiin tarvitaan paljon ja monipuolisesti erilaisia valmiiksi sovelluskehysten mukana tulevia ominaisuuksia.

Kolmas tutkimuskysymys on ”Onko ominaisuuksien muokkaaminen yksilölliseksi helpoa?”. Sovelluskehysten tarjoavat alustan, jota yleensä muokataan yksilölliseksi jotta saadaan visuaalinen ilme tavoitellun kaltaiseksi. Kolmannen tutkimuskysymyksen tavoitteena on selvittää, minkälaisia kokemuksia kehittäjillä on sovelluskehysten muokkauksesta omien tarpeidensa mukaiseksi.

4 Kehittäjien kokemukset

Vertailun kohteena on kolme sovelluskehystä Skeleton, Bootstrap ja Foundation. Tässä luvussa käydään läpi näiden kehysten keskeisimpiä toiminnallisuuksia. Kehyksistä löytyy huomattavan paljon ominaisuuksia (liite 1) ja niihin on saatavilla myös kolmansien osapuolien laajennuksia. Esittelyssä ovat vain yleisimmät perusominaisuudet kustakin alustasta.

Sovelluskehysten näkyvin osa on rakenne (grid). Se on ydintoiminnallisuus joka mahdollistaa suunnittelijoiden ja kehittäjien tehdä prototyyppisiä sivuston asettelusta. Lisäksi se mahdollistaa kehittäjien tehdä nopeita muutoksia varmistuen, että asettelu toimii jatkossakin. (Pettit 2014b.)

4.1 Bootstrap

Bootstrap on ilmainen avoimen lähdekoodin kokoelma työkaluja joilla voi luoda web-sivuja ja web-applikaatioita. Alun perin Twitter Bootstrap nimellä tunnettu sovelluskehys kehitettiin Twitterin sisäiseen käyttöön. Alkuperäiset kehittäjät olivat Mark Otto ja Jacob Thornton, mutta tällä hetkellä Bootstrap on todella laaja projekti ja siihen osallistuu lukuisia kehittäjiä. (Wikipedia 2015a.)

4.1.1 Mediakysely

Mediakyselyillä päätetään kuinka leveälle laitteelle muotoilu pätee. Bootstrapissa on valmiiksi tehty neljä luokkaa joiden avulla mediakyselyt voidaan kohdistaa erikokoisille laitteille. Mediakyselyjä käytetään yleensä yhteistyössä ruudukon asettelun kanssa. Nämä neljä luokkaa ovat alla listattuna kohdennettujen pikseleiden määrän kanssa. (Bootstrap 2015.)

- puhelimet, alle 768 pikseliä
- tabletit, yli tai yhtä suuri kuin 768 pikseliä
- pöytäkoneet, yli tai yhtä suuri kuin 992 pikseliä
- suuret pöytäkoneet, yli tai yhtä suuri kuin 1200 pikseliä

4.1.2 Joustava ruudukko

Bootstrap sisältää joustavan ruudukon, joka jakautuu kahteentoista osaan (kuva 9). Ruudukon voi jakaa haluamallaan tavalla, yhdelle rivillä ei pitäisi kuitenkaan olla yli kahtatoista osaa. Tässä tapauksessa rajoituksen yli menevä osa siirtyy seuraavalle riville. Ruudukon luokkia voidaan myös käyttää yhteistyössä mediakyselyjen kanssa ja kohdistaa erilainen

ruudukon rakenne näytön koon mukaan. Tällöin samalle div-elementille annetaan useampi CSS-luokka, jossa määritellään sekä kuinka suuri osa ruudukosta tälle elementille kuuluu sekä minkä kokoisille näytöille. Voidaan esimerkiksi laittaa sama elementti pienillä näytöillä koko näytön levyiseksi ja keskikokoisilla näytöille täyttämään kaksi kolmasosaa näytöstä, näin sivuston rakenne on muutettavissa laitteen mukaan mukautuvaksi. (Bootstrap 2015.)

.col-md-1	.col-md-1	.col-md-1	.col-md-1	.col-md-1	.col-md-1	.col-md-1	.col-md-1	.col-md-1	.col-md-1	.col-md-1	.col-md-1
.col-md-8									.col-md-4		
.col-md-4				.col-md-4				.col-md-4			
.col-md-6						.col-md-6					

Kuva 9. Bootstrapin joustava ruudukko (Bootstrap 2015)

4.1.3 Typografia

Bootstrapissa on olemassa tyylit kaikille kuudelle erikokoiselle otsikolle. Lisäksi on saatavilla CSS-luokat h1 – h6, joilla otsikkojen fontin ja koon voi siirtää tavalliseen tekstiin span-elementin avulla, saadakseen otsikon, joka ei ole omalla rivillään vaan tekstin seassa. Otsikkojen lisäksi voi luoda otsikon viereen toisen kevyemmän vaalean harmaana värjätyn pienemmän toissijaisen tai tarkentavan tekstin (kuva 10). Kevyemmän alatekstin voi luoda joko HTML small-tagilla tai CSS small-luokalla. (Bootstrap 2015.)

h1. Bootstrap heading Secondary text

h2. Bootstrap heading Secondary text

h3. Bootstrap heading Secondary text

h4. Bootstrap heading Secondary text

h5. Bootstrap heading Secondary text

h6. Bootstrap heading Secondary text

Kuva 10. Bootstrapin otsikot ja small-tekstit (Bootstrap 2015)

Yleinen perusfontin koko on Bootstrapissa 14 pikseliä ja rivin korkeus 1.428. Tämä fontti pätee body-elementtiin ja kaikkiin kappaleisiin (p-elementti). Lisäksi kappaleiden alle tulee marginaali, joka on puolet rivin korkeudesta eli oletuksena 10 pikseliä. Kappaleisiin voi myös luoda ensimmäisen kappaleen, joka on suurempi kuin muut kappaleet käyttämällä - CSS:n lead-luokkaa. Tekstistä voi korostaa (highlight) kappaleita käyttämällä HTML-tagia mark. (Bootstrap 2015.)

Tekstistä voi näyttää että osa on poistettu käyttämällä HTML-tagia del, jolloin teksti yliviivataan. Lisäksi on olemassa tavallinen yliviivaus HTML-tagilla s. Halutessaan näyttää että teksti on lisätty hiljattain voidaan käyttää HTML-tagia ins, jolloin teksti alleviivataan. Tekstiin on olemassa myös kursivoinnille ja lihavoinnille omat tyyliinsä, kursivointi i-tagilla ja lihavointi b-tagilla. Bootstrapissa on vielä lisäksi omat valmiit tagit ja tyyli lyhenteille, osoitteille, tekstin tasaamiselle (vasen, keskellä, oikea), tekstin muuttaminen isoille tai pienille kirjaimille, alkukirjaimille ja sitaateille. (Bootstrap 2015.)

4.1.4 Painikkeet

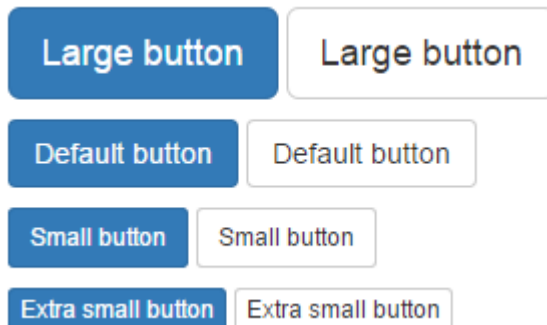
Painike CSS-luokkia voidaan käyttää a, button ja input elementtien kanssa. Button-elementin käyttö on suositeltua, koska sillä on paras tuki erilaisten selaimien kanssa. Lisäksi menu elementtien nav ja navbar kanssa toimii ainoastaan button-elementti. Käytettäessä a-elementtiä painikkeena tulee sille myös määritellä rooliksi button (role="button"). Painikkeille on olemassa semanttisesti erilaisia tarkoituksia, jotka näkyvät erilaisina tyyleinä. Olemassa olevat luokat ja painikkeiden värit ovat listattuna (kuva 11). (Bootstrap 2015.)

- btn-default, oletustyyli
- btn-primary, tärkeä painike
- btn-success, onnistuminen
- btn-info, tiedoksi
- btn-warning, varoitus
- btn-danger, vaara



Kuva 11. Bootstrapin painikkeet (Bootstrap 2015)

Painikkeiden tyylejä sovellettaessa kerrostetaan CSS-luokkia, jokaiseen painikkeeseen tulee btn-luokka ja lisäksi mahdollisia lisäluokkia. Luokat syötetään HTML-elementtiin peräkkäin, btn tulee ensimmäisenä ja tämän jälkeen tarkentavia esimerkiksi painikkeen tyyppi edellä mainituista vaihtoehdoista. Painikkeisiin on olemassa myös eri kokoja (kuva 12), näitä sovelletaan CSS-luokilla: suuri (btn-lg), pieni (btn-sm) ja hyvin pieni (btn-xs). Peruskokoinen painike sijoittuu suuren ja pienin painikkeen välille, mutta sen käyttämiseen ei tarvitse laittaa erikseen CSS-luokkaa vaan sitä käytetään oletuksena. (Bootstrap 2015.)



Kuva 12. Painikkeiden eri koot (Bootstrap 2015)

Painikkeet näyttävät alas painetuilta tummemmalla taustalla. Button-elementeille tämä toteutetaan active pseudo-luokalla, sama vaikutelma voidaan tehdä myös lisäämälle CSS-luokka active painikkeeseen. Laittamalla painikkeen disabled attribuutti tilaan disabled painike muuttuu haaleammaksi, osoittamaan ettei se ole aktiivinen. (Bootstrap 2015.)

4.1.5 Lomakkeet

Lomakkeissa kaikkiin yksittäisiin lomakkeen elementteihin sovelletaan Bootstrapin valmiita CSS-tyylejä. Kaikki tekstiä sisältävät elementit: input, textarea ja select laitetaan noudat-

tamaan 100 prosentin leveyttä suhteessa ylempään (parent) elementtiin käyttämällä niihin form-control CSS-luokkaa (kuva 13). Nämä tietojen syöttökentät ja niihin kuuluvat label-elementit tulee laittaa form-group div-elementteihin jolloin tasaukset ja elementtien sijoittelu toimii oikein. Hierarkia menee siis ensin form ja tämän alle lukuisia form-control div-elementtejä ja mikäli käytössä on erikoistyylliset input-group-elementit, menevät nämä form-control elementin sisään. (Bootstrap 2015.)

Email address

Password

File input

 Ei valittua tiedostoa

Example block-level help text here.

Kuva 13. Bootstrapin oletus lomaketyyli (Bootstrap 2015)

Oletuslomakkeessa syöttökentät ja painikkeet on laitettu päällekkäin, mutta kontrollit voi laittaa myös riviin käyttämällä inline-form CSS-luokkaa (kuva 14). Elementin johon inline-form luokkaa käytetään ei tarvitse olla form, mutta jotta riviin asettaminen toimii tulee näytön leveyden olla vähintään 768 pikseliä. Leveyksiä voi joutua muuttamaan alemmille elementeille muuttamalla CSS width-attribuutti arvoon auto, jotta 100 prosentin arvo ei riko riviin laitettavaa lomaketta. (Bootstrap 2015.)

Name **Email**

Kuva 14. Bootstrapin rivissä oleva lomake (Bootstrap 2015)

Kahden edellä mainitun lomaketyylin lisäksi Bootstrapissa on saatavana horisontaalin lomake (kuva 15). Tässä lomaketyylissä sovelletaan Bootstrapin joustavan ruudun tyyliä lomakkeeseen horisontaalisen asettelun saavuttamiseksi. Form-elementtiin liitetään form-horizontal CSS-luokka, tämä muuttaa form-group div-elementtien muotoilua. Tämän jälkeen form-group elementit käyttäytyvät kuin ruudun row-luokkaa olisi sovellettu niihin, muodostamalla form-groupeista rivejä. (Bootstrap 2015.)

Email

Password

Remember me

Kuva 15. Horizontal-form lomake (Bootstrap 2015)

4.1.6 Listat

Bootstrapissa on oletustyyli yleisimmille erilaisille listatyypeille (kuva 16). Järjestämätöntä listaa merkitään ul-elementillä, järjestämättömän listan sisälle voi laittaa toisen listan ja nämä kaksi erotetaan toisistaan erilaisella tyyllillä. Järjestettyä listaa merkitään ol-elementillä, jolloin listan kohdat on merkitty numeroilla. Näissä kahdessa listatyypissä on sisennys, mikäli tämän haluaa jättää pois, voi käyttää list-unstyled CSS-luokkaa. List-unstyled poistaa sisennyksen ja merkit rivien alusta, mutta listan sisällä olevassa listassa on vastaavanlainen muotoilu kuin ul-elementillä tehdyssä listassa. Lisäksi on olemassa vielä list-inline CSS-luokka, jolla elementit järjestyvät vierekkäin samalle riville, tätä luokkaa käytetään yhteistyössä ul-elementin kanssa. (Bootstrap 2015.)

- | | |
|---|--|
| <ul style="list-style-type: none"> • Integer molestie lorem at massa • Facilisis in pretium nisl aliquet • Nulla volutpat aliquam velit <ul style="list-style-type: none"> ◦ Phasellus iaculis neque ◦ Purus sodales ultricies <p>Integer molestie lorem at massa</p> <p>Facilisis in pretium nisl aliquet</p> <p>Nulla volutpat aliquam velit</p> <ul style="list-style-type: none"> ◦ Phasellus iaculis neque ◦ Purus sodales ultricies | <ol style="list-style-type: none"> 1. Lorem ipsum dolor sit amet 2. Consectetur adipiscing elit 3. Integer molestie lorem at massa 4. Facilisis in pretium nisl aliquet 5. Nulla volutpat aliquam velit <p>Integer molestie lorem at massa</p> <p>Facilisis in pretium nisl aliquet</p> <p>Nulla volutpat aliquam velit</p> <p>Integer molestie lorem at massa</p> <p>Facilisis in pretium nisl aliquet</p> <p>Nulla volutpat aliquam velit</p> |
|---|--|

Kuva 16. Erilaiset listat ul, ol, unstyled ja inline (Bootstrap 2015)

4.1.7 Taulukot

Bootstrapissa on useita erilaisia taulukon asettelutapoja. Taulukon perusasettelun (kuva 17) käyttämiseksi table-elementtiin lisätään CSS-luokka table. Taulukkoja käytetään nykyään monenlaisissa yhteyksissä, tämän tähden taulukot pitää vielä erikseen määritellä CSS-luokalla joka on saman niminen kuin itse elementti. Perusasettelussa taulukossa on

suhteellisen kevyet välit elementtien välillä ja rivit on erotettu toisistaan vaakaviivoin. (Bootstrap 2015.)

#	First Name	Last Name	Username
1	Mark	Otto	@mdo
2	Jacob	Thornton	@fat
3	Larry	the Bird	@twitter

Kuva 17. Perustaulukko (Bootstrap 2015)

Table-luokan asetettua, voidaan taulukosta tehdä raidallinen ja värittää joka toinen rivi taulukosta lisäämälle table-striped-luokkaa taulukkoon. Tämä raidoitus pätee vain taulukon perusriveihin tbody-elementin alla, ei otsikoihin. Raidallisen tyylin lisäämistä ei tueta Internet Explorer 8-selaimessa, sillä raidoitus käyttää :nth-child CSS valitsinta jota Internet Explorer 8 ei tue. Table-bordered-luokan avulla voidaan lisätä koko taulukkoon ja sen kaikille riveille lisätä reunat. Table-hover-luokkaa käyttämällä taulukon aktiivinen rivi, jonka yllä kursori on vaihtaa taustaväriä ja korostaa aktiivisen rivin. Table-condensed luokan avulla voidaan tiivistää taulukon muotoilua, jolloin solujen sisällön ja reunojen väliin jäävä tila puolitetaan. (Bootstrap 2015.)

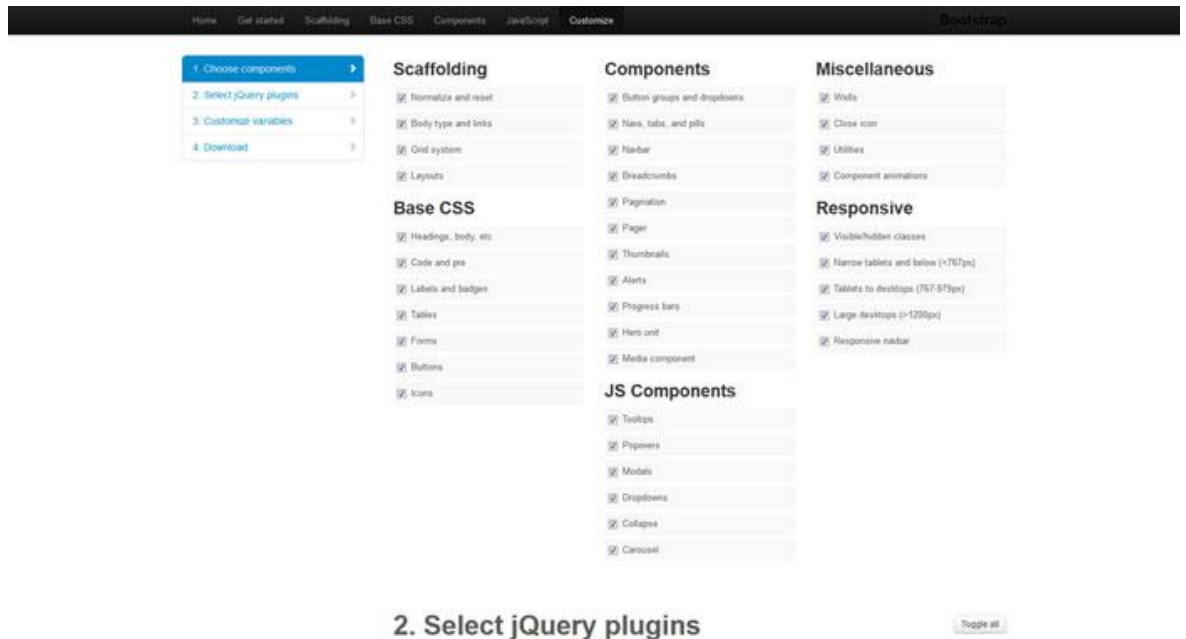
#	Table heading	Table heading	Table heading	Table heading	Table heading	Table heading
1	Table cell	Table cell	Table cell	Table cell	Table cell	Table cell
2	Table cell	Table cell	Table cell	Table cell	Table cell	Table cell
3	Table cell	Table cell	Table cell	Table cell	Table cell	Table cell

Kuva 18. Responsiivinen taulukko jossa lisäksi table-border (Bootstrap 2015)

Taulukoissa on myös mahdollista käyttää Bootstrapissa käytettyjä kontekstiin liittyviä luokkia: active, success, info, warning ja danger. Näitä luokkia käytetään useissa muissakin elementeissä kuten painikkeissa ja ne sisältävät vastaavat värit. Taulukosta responsiivisen voi tehdä käyttämällä table-responsive-luokkaa jota käytetään table-elementissä. Responsiivinen taulukko lisää vaakasuuntaisen scroll-toiminnon (kuva 18) kun selaimen leveys alittaa 768 pikseliä. (Bootstrap 2015.)

4.1.8 Positiiviset

Bootstrapin käytön tärkein positiivinen puoli on ajan ja resurssien säästö. Siinä on paljon valmiiksi mukana tulevia ominaisuuksia ja se tarjoaa kattavan sovelluskehityksen suoraan käyttöönotettavaksi. Lisäksi Bootstrap on niin suosittu, että sille on kehitetty useita kolmannen osapuolen kirjastoja ja lisätoiminnallisuuksia. Lisäksi voit valita mitä toiminnallisuuksia ja ominaisuuksia otat käyttöön Bootstrapista, voit ladata version jossa on vain sinun tarvitsemasi ominaisuudet (kuva 19). (Design Your Way 2015.)



Kuva 19. Bootstrapin ominaisuuksien valitseminen (Design Your Way 2015)

4.1.9 Kriittikkiä

Huonoista puolista Bootstrapissa yksi suurimmista on monimutkainen HTML-kielen käyttö, jossa käytetään lukuisia CSS-luokkia HTML-koodissa, tällöin web-sivustosta tulee suhteessa melko laaja ja HTML-koodia on paljon. Toinen keskeisistä huonoista puolista, joka osaltaan liittyy myös monimutkaisuuteen on se että Bootstrap on raskas, sillä se sisältää paljon CSS:ää ja Javascript-koodia. Bootstrapia ei ole suunniteltu latausaika mielessä. Kolmas keskeinen huono puoli Bootstrapin yleisyys joka johtaa siihen, että saman näköisiä ja saman oloisia käyttöliittymiä ja web-sivuja on paljon. Usein kehittäjät käyttävät Bootstrapia ilman suurta yksilöimistä, jolloin sivuston visuaalinen ilme voidaan kokea liian tavalliseksi. (Design Your Way 2015.)

4.2 Foundation

Foundation on ZURB nimisen yhtiön julkaisema responsiivinen sovelluskehys. Ensimmäinen versio julkaistiin vuonna Syyskuussa 2011. Foundation on julkaistu MIT-lisenssin, joten sitä voi käyttää myös suljetun lähdekoodin kaupallisiin ohjelmistoihin. (Wikipedia 2015b.)

4.2.1 Mediakysely

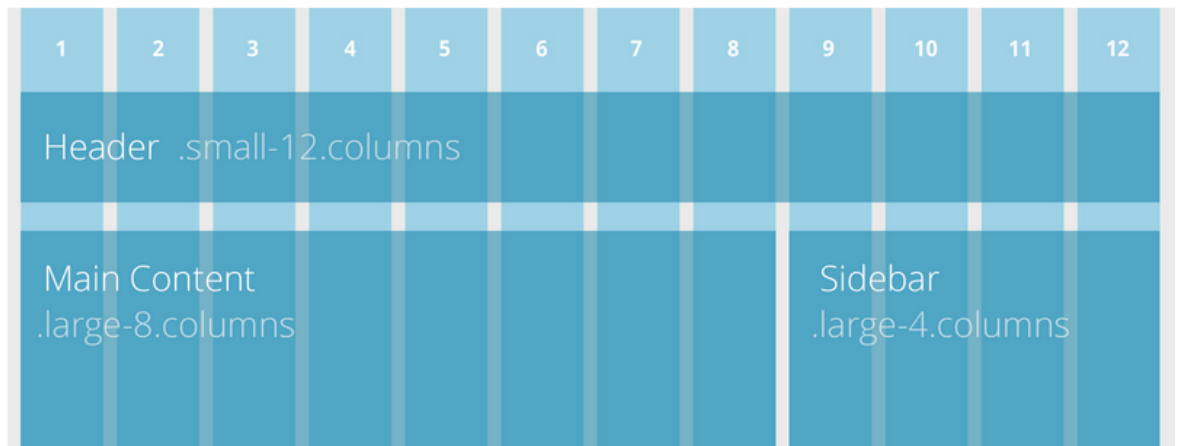
Mediakyselyt Foundationissa on pyritty pitämään suhteellisen pelkistettynä. Tarkoituksena on että prosentti mittayksiköihin perustuva joustava ruudukko hoitaa suurimmin osan muotoiluun liittyvästä työstä. Mediakyselyissä Foundation käyttää suhteellista em-mittayksikköä. Rem-mittayksikköä ei vielä käytetä sillä kaikki selaimet eivät tue sitä, siihen on tarkoitus kuitenkin siirtyä tulevaisuudessa. Foundation tukee Sass-tyylitiedostoja jotka mahdollistavat muuttujien käytön CSS-tiedostojen yhteydessä. Mediakyselyille on määritelty valmiiksi muuttujia erikokoisille näytöille, joita voi muuttaa mikäli käyttää Sass:ia projektissaan. Alla on listattu mediakyselyjen standardi koot. (Foundation 2015c.)

- pienet näytöt, alle 40 em eli maksimileveys 640 pikseliä
- keskisuuret näytöt, yli 40 em eli minimileveys 641 pikseliä
- suuret näytöt, yli 64,063 em eli minimileveys 1025 pikseliä
- erikoissuuret näytöt, yli 90,063 em eli minimileveys 1441 pikseliä
- hyvin suuret näytöt, yli 120,063 em eli minimileveys 1921 pikseliä

4.2.2 Joustava ruudukko

Foundationissa joustava ruudukko on toteutettu siten, että ensin luodaan vaakasuuntainen div-elementti, johon sijoitetaan row-luokka. Tästä div-elementistä tulee vaakasuuntainen laatikko, jonka sisälle asetetaan pystysuuntaiset sarakkeet (kuva 20). Pystysuuntaiset sarakkeet asetetaan lisäämällä div-elementtiin column-luokka tai columns-luokka. Näiden kahden luokan välillä ei ole toiminnallista eroavaisuutta. (Foundation 2015f.)

Joustavia ruudukko voi laittaa sisäkkäin haluamansa määrän. Sisäkkäin laitettavien ruudukoiden määrää ei ole rajoitettu. Sisäkkäin asettaminen tapahtuu laittamalla row-luokalla varustetun elementin column-luokalla varustetun elementin sisälle. (Foundation 2015f.)



Kuva 20. Foundationin joustava ruudukko (Foundation 2015e)

4.2.3 Typografia

Foundationissa on oletustyyliä kaikille tavallisilla otsikko-elementeille h1-h6. Lisäksi on olemassa kevyempiä, hieman vaaleammalla näkyviä aliotsikoiksi tarkoitettu otsikkoja. Näitä käytetään lisäämällä h-elementtiin subheader-luokka. Tämän lisäksi on olemassa pienempi vaalea otsikko, jota voidaan käyttää esimerkiksi yhteistyössä pääotsikon vierellä. Näitä pienempiä otsikoita voi käyttää sijoittamalla h-elementin sisällä olevan tekstin small-elementin sisälle. Näin on mahdollista rajata vain osa h-elementin sisällä olevasta otsikosta pienemmälle otsikkotyypille ja sijoittaa sekä pääotsikko, että tarkentava otsikko yhden elementin sisälle. (Foundation 2015d.)

h1. This is a very large header.

h2. This is a large header.

h3. This is a medium header.

h4. This is a moderate header.

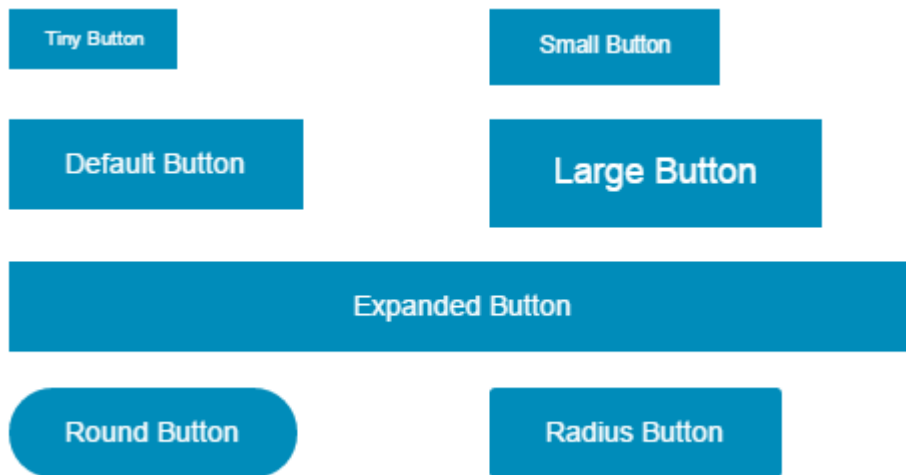
h5. This is a small header.

h6. This is a tiny header.

Kuva 21. Foundationin oletus otsikkotyylit

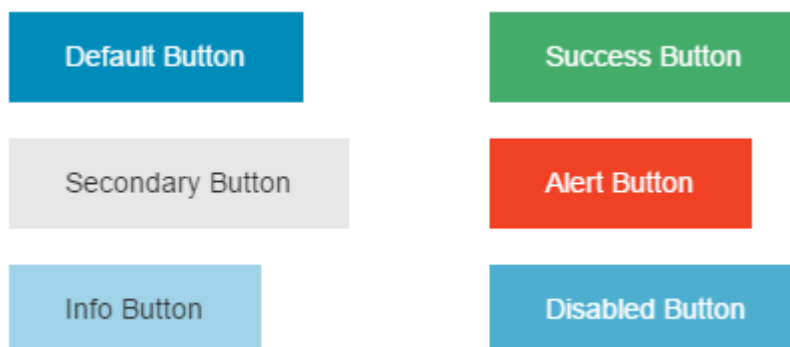
4.2.4 Painikkeet

Foundation sisältää button CSS-luokan jonka avulla HTML linkeistä eli a-elementeistä voidaan luoda painike. Erilaisia kokoja on useita kaikissa painikkeissa on CSS-luokan nimessä ensin button ja sitä seuraa tarkempi määre: tiny, small, large tai expand (kuva 22). Oletus painike on default ja sitä käytetään tavallisella button-luokalla ilman lisämääreitä. Eri kokoisten painikkeiden lisäksi on olemassa kaksi luokkaa jotka vaikuttavat painikkeen kulmien pyöreyyteen round (täysin pyöreät) ja radius (hieman pyöristetyt). Painikkeiden perustyyli on hyvin pelkistetty, jotta niiden päälle voi laittaa omia CSS-tyylejä ja luoda oman personoidun visuaalisen ilmeen mahdollisimman helposti. (Foundation 2015b.)



Kuva 22. Foundationin painikkeiden koot (Foundation 2015b)

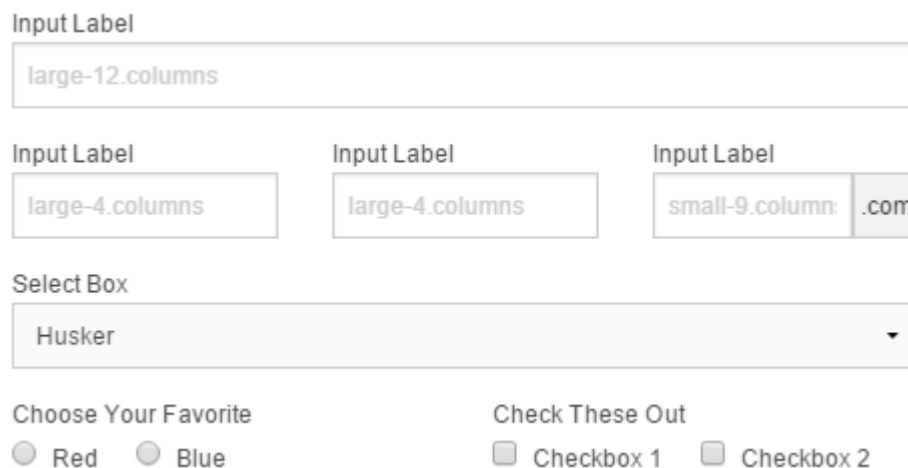
Painikkeilla on myös olemassa omia värejään: default, success, secondary, alert, info ja disabled (kuva 23). Niitä käytetään painikkeiden kokojen tapaan luokalla jonka nimessä on ensin button ja sen jälkeen tarkempi määrä. Värejä voi muuttaa muokkaamalla vastavaa CSS-luokkaa, väreillä on myös oma semanttinen merkityksensä ja niitä löytyy siis yleisimpiin tarkoituksiin. (Foundation 2015b.)



Kuva 23. Painikkeiden oletusvärit (Foundation 2015b)

4.2.5 Lomakkeet

Lomakkeiden luonnista on pyritty Foundationissa tekemään mahdollisimman joustavaa. Lomakkeet rakennetaan yhdistelmällä perinteisiä lomake-elementtejä ja Foundationin joustavan ruudukon rivi- ja sarake-elementtejä. Lomakkeiden elementtien tyylitiedot kohdistuvat elementteihin tyyppin perusteella, eivät CSS-luokkien perusteella. (Foundation 2015a.)



Kuva 24. Foundation lomake sarakeilla (Foundation 2015a)

Lomake-elementtien kokoa voidaan muokata muutamalla erilaisella tavalla. Input-elementtien kokoa voi muuttaa käyttämällä sarakkeiden kokoja. Toinen mahdollisuus on käyttää rivejä ja ryhmitellä lomakkeen elementtejä rivien sisälle. Yllä olevassa kuvassa on käytetty rivejä ja niiden sisällä myös sarakkeita (kuva 24). Foundationissa lomakkeiden oletustyyli on hyvin pelkistetty, sillä tarkoitus on että jokainen muokkaa niistä omalle sivustolleen tai projektilleen sopivat versiot. Rivit ja rivien sisällä olevat elementit perivät CSS padding-attribuuttia ylemmiltä luokilta jollain rivien elementit on valmiiksi tasattu sopivasti. (Foundation 2015a.)

4.2.6 Listat

Foundationissa on viisi erilaista li-elementtejä käyttävää listaa, jotka käsittävät yleisimmät listatyyppit. Tämän lisäksi on yksi dl-elementtiä käyttävä lista, jossa listattavan asian yhteydessä on myös kuvaus. Ul-elementtejä käyttäviä listoja on neljä, listojen nimet tulevat listattavien asioiden edessä esiintyvistä merkistä. (Foundation 2015d.)

- | | |
|--------------------|--------------------|
| • List item | ◦ List item |
| • List item | ◦ List item |
| • Nested list item | ◦ Nested List Item |
| • Nested list item | ◦ Nested List Item |
| ▪ List item | List item |
| ▪ List item | List item |
| ▪ Nested List Item | Nested List Item |
| ▪ Nested List Item | Nested List Item |

Kuva 25. Foundation ul-listatyypit: disc, circle, square ja no-bullet (Foundation 2015d)

Ensimmäinen ul-lista on CSS-luokalla disc, jolloin listattavien asioiden eteen tulee täysi ympyrä (kuva 25). Toinen ul-lista on circle, jolloin merkinä käytetään sisältä tyhjää ympyrää. Kolmas ul-lista on nimeltään square, jolloin listan elementtien alkuun tulee neliö. Neljäs ul-lista on no-bullet, jolloin elementeillä ei ole merkkiä edessään eikä niitä ole sisennetty. (Foundation 2015d.)

Järjestetty ol-lista on tyyliltään vastaava kuin kolme ensimmäistä ul-listaa, paitsi siinä on numero ennen jokaista elementtiä. D1-elementillä käytettävässä listassa on listan otsikko, jota seuraa tarkempi kuvaus haaleammalla tekstillä. (Foundation 2015d.)

4.2.7 Taulukot

Taulukon luonnissa Foundationissa käytetään HTML-kielen oletus taulukkorakennetta. Taulukko aloitetaan table-elementillä, jonka sisälle sijoitetaan thead ja tbody-elementit. Thead-elementin sisälle sijoitetaan taulukon otsikot ja tbody-elementin sisälle sijoitetaan taulukon tavalliset, sisältö sisältävät rivit. Foundation luo tälle perustaulukolle muotoilun (kuva 26) ja se toimii yhteistyössä joustavan ruudukon elementin kanssa. Taulukkoja tulisi käyttää vain tietojen esittämiseen, ei sivuston asetteluun luomiseen. (Foundation 2015g.)

Table Header	Table Header	Table Header	Table Header
Content Goes Here	This is longer content Donec id elit non mi porta gravida at eget metus.	Content Goes Here	Content Goes Here
Content Goes Here	This is longer Content Goes Here Donec id elit non mi porta gravida at eget metus.	Content Goes Here	Content Goes Here
Content Goes Here	This is longer Content Goes Here Donec id elit non mi porta gravida at eget metus.	Content Goes Here	Content Goes Here

Kuva 26. Foundationin oletus taulukkotyylit (Foundation 2015g)

4.2.8 Positiiviset

Positiivisia puolia Foundationissa on mobile first-filosofian mukaan toteutettu joustava ruudukko joka hyödyntää rem mittayksikköä pikseleiden sijaan. Toinen positiivinen asia Foundationissa on lähestymistapa, jossa valmiit käyttöliittymäkomponentit ovat hyvin pelkistettyjä ja kehittäjiä ja suunnittelijoita suositellaan muokkaamaan niistä omaan projektiin sopivia, näin lopputuloksista tulee hyvin yksilöllisiä. (Design Modo 2014.)

4.2.9 Kritiikkiä

Huonoja puolia Foundationissa on kehittäjäyhteisön koko. Vaikka sovelluskehityksellä on mahdollista tehdä laajoja kaupallisia projekteja, ei sen kehittäjäyhteisö ole kuitenkaan yhtä laaja kuin esimerkiksi Bootstrapilla, jolle löytyy paljon kolmansien osapuolien luomia lisäosia. Foundationin lisäosat ja toiminnallisuudet lepäävätkin vahvasti Foundationin kehittäneen Zurbin varassa. (Design Modo 2014.)

Foundationille on olemassa valmiita tyylejä ja teemoja joita voi ladata ja ostaa internetistä. Näitä ei kuitenkaan ole yhtä paljon kuin Bootstrapissa, johtuen levinneisyydestä ja näin ollen Foundationilla kehittäminen vaatii myös jonkin verran graafista osaamista. Vaikka Foundationin oletustyylit ovat pelkistetyt ja helppo käyttää, monen back end-kehittäjän on helpompi käyttää valmista teemaa ainakin pohjana, joten tässä mielessä Bootstrapilla on enemmän valinnan varaa. (Schenker 2014.)

Mikäli haluat kehittää sivuston mahdollisimman laajalla selaintuella, Foundation ei tue kaikkein vanhimpia selaimia kovin hyvin, sillä siinä on monia uusia graafisia ominaisuuksia ja elementtejä. Projektissa, jossa täytyy tukea myös Internet Explorer 8:aa, ei Foundationissa ole tukea tälle, toisin kuin Bootstrapissa. (Schenker 2014.)

4.3 Skeleton

Skeleton kevyt CSS sovelluskehys jonka on luonut Dave Gamache. Skeleton koostuu kahdesta CSS-tiedostosta normalize.css ja skeleton.css. Skeletonin tärkein ominaisuus on joustava ruudukko, lisäksi se tarjoaa tyylit yleisimmille HTML-elementeille. Skeleton rakentuu mobile first-idealle. (Martsoukos 2015.)

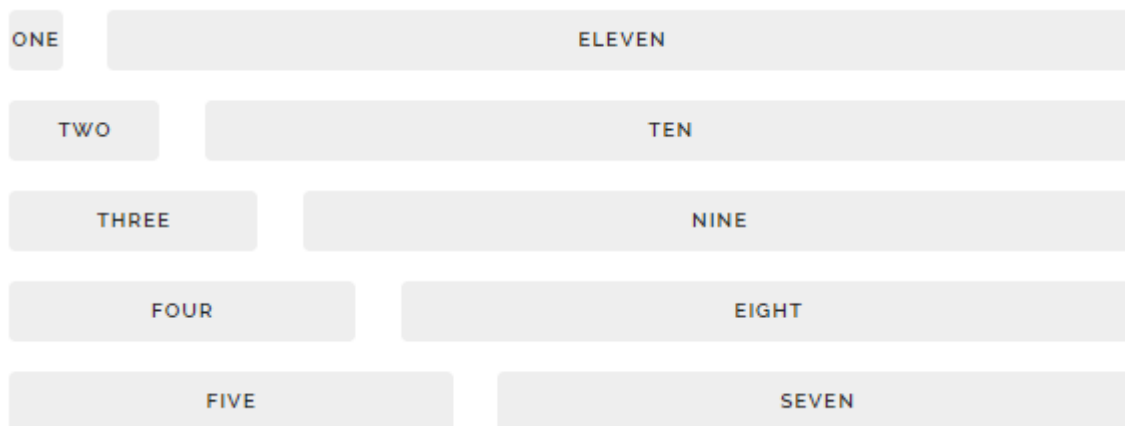
4.3.1 Mediakysely

Mediakyselyillä voidaan kohdistaa erilevyisille laitteelle muotoilu. Skeletonissa on valmiiksi tehty viisi luokkaa joiden avulla mediakyselyt voidaan kohdistaa erityyppisille laitteille. Skeletonissa mediakyselyt ovat mobile first eli ne kohdistuvat minimi leveyteen. Mediakyselyjä käytetään yleensä yhteistyössä ruudukon asettelun kanssa. Nämä viisi luokkaa ovat alla listattuna kohdennettujen pikselien määrän kanssa. (Skeleton 2015.)

- puhelimet 400 pikseliä
- phabletit 550 pikseliä
- tabletit 750 pikseliä
- pöytäkone 1000 pikseliä
- pöytäkone HD 1250 pikseliä

4.3.2 Joustava ruudukko

Skeletonin ruudukko on 12 saraketta leveä joustava ruudukko, jonka maksimileveys on 960 pikseliä. Maksimileveyttä voi muuttaa ja sarakkeet muuttuvat leveyttä tähän perustuen. Tavoitteena on ollut mahdollisimman kevyt syntaksi joustavan ruudukon osalta. Koko sivuston pääelementti on div-elementti johon laitetaan container-luokka. Tämän elementin sisälle sijoitetaan row-luokalla varustetut div-elementit, näiden sisälle column-luokalla varustetut sarakkeet (kuva 27). (Skeleton 2015.)



Kuva 27. Skeletonin joustava ruudukko esimerkkejä (Skeleton 2015)

4.3.3 Typografia

Tyyppi on määritelty rem-muotoon, joten fonttien koot ja suhteet voidaan määrittää yhdellä HTML-elementillä. Suoraan Skeleton ei muuta HTML-elementin fontin kokoa, mutta sitä on mahdollista mutta tarpeen vaatiessa. Typographian perusta on Raleway-fontti, jonka peruskoko on 15 rem (15 pikseliä) ja rivikoko 16 rem (24 pikseliä). Muita perustyyppejä kuten ankkurit (linkit), lihavoitu, kursivi ja alleiviivattu on myös olemassa (kuva 28). Otsikoita on olemassa kuutta eri kokoa h1 – h6, jokaisella oman kokoisensa fontti ja kirjaintenväli. (Skeleton 2015.)

[anchors](#), **strong**, *emphasis*, and underline

Heading `<h1>` 50rem

Heading `<h2>` 42rem

Heading `<h3>` 36rem

Heading `<h4>` 30rem

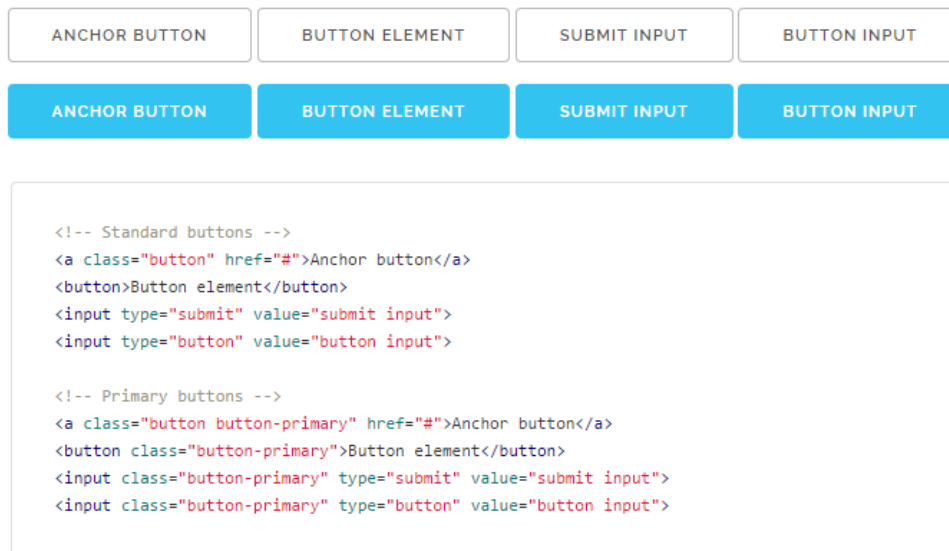
Heading `<h5>` 24rem

Heading `<h6>` 15rem

Kuva 28. Skeletonin eri tyyliset teksti ja otsikot (Skeleton 2015)

4.3.4 Painikkeet

Skeleton sisältää kahdentyyppisiä painikkeita. Standardi button-elementti on pelkistetty kun taas button-primary-luokalla määritelty elementti on voimakkaampi ja näkyvämpi (kuva 29). Painikkeiden tyyliä sovelletaan lukuisiin sopiviin lomake-elementteihin, mutta niitä voidaan soveltaa myös button-luokan avulla muihinkin elementteihin. (Skeleton 2015.)



Kuva 29. Skeletonin kaksi erilaista painiketta (Skeleton 2015)

4.3.5 Lomakkeet

Lomakkeet voivat olla normaalisti vaativia toteuttaa, mutta Skeletonissa niihin on olemassa valmiit tyylit. Kaikki syöte-kentät, valinta-kentät (select) ja painikkeet on normalisoitu yhteiselle korkeudella toimimaan eri selaimilla (kuva 30). Kun lomakkeiden eri elementit on kokonsa puolesta normalisoitu ja saman kokoisia niitä voidaan laittaa sekä päällekkäin, että toistensa viereen luontevasti. Lomakkeiden käytössä ei tarvita erillistä syntaksia, niihin ei tarvitse laittaa CSS-luokkia jotka erikseen määrittelisivät lomakkeen peruselementtien tyylin. Skeletonissa lomakkeiden peruselementit on valmiiksi muotoiltu Skeletonin tyyliin. (Skeleton 2015.)

Your email:

Reason for contacting:

Message:

Send a copy to yourself

Kuva 30. Skeletonin lomake perustyyli (Skeleton 2015)

4.3.6 Listat

Järjestämättömillä listoilla on perustyyli, ne käyttävät list-style CSS-attribuutissa ympyränmuotoista listan alkumerkkiä. Listat on myös sisennetty perustekstistä melko selvästi.

Myös järjestetyille listoille on oma tyyhinsä (kuva 31), ne käyttävät desimaali lista-tyyliä. Listoja voidaan myös laittaa toistensa sisälle haluamallaan tavalla. Järjestämättömän listan sisälle voi laittaa järjestetyn listan ja järjestetyn listan sisälle voi laittaa järjestämättömän listan. (Skeleton 2015.)

- | | |
|--|---|
| <ul style="list-style-type: none"> ◦ Unordered lists have basic styles ◦ They use the circle list style <ul style="list-style-type: none"> ◦ Nested lists styled to feel right ◦ Can nest either type of list into the other ◦ Just more list items mama san | <ol style="list-style-type: none"> 1. Ordered lists also have basic styles 2. They use the decimal list style <ul style="list-style-type: none"> ◦ Ordered and unordered can be nested ◦ Can nest either type of list into the other 3. Last list item just for the fun |
|--|---|

Kuva 31. Skeletonin kaksi erilaista listaa (Skeleton 2015)

4.3.7 Taulukot

Taulukkoja luotaessa tulee Skeletonissa käyttää HTML:n virallista taulukko-syntaksia, jossa table-elementin sisällä tulee laittaa molemmat, sekä thead ja tbody-elementit. Taulukoihin ei tarvitse soveltaa mitään taulukkokohtaista CSS-luokkaa. Myöskään yksittäisille taulukon elementeille ei tarvitse käyttää niille erikoisesti suunniteltuja luokkia, vaan Skeletonissa on tyyliä luotu valmiiksi taulukon HTML-peruselementeille. Taulukon otsikot on lihavoitu ja taulukon rivien tekstit noudattavat Skeletonin perustypografiaa (kuva 32). Taulukon rivit on eroteltu toisistaan rivien välissä olevin viivoin. (Skeleton 2015.)

Name	Age	Sex	Location
Dave Gamache	26	Male	San Francisco
Dwayne Johnson	42	Male	Hayward

Kuva 32. Skeleton taulukko (Skeleton 2015)

4.3.8 Positiiviset

Skeleton sisältää vähemmän valmiita komponentteja kuin useat muut sovelluskehikset, näin ollen sopii hyvin nopeasti käyttöönotettavaksi. Lisäksi Skeletonin graafiset komponentit ovat hyvin pelkistettyjä, niiden päälle on helppo lisätä omia CSS-tiedostojaan ja niiden muokkaaminen on yksinkertaista. Hyvä puoli on myös se, että Skeleton on helppo lisätä aloitettuun tai valmiiseen projektiin jälkikäteen, sillä se ei vaadi monimutkaisia muokkauksia. Yksinkertaisuutensa ja selkeytensä takia Skeletonia on helppo käyttää valmiiden omien komponenttien kanssa. (Hamilton 2013.)

4.3.9 Kritiikkiä

Huonoina puolina Skeletonissa on se, että se on hyvin kevyt ja pelkistetty eikä se sisällä yhtä kattavia ominaisuuksia ja yhtä paljon valmiita komponentteja kuin laajemmat sovel-
luskehukset. Siten Skeleton onkin enemmän vain lähtökohta joka tarjoaa joustavan ruu-
dukon, eikä se tarjoa perusteiden lisäksi paljon muuta. Näin ollen Skeletonilla kehittävän
on tehtävä itse tai hankittava muualta useat eri peruskomponentit. (Hamilton 2013.)

5 Pohdinta

Opinnäytetyössä selvitettiin kolmen responsiivisen sovelluskehityksen eroja vertailemalla niitä toisiinsa ja keräämällä kehittäjien kokemuksia kirjallisista lähteistä. Aihe on kohtuullisen uusi joten lähteinä käytin pitkälti web-pohjaisia julkaisuja ja artikkeleita. Julkaistuja kirjoja tai tieteellisiä artikkeleita on melko vähän saatavilla. Lähteiden määrän lisäksi ongelmallista oli niiden laatu, aiheesta kirjoitetut artikkelit vaihtelevat voimakkaasti laadukkaista kattavasti yleisesityksistä huomattavasti heikotasoisiin lähteisiin.

Heikommista lähteistä näkyy usein taustalla kirjoittajan oma tavoite tuoda esiin jotain tiettyä teknistä asiaa pelkästään positiivisessa valossa. Tilastoja aiheesta, esimerkiksi nopeudesta tai tietyn sovelluskehityksen pienestä koosta, saatettiin myös esittää aiheena olevan sovelluskehityksen eduksi. Artikkelin kirjoittaja oli mahdollisesti valinnut toisesta sovelluskehityksestä tarkoituksella laajempi tai aiempi versio vertailukohtaksi.

Lähteisiin liittyy myös sovelluskehysten yleisyys. Bootstrap on kolmesta käsitellyssä oleesta sovelluskehityksestä selvästi yleisin. Näin tästä löytyi huomattavasti helpommin lähteitä kuin kahdesta muusta sovelluskehityksestä. Skeleton on suhteellisen uusi ja selkeästi harvinaisin ja se esiintyi usein vertailuna kahden muun sovelluskehityksen kanssa. Pelkästään Skeletonia käsitteleviä lähteitä ja artikkeleita oli huomattavan vaikeaa löytää. Bootstrap sai yleisyytensä vuoksi usein kohtaansa kritiikkiä, sillä sen oletustyyliä käytetään hyvin paljon eri sivustoilla. Monen artikkelin lähtökohta oli tuoda vaihtoehtoja Bootstrapille ja siinä havaituille kehittämiskohteille

Opinnäytetyöprosessi oli suhteellisen nopea ja sisälsi varsinkin aikataulullisia haasteita. Oman oppimisen kannalta huomasin, että projektin aikatauluttaminen on tärkeää. Tieteellinen kirjoittaminen lähdeviitteineen on työlästä ja paljon aikaa vievää. Opinnäytetyöhön kuuluvat välikokoukset pakottavat prosessia eteenpäin ja saavat työstämään tekstiä selkeät päämäärät tavoitteena.

Oman oppimisen kannalta olen oppinut paljon uutta responsiivisista sovelluskehityksistä ja niiden ominaisuuksista. Ennen opinnäytetyöprosessia, olin tutustunut lähinnä Bootstrapiin ja siihenkin hyvin pintapuolisesta. Samalla kun olen selvittänyt kyseisten kolmen sovelluskehityksen ominaisuuksia, olen oppinut myös paljon yleistä sovelluskehityksistä ja kuinka paljon julkaistaan jatkuvasti uusia. Nyt minulla on myös huomattavasti kattavampi kuva siitä mihin tarkoituksiin esimerkiksi Bootstrap soveltuu ja en välttämättä valitsisi sitä jokaiseen projektiin automaattisesti vaan punnitsisin ensin muitakin vaihtoehtoja.

Opinnäytetyön jatkotutkimuksen aiheena suosittelisin muistakin uusista ja merkittävistä tai suosiota saavuttaneista sovelluskehysistä kirjoittamista. Uikit ja Semantic UI ovat kaksi melko uutta ja suhteellisen laajaa sovelluskehystä, joista voisi olla hyvä saada lisää tietoa. Näistä kahdesta on saatavilla suhteellisen paljon tietoa ja kirjoitettuja artikkeleita web-pohjaisissa julkaisuissa.

Lähteet

Breheeny, R. Jung, E. Zürrer, M. 2012. Responsive design – harnessing the power of media queries. Luettavissa: <http://googlewebmastercentral.blogspot.jp/2012/04/responsive-design-harnessing-power-of.html>. Luettu 18.4.2015.

Bootstrap. 2015. CSS. Luettavissa: <http://getbootstrap.com/css/>. Luettu 30.4.2015.

Bright, P. 2014. HTML5 specification finalized, squabbling over specs continues. Luettavissa: <http://arstechnica.com/information-technology/2014/10/html5-specification-finalized-squabbling-over-who-writes-the-specs-continues/>. Luettu 18.4.2015.

Cashmore, P. 2012. Why 2013 Is the Year of Responsive Web Design. Luettavissa: <http://mashable.com/2012/12/11/responsive-web-design/>. Luettu 10.4.2015.

Chan, J. 2013. Responsive Web Design is Not the Future. Luettavissa: <http://sixrevisions.com/mobile/responsive-web-design-not-the-future/>. Luettu 18.4.2015.

De Graeve, K. 2011. Responsive Web Design. Luettavissa: <https://msdn.microsoft.com/en-us/magazine/hh653584.aspx>. Luettu 7.4.2015.

Danger Gardner, L. 2014. What We Mean When We Say “responsive”. Luettavissa: <http://alistapart.com/column/what-we-mean-when-we-say-responsive>. Luettu 17.4.2015.

Design Modo. 2014. Bootstrap vs. Foundation: Which Framework Should You Choose? [Infographic] Luettavissa: <http://designmodo.com/bootstrap-vs-foundation/>. Luettu 22.4.2015.

Design Your Way. 2015. Luettavissa: <http://www.designyourway.net/blog/inspiration/the-case-against-using-bootstrap-to-design-websites/>. Luettu 22.4.2015.

Firtman, M. 2014. You May Be Losing Users If Responsive Web Design Is Your Only Mobile Strategy. Luettavissa: <http://www.smashingmagazine.com/2014/07/22/responsive-web-design-should-not-be-your-only-mobile-strategy/>. Luettu 17.4.2015.

Foundation. 2015a. Building Forms With HTML. Luettavissa: <http://foundation.zurb.com/docs/components/forms.html>. Luettu 5.5.2015.

Foundation. 2015b. Buttons. Luettavissa:

<http://foundation.zurb.com/docs/components/buttons.html>. Luettu 5.5.2015.

Foundation. 2015c. Media Queries. Luettavissa: <http://foundation.zurb.com/docs/media-queries.html>. Luettu 5.5.2015.

Foundation. 2015d. Typography. Luettavissa:

<http://foundation.zurb.com/docs/components/typography.html>. Luettu 6.5.2015.

Foundation. 2015e. The Grid. Luettavissa: <http://foundation.zurb.com/grid.html>. Luettu 7.5.2015.

Foundation. 2015f. Grid. Luettavissa:

<http://foundation.zurb.com/docs/components/grid.html>. Luettu 14.5.2015.

Foundation. 2015g. Tables. Luettavissa:

<http://foundation.zurb.com/docs/components/tables.html>. Luettu 14.5.2015.

Frost, B. 2012. Creating a Mobile-First Responsive Web Design. Luettavissa:

<http://www.html5rocks.com/en/mobile/responsivedesign/>. Luettu 22.4.2015.

Frost, B. 2013. Performance as design. Luettavissa:

<http://bradfrost.com/blog/post/performance-as-design/>. Luettu 18.4.2015.

Gartner. 2015. Gartner Says Smartphone Sales Surpassed One Billion Units in 2014. Luettavissa: <http://www.gartner.com/newsroom/id/2996817>. Luettu 20.4.2015.

Ghazarian, A. 2014. The Pros and Cons of Responsive Web Design vs. Mobile Website vs. Native App. Luettavissa: <http://designmodo.com/responsive-design-vs-mobile-website-vs-app/>. Luettu 18.4.2015.

Guertler, M. 2012. Responsive Design: the Key to Responsive Mobile BI Applications. Business Intelligence Journal

Hamilton, C. 2013. Which Responsive Framework is Right for You? Luettavissa:

<http://www.ektron.com/Blogs/Corey-Hamilton/Which-Responsive-Framework-is-Right-for-You-/>. Luettu 25.4.2015.

- Korf, M & Oksman, E. 2012. Native, HTML5, or Hybrid: Understanding Your Mobile Application Development Options. Luettavissa:
https://developer.salesforce.com/page/Native,_HTML5,_or_Hybrid:_Understanding_Your_Mobile_Application_Development_Options. Luettu 22.4.2015.
- Korpela, J & Lehdonvirta, P. 2013. HTML5 sovellusalustana. RPS-yhtiöt. Vaasa.
- Knight, K. 2011. Responsive Web Design: What It Is and How To Use It. Luettavissa:
<http://www.smashingmagazine.com/2011/01/12/guidelines-for-responsive-web-design/>.
Luettu 18.4.2015.
- Johnson, J. 2013. Mobile First Design: Why It's Great and Why It Sucks. Luettavissa:
<http://designshack.net/articles/css/mobilefirst/>. Luettu 22.4.2015.
- Marcotte, E. 2010. Responsive Web Design. Luettavissa:
<http://alistapart.com/article/responsive-web-design>. Luettu 10.4.2015.
- Martsoukos, G. 2015. Getting Started with Skeleton, the Simple CSS Boilerplate. Luettavissa: <http://www.sitepoint.com/getting-started-with-skeleton-simple-css-boilerplate/>. Luettu 14.5.2015.
- Mozilla Developer Network. 2015a. CSS3. Luettavissa: <https://developer.mozilla.org/en-US/docs/Web/CSS/CSS3>. Luettu 20.4.2015.
- Mozilla Developer Network. 2015b. Introduction to HTML5. Luettavissa:
https://developer.mozilla.org/en-US/docs/Web/Guide/HTML/HTML5/Introduction_to_HTML5. Luettu 18.4.2015.
- Mozilla Developer Network. 2015c. CSS media queries. Luettavissa:
https://developer.mozilla.org/en-US/docs/Web/Guide/CSS/Media_queries. Luettu 17.5.2015.
- O'Dell, J. 2013. Here's a chart to decide whether to build a native or web mobile app. Luettavissa: <http://venturebeat.com/2013/07/29/native-v-web-chart/>. Luettu 23.4.2015.
- Podjarny, G. 2014. Real World RWD Performance – Take 2. Luettavissa:
<http://www.guyppo.com/real-world-rwd-performance-take-2/>. Luettu 17.4.2015.

Pettit, N. 2014a. The 2014 Guide to Responsive Web Design. Luettavissa: <http://blog.teamtreehouse.com/modern-field-guide-responsive-web-design>. Luettu 20.4.2015.

Pettit, N. 2014b. Should You Use Bootstrap or Foundation? Luettavissa: <http://blog.teamtreehouse.com/use-bootstrap-or-foundation>. Luettu 20.4.2015.

Rudolph, P. 2014. Hybrid Mobile Apps: Providing A Native Experience With Web Technologies. Luettavissa: <http://www.smashingmagazine.com/2014/10/21/providing-a-native-experience-with-web-technologies/>. Luettu 22.4.2015.

Schenker, M. 2014. Bootstrap vs. Foundation: Which Framework is Better? Luettavissa: <https://bootstrapbay.com/blog/bootstrap-vs-foundation/>. Luettu 19.5.2014.

Skeleton. 2015. A dead simple, responsive boilerplate. Luettavissa: Luettavissa: <http://getskeleton.com/>. Luettu 4.5.2015.

Statista. 2015. Number of smartphones sold to end users worldwide from 2007 to 2014 (in million units). Luettavissa: <http://www.statista.com/statistics/263437/global-smartphone-sales-to-end-users-since-2007/>. Luettu 20.4.2015.

Vermilion. 2014. Responsive CSS Framework Comparison. Luettavissa: <http://responsive.vermilion.com/compare.php>. Luettu 20.4.2015.

Vinaora. 2014. Do Modern Responsive Websites benefits Google ranking? Luettavissa: <http://vinaora.com/2014/08/do-modern-responsive-websites-benefits-google-ranking/>. Luettu 21.4.2014.

W3C (World Wide Web Consortium). 2014. HTML5 A vocabulary and associated APIs for HTML and XHTML. Luettavissa: <http://www.w3.org/TR/2014/REC-html5-20141028/>. Luettu 18.4.2015.

W3Schools. 2015a. CSS3 @media Rule. Luettavissa: http://www.w3schools.com/cssref/css3_pr_mediaquery.asp. Luettu 18.4.2015.

W3Schools. 2015b. CSS3 Introduction. Luettavissa: http://www.w3schools.com/css/css3_intro.asp. Luettu 20.4.2015.

Wikipedia. 2015a. Bootstrap (front-end framework). Luettavissa:
http://en.wikipedia.org/wiki/Bootstrap_%28front-end_framework%29. Luettu 7.5.2015.

Wikipedia. 2015b. Foundation (framework). Luettavissa:
[http://en.wikipedia.org/wiki/Foundation_\(framework\)](http://en.wikipedia.org/wiki/Foundation_(framework)). Luettu 7.5.2015.

Wilczynski, S. 2014. Front-end frameworks. Introduction (part 1). Luettavissa:
<http://www.merixstudio.com/blog/front-end-frameworks-introduction-part-1/>. Luettu 20.4.2015.

Liitteet

Liite 1. Vertailu sovelluskehysten ominaisuuksista

UI & Widgets	Bootstrap 3	Foundation v5.5	Skeleton v2.0.1
Alerts	Y	Y	N
Badges	Y	Y	N
Breadcrumbs	Y	Y	N
Buttons	Y	Y	Y
Carousel	Y	Y	N
Collapse/Accordion	Y	Y	N
Dropdown	Y	Y	N
Forms	Y	Y	Y
Form Validation	N	Y	N
Grids	Y	Y	Y
Icons	Y	Y	N
Labels	Y	Y	N
Lists	Y	Y	Y
Lists (horizontal)	Y	Y	N
Media Object	Y	N	N
Modal window	Y	Y	N
Navigation	Y	Y	N
Navigation - Sticky	Y	Y	N
Navigation - Off Canvas	N	Y	N
Pagination	Y	Y	N
Panels	Y	Y	N
Popovers	Y	Y	N
Pricing tables	N	Y	N
Print styles	Y	Y	N
Progress bars	Y	Y	N
Responsive media	N	Y	N
Right-to-Left	N	Y	N
Scrollspy	Y	Y	N
Tables	Y	Y	Y
Tabs	Y	Y	N
Thumbnails	Y	Y	N
Tooltips	Y	Y	N
Tour	N	Y	N
Typeahead	N	N	N
Typography	Y	Y	Y
Video scaling	Y	Y	N