

Opinnäytetyö (AMK)

Tietojenkäsittely

Yrityksen tietojärjestelmät

2016

Maria Aaltonen

RESPONSIIVISEN MOBILE FIRST -SIVUSTOPOHJAN SUUNNITTELU JA TOTEUTUS



TURUN AMMATTIKORKEAKOULU
TURKU UNIVERSITY OF APPLIED SCIENCES

OPINNÄYTETYÖ (AMK) | TIIVISTELMÄ

TURUN AMMATTIKORKEAKOULU

Tietojenkäsittely | Yrityksen tietojärjestelmät

2016 | 41

Tuomo Helo

Maria Aaltonen

RESPONSIIVISEN MOBILE FIRST - SIVUSTOPOHJAN SUUNNITTELU JA TOTEUTUS

Tämän opinnäytetyön tavoitteena oli suunnitella ja toteuttaa ensimmäinen mobile first -tekniikalla tehty template eli sivustopohja R2-sisällönhallintajärjestelmään. Lähtökohtana oli tarve luoda mobiililähtöinen sivustopohja kasvavasta mobiilitietoliikenteestä johtuen. Opinnäytetyön toimeksiantajana oli Mobiilimarkkinointi Routa.

Opinnäytetyössä perehdyttiin responsiiviseen web designiin ja mobiililähtöiseen suunnitteluun. Kun sivusto suunnitellaan mobile first -periaatteella, mobiilikäyttäjille pystytään tarjoamaan parempi käyttäjäkokemus kuin jos sivusto suunniteltaisiin ensin tietokoneita varten.

Sivustopohjan toteutusvaiheessa otettiin huomioon toimeksiantajan toiveet sivustopohjan sisällöstä ja ulkoasusta. Lisäksi opinnäytetyössä käytiin läpi, millaisia ongelmia toteutuksessa tuli vastaan.

Lopputuloksena Mobiilimarkkinointi Routa sai otettua käyttöön ensimmäisen mobile first -tekniikalla toteutetun sivustopohjan R2-järjestelmässä. Projektin jatkokehitysideana on parantaa sivustopohjaa tuomalla siihen lisää käyttöliittymäelementtejä.

ASIASANAT:

Responsiivinen suunnittelu, web-suunnittelu, www-sivustot, mobiililaitteet

BACHELOR'S THESIS | ABSTRACT

TURKU UNIVERSITY OF APPLIED SCIENCES

Business Information Technology | Business Information Systems

2016 | 41

Tuomo Helo

Maria Aaltonen

THE DESIGN AND IMPLEMENTATION OF A RESPONSIVE MOBILE FIRST TEMPLATE

The objective of this thesis was to design and implement a template into the R2 content management system using the mobile first approach. The starting point was the need to create a mobile-oriented template due to the growing mobile data traffic. The thesis was commissioned by Mobiilimarkkinointi Routa.

The thesis focuses on responsive web design and mobile first design. A site that has been designed using the mobile first technique can offer a better mobile user experience.

The client's wishes about the content and design of the template were taken into account in the implementation. The thesis also presents what kind of problems were faced during the implementation phase.

As a final result, Mobiilimarkkinointi Routa released its first mobile first template in the R2 content management system. Further development work for this project is to improve the template by bringing new user interface elements to it.

KEYWORDS:

Responsive web design, web design, www-sites, mobile devices

SISÄLTÖ

SANASTO	6
1 JOHDANTO	7
2 RESPONSIIVINEN WEB DESIGN	8
2.1 Responsiivisen web designin tärkeys	9
2.2 Erillinen mobiilisivusto	10
2.3 Responsiivisuus käytännössä	11
3 MOBIILILÄHTÖINEN SUUNNITTELU	15
3.1 Graceful degradation ja progressive enhancement	15
3.2 Mobile first	17
4 SIVUSTOPOHJAN TOTEUTUS	21
4.1 R2-sisällönhallintajärjestelmä	21
4.2 Suunnittelu	23
4.3 Sisällön toteutus R2-järjestelmään	26
4.4 Sivustopohjan ulkoasun toteutus	29
4.5 Uutiset-moduulin toteutus ja tyyllittely	33
4.6 Sivustopohjan testaus	37
YHTEENVETO	39
LÄHTEET	40

KUVAT

Kuva 1. Responsiivisen sivuston käyttäytyminen (Rocheleau 2015).	8
Kuva 2. Perinteinen ja responsiivinen sivusto (Makino & Phan 2015).	10
Kuva 3. Kolme eri tapaa määrittellä media query.	14
Kuva 4. Graceful degradation (Zurb 2015).	16
Kuva 5. Progressive enhancement (Zurb 2015).	16
Kuva 6. Kosketusnäytön käytön helppous (Aarabi 2013).	20
Kuva 7. Kuvakaruseelin asetukset.	22
Kuva 8. Hahmotelma templatien etusivusta älypuhelimelle suunniteltuna.	25
Kuva 9. Työpöytäversion ongelma keskityksen kanssa.	28
Kuva 10. Sivustopohjassa käytetty räätälöity tilauslomake.	29

Kuva 11. Mobiili- ja työpöytäversion navigaatio.	31
Kuva 12. Nosto Palvelut-sivulle testaustekstillä.	32
Kuva 13. Esimerkki tuotteet-kategoriaan kuuluvasta uutisesta.	34
Kuva 14. Tuotteet-listan kolme ensimmäistä tuotetta.	35
Kuva 15. Tarjoukset-kategorian uutiset ovat yhtä korkeita.	36

KUVIOT

Kuvio 1. Tietoliikenteen määrä vuosina 2011-2014 (StatCounter 2015).	17
Kuvio 2. Sivustopohjan sivukartta.	24

TAULUKOT

Taulukko 1. Esimerkki tyylimäärittelyissä käytettävistä etuliitteistä.	37
--	----

SANASTO

Breakpoint	Kohta, jossa media query erottaa sivuston rakenteen kahteen tai useampaan muunnelmaan (Peterson 2014, 116).
Desktop first	Sivuston suunnittelutapa, jossa työpöytäversio sivustosta suunnitellaan ensin (Code My Views 2016).
Hakukoneoptimointi	Verkkosivun optimointi käyttäjiä ja hakukoneita varten, joka tähtää verkkosivun parhaaseen mahdolliseen hakukonenäkyvyyteen (Google-optimointi.com 2016).
Hero-kuva	Sivostosuunnittelun trendi, jossa sijoitetaan iso kuva sivun alkuun (Cousins 2014).
Hover-tila	CSS:n hover-tila saadaan aktivoitua, kun kursori viedään jonkin elementin päälle klikkaamatta sitä (Peterson 2014, 236).
HTTP-pyyntö	Asiakkaan, esimerkiksi selaimen, lähettämä pyyntö palvelimelle, johon palvelin vastaa haluamallaan tavalla (Kuopion yliopisto 2016).
JQuery	JavaScript-kirjasto, joka tekee esimerkiksi HTML-dokumentin läpikäynnistä ja manipuloinnista helpompaa (jQuery.com 2016).
Media query	Kohdistaa tiettyjä CSS-tyylejä eri laitteiden ominaisuuksille (Fielding 2014, 59).
Mobile first	Sivuston suunnittelutapa, jossa mobiiliversio sivustosta suunnitellaan ensin (Code My Views 2016).
Viewport	Käyttäjälle näkyvä osa verkkosivusta. Viewport vaihtelee laitteen näytön ja ikkunan koon mukaan. (W3schools 2016a).

1 JOHDANTO

Tämän opinnäytetyön tavoitteena on suunnitella ja toteuttaa responsiivinen, mobile first -periaatteella toteutettu template eli sivustopohja toimeksiantajan käyttämään R2-sisällönhallintajärjestelmään. Toimeksiantaja on Mobiilimarkkinointi Routa Oy. Työn teoriaosuudessa perehdytään responsiiviseen web designiin, miksi se on tärkeää ja miten se toteutetaan teoreettisella tasolla sekä miten mobiililähtöinen suunnittelu eroaa perinteisestä desktop first -ajattelutavasta ja miksi se on parempi tapa. Soveltavassa osuudessa käydään läpi, millaisia haasteita sivustopohjan suunnittelu ja toteutus mobiililähtöisesti ja R2-järjestelmään asettavat.

Mobiilimarkkinointi Routa Oy on digitaalinen markkinointitoimisto ja osa Sanoma-konsernia. R2-järjestelmässä on noin neljäkymmentä sivustopohjaa ja Roudan kotisivusuunnittelijat valitsevat luodessaan uutta sivustoa aina jonkin näistä templateista pohjaksi. Vaikka sivustopohjia on suhteellisen paljon, aktiivisessa käytössä ei ole kuin muutama. Uuden sivustopohjan luominen lisää variaatiota. Lisäksi poiketen muista R2-järjestelmän templateista, uusi template toteutetaan mobile first -periaatteen mukaan eli mobiililähtöisesti.

Opinnäytetyön lähestymistapana on toimintatutkimus eli tutkimusongelmaa tutkitaan käytännössä tarkoituksena kehittää tai muuttaa sen hetkistä tilannetta. Tavoitteena ei siis ole vain tutkia, miten asiat ovat vaan myös miten niiden tulisi olla. (Kajaanin ammattikorkeakoulu 2015.)

2 RESPONSIIVINEN WEB DESIGN

Responsiivinen web design on Ethan Marcotten keksimä termi, jolla tarkoitetaan www-sivuston rakennetta, joka mukautuu näytön resoluution mukaan (Frost 2011). Responsiivisen web designin ideana on että sivusto mukautuu käyttäjän toimien, laitteen, näytön koon ja laitteen orientaation mukaan (Knight 2011). Tämän periaatteen mukaan sivuston rakenne ei mene rikki vaan sopeutuu näytön kokoon esimerkiksi tietokoneelta tablettiin vaihdettaessa.



Kuva 1. Responsiivisen sivuston käyttäytyminen (Rocheleau 2015).

Responsiivisen web designin ansiosta ei siis ole tarvetta tehdä uutta muotoilua jokaiselle eri näytön koolle. Sen sijaan muotoiluja voidaan pääpiirteissään suunnata eri laiteryhmillä. Tyypillisesti on neljä yleistä näytön kokoa, joihin responsiivinen web design on tähdännyt: iso tietokonenäyttö, pienempi tietokonenäyttö tai kannettavan tietokoneen näyttö, tabletti ja älypuhelin (Byers 2015a). Kuvassa 1 on esitelty miten responsiivinen sivusto käyttäytyy näillä neljällä näyttökoolla.

2.1 Responsiivisen web designin tärkeys

Responsiivisen sivuston luomisesta on tullut erittäin tärkeää, sillä yhä useampi käyttää internetiä mobiililaitteella. Sivuston responsiivisuus on myös 21. huhtikuuta 2015 lähtien vaikuttanut hakukoneoptimointiin, kun Google lisäsi algoritmiinsa mobiiliystävällisyyden. Googlen kehoitteena on tarjota sama HTML kaikille laitteille ja käyttää media queryjä ulkoasun kohdistamiseen eri laitteille. Googlen käyttäjät pystyvät täten löytämään korkealaatuisia hakutuloksia, joissa teksti on luettavaa ilman zoomaamista, linkkien välissä on tarpeeksi tilaa ja sivulla vältetään horisontaalista vierittämistä ja toimimatonta sisältöä, kuten Adobe Flashia. Vaikka tämä Googlen päivitys koskee kansainvälisesti kaikkia kieliä, se vaikuttaa vain yksittäisten sivujen eikä koko sivustojen sijoitukseen mobiililaitteilla tehtävillä hauilla. (May 2015; Makino & Phan 2015; Barsby 2015.)

Perinteisissä tietokoneissa responsiivisuus ei ole kaikkein tärkein asia websivustoa käytettäessä eikä välttämättä vielä tableteissakaan. Älypuhelimissa verkkosivun responsiivisuus on kuitenkin erittäin tärkeää käyttökokemuksen kannalta. Perinteisen mallin mukaan suunniteltu verkkosivu ei skaalaudu pieneen näyttöön, vaan sivusto pyrkii automaattisesti näyttämään kaiken sisällön näytöllä. Tämän vuoksi mobiililaitteella esimerkiksi teksti on liian pientä luettavaksi ja kuvat liian isoja. Kuvassa 2 on esitelty vasemmalla perinteisen mallin mukaan suunniteltu sivusto ja oikealla responsiivisesti suunniteltu sivusto.



Kuva 2. Perinteinen ja responsiivinen sivusto (Makino & Phan 2015).

Tärkein syy käyttää responsiivista web designia on se, että responsiivinen sivusto toimii oikein kaikilla tällä hetkellä markkinoilla olevilla laitteilla, mutta todennäköisesti myös tulevaisuudessa markkinoille tulevilla laitteilla. Responsiivisessa web designissa ei myöskään ole riskiä, että työpöytäversion käyttäjät näkisivät mobiiliversion sivustosta tai päinvastoin. Sivustot, joilla on erillinen mobiiliversio, käyttävät laitteen tunnistusta määrittelläkseen, kumpi sivustoista käyttäjälle tulisi näyttää: mobiili- vai työpöytäversio. Tämä prosessi ei kuitenkaan ole täysin virheetön, vaan välillä käyttäjä päätyykin väärään versioon. Laitteen tunnistus voi myös pidentää sivuston latausaikaa. Responsiivisessa web designissa sivustosta ei kuitenkaan ole kuin yksi versio, joten sivusto näytetään aina oikein. (Peterson 2014, 14.)

2.2 Erillinen mobiilisivusto

Vaikka responsiivinen web design on suositeltu ja yleisesti käytetty tapa suunnitella sivustoja, se ei kuitenkaan ole ainoa eikä ilman vikoja. Erillinen mobiilisivusto on suunniteltu erityisesti mobiililaitteita varten ottaen huomioon niiden asettamat

rajoitukset ja mahdollisuudet. Mobiilisivustolle pystytään siis paremmin suunnittelemaan erityisesti mobiilikäyttäjiä varten suunnattu käyttöliittymä ja käyttäjäkokemus. Mobiilisivuston täytyy kuitenkin olla hieman responsiivinen, jotta se mukautuu erikokoisiin mobiililaitteisiin. Mobiilisivusto voidaan luoda päädomainin alidomainiin, alikansioon tai kokonaan erilliselle domainille. (Ghazarian 2014; Robinson 2015.)

Käytettäessä erillistä mobiilisivustoa sivustoja on siis kaksi – pääsivusto ja mobiilisivusto. Kahden sivuston ylläpitäminen vie enemmän aikaa ja on täten kalliimpaa. Sen sijaan responsiivisesti suunnitellussa sivustossa on vain päädomain. Sivustoa on helpompi päivittää, eikä ole tarvetta monimutkaiselle uudelleenohjaukselle mobiilisivustolle. Responsiiviselle sivustolle ei myöskään tarvitse tehdä kuin yksi hakukoneoptimointi ja yksi sivusto on yksinkertaisesti halvempi toteuttaa ja ylläpitää kuin kaksi. (Ghazarian 2014; Robinson 2015.)

Responsiivinen web design ei kuitenkaan tarjoa aivan täydellistä mobiilikeskeistä käyttäjäkokemusta ja suuret sivustot, jotka toimivat hyvin työpöytäversiossa voivat olla hitaita mobiilissa. Responsiivinen web design on myös suhteellisen uusi käsite, eivätkä kaikki vanhat selaimet tue sitä täydellisesti. (Ghazarian 2014; Robinson 2015.)

2.3 Responsiivisuus käytännössä

Responsiivisuus saavutetaan käyttämällä joustavaa rakennetta, joustavia kuvia ja muuta mediaa sekä CSS:n media queryjä (Frost 2011). Joustavan rakenteen leveys vaihtelee riippuen laitteen viewportista eli ikkunan koosta. Toisin kuin kiinteä rakenne, jossa leveydet on määritelty pikseleissä, joustavassa rakenteessa leveydet määritellään prosenteissa. (Fielding 2014, 82.) Esimerkiksi jos kolumnin leveys on 60 prosenttia ja viewportin leveys on 1600 pikseliä, kolumnin leveys on 960 pikseliä. Jos taas viewportin leveys on 480 pikseliä, kolumnin leveys on 288 pikseliä.

Kuvat ovat tärkeä osa jokaista nettisivua. Ne ovat myös suurimpia syyllisiä rakenteen rikkomiseen. Perussääntö on asettaa kuvien maksimileveydeksi 100 prosenttia (Rocheleau 2015). Tämän ansiosta kuvat latautuvat alkuperäiskoossaan, jos kuvan sisältävässä elementissä on tarpeeksi tilaa. Jos kuvan sisältävä elementin, selaimen tai viewportin leveys on vähemmän kuin kuvan leveys, kuva mukautuu eli pienenee eikä riko sivuston rakennetta.

Media queryt ovat responsiivisen web designin ydin, koska niiden avulla voidaan kohdistaa tiettyjä CSS-tyylejä eri laitteiden ominaisuuksille. Yleisin tapa on kohdistaa eri tyylejä laitteiden viewportin leveyden mukaan, mutta media queryjä on monia muitakin. (Fielding 2014, 59.) Media queryt mahdollistavat sivuston erilaisen rakenteen ja ulkonäön riippuen siitä käytetäänkö sivua mobiilissa, tabletissa vai tietokoneella.

Media queryjen avulla voidaan esimerkiksi pienellä näytöllä näyttää kaikki sisältö vain yhdessä kolumnissa, mutta isommalla näytöllä kahdessa tai kolmessa. Navigaatio voidaan näyttää työpöytäversiossa kokonaisena, mutta mobiilissa piilotettuna pudotusvalikkona. Kuvat voidaan näyttää erikokoisina tai rajattuina ja fonttien koot voidaan pienentää paremmin pienelle näytölle sopiviksi. (Peterson 2014, 104–105.)

Media queryt voidaan jakaa kahteen eri osaan: media tyypeihin ja media expressioneihin. Media tyypit esiteltiin ensimmäistä kertaa CSS2.1-versiossa. Media tyyppien avulla voidaan määritellä eri tyyllitelyjä eri laitteille. Media tyypejä on yhteensä kymmenen, joista kolme on yleisimmin käytettyjä: all, print ja screen. All kohdistuu nimensä mukaisesti kaikille laitetyppeille, mutta print kohdistuu vain tulostimille ja screen näytöille. Muita media tyypejä ovat mm. television kaltaiset laitteet, projektorin kaltaiset laitteet ja pistekirjoituslaitteet. (Fielding 2014, 59–60.)

Varsinaiset media queryt eli expressionit lisättiin CSS3:en ja niiden tavoitteena oli antaa websuunnittelijoille enemmän vaikutusvaltaa miten sivustoja näytetään eri selaimilla ja laitteilla. Media queryt mahdollistavat siis tyylien kohdistamisen laitetyppeihin, mutta myös laitteiden ominaisuuksiin. Niillä voidaan hakea monia eri laitteen ominaisuuksia muun muassa viewportin korkeus ja leveys, laitteen

näytön korkeus ja leveys, laitteen orientaatio, kuvasuhde tai resoluutio. Näistä moniin voidaan vielä liittää max- tai min-etuliite. (Fielding 2014, 61; Peterson 2014, 110.)

Media queryt koostuvat ainakin yhdestä media tyypistä ja voivat lisäksi sisältää yhden tai useamman media expressionin (Fielding 2014, 63). Media querylle (expressionille) on lisäksi annettava jokin arvo eli breakpoint. Breakpoint on kohta, jossa media query erottaa sivuston rakenteen kahteen tai useampaan muunnelluun (Peterson 2014, 116). Yleisesti käytettyjä breakpointeja ovat 480 pikseliä älypuhelimille, 768 pikseliä tabletille pystysuunnassa, 1024 pikseliä tabletille vaakasuunnassa ja kaikki tätä leveämmät viewportit (Sharkie & Fisher 2013, 96). Breakpointien määrittelemisen laitteiden leveyden mukaan ei kuitenkaan ole paras käytäntö. Breakpoint tulisi määrittää vain silloin kun se on tarpeellinen eli silloin kun halutaan estää sivuston rakenteen rikkoutuminen.

Kolme eri tapaa käyttää media queryjä ovat

- kokonaan erilliset tyylitiedostot media queryille
- @importin käyttö päätyylitiedostossa
- media queryjen käyttö tyylitiedoston sisällä.

Erillisissä tyylitiedostoissa HTML-tiedoston <head>-tagin sisälle lisätään <link>-tagi, joka linkittää tyylitiedoston HTML-tiedostoon. Kaikki <link>-tagilla toteutetut tyylitiedostot ladataan käyttäjän laitteeseen, mutta ne eivät ole aktiivisia jos niitä ei tarvita. Erilliset tyylitiedostot lisäävät HTTP-pyyntöjen määrää ja voivat täten lisätä sivuston latautumisaikaa. @import käyttää myös erillisiä tyylitiedostoja, mutta @import määrittää päätyylitiedostossa HTML-tiedoston sijaan. Samaan tapaan kuin <link>-tagi @import lisää myös HTTP-pyyntöjen määrää. @import voi lisäksi estää tyylitiedostojen lataamisen samanaikaisesti. Media queryt voidaan myös kirjoittaa päätyylitiedoston sisään, jolloin ei ole tarvetta erillisille tyylitiedostoille. CSS-tiedoston sisällä olevat media queryt kumoavat olemassa olevat tyylimäärittelyt, kun media queryn asettamat ehdot toteutuvat. (Fielding 2014, 60;65–66.) Kuvassa 3 on esitetty nämä kolme eri tapaa määrittellä sama media

query: tyylit otetaan käyttöön kun laitteen tyyppi on näyttö, sen viewport on maksimissaan 500 pikseliä ja orientaatio on pystysuunta.

```
1 <link rel="stylesheet" media="screen and (max-width:500px) and (orientation:portrait)" href="esimerkki.css">
2
3
4 @import url("esimerkki.css") screen and (max-width: 500px) and (orientation: portrait);
5
6
7 @media screen and (max-width:500px) and (orientation:portrait){
8
9 }
10
```

Kuva 3. Kolme eri tapaa määritellä media query.

Responsiivisessa web designissa työpöytäversion muuttaminen pienille näytöille sopivaksi ei kuitenkaan välttämättä riitä, sillä mobiilisivujen käyttötarkoitus voi olla eri kuin työpöytäversion. Ihmiset selailevat internetiä mobiililaitteella silloin, kun heillä on tylsää, kiire tai he ovat eksyksissä (Creative Bloq 2012). Mobiililaitteen käyttäjä ei ole kiinnostunut siitä, kuinka kaunis sivusto on, vaan esimerkiksi yrityksen aukioloajoista ja sijainnista.

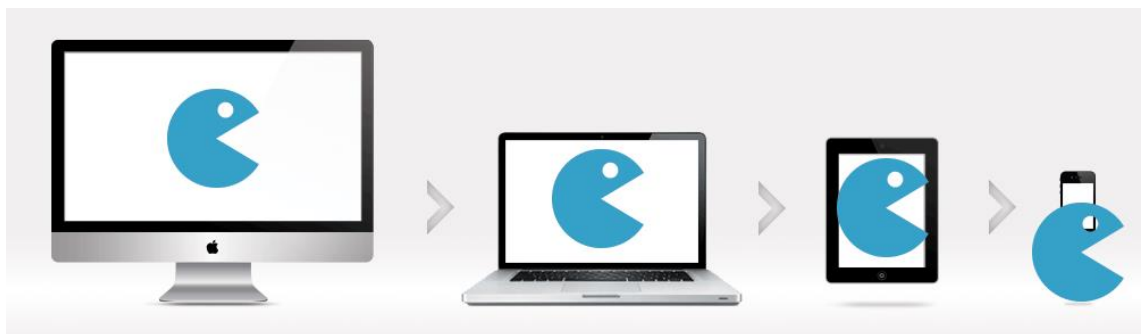
Responsiivisen web designin ansiosta sivuston elementtejä on mahdollista järjestää uudelleen ja pienentää suhteellisesti, jotta ne mahtuvat pienemmillekin näytöille. Kaikkien sisältöalueiden sovittaminen pienelle näytölle ei välttämättä ole paras ratkaisu. (Knight, 2011.) Sivustoa suunniteltaessa tulee ottaa huomioon, että kaikkea sisältöä ei tarvitse tai edes kannata näyttää mobiilissa. Piilotettavan sisältöalueen media queryyn voidaan määritellä CSS-tyyli *"display:none;"*. Esimerkiksi kuvakarusellit voivat antaa työpöytäversiolle hyvän visuaalisen ilmeen, mutta mobiilikäyttäjä ei välttämättä ole kiinnostunut miten hienoja kuvia sivustolla on. Tämän lisäksi mobiililaitteiden internetyhteydet eivät yleensä ole yhtä nopeita kuin tietokoneiden, joten kuvakarusellien lataaminen ja pyörittäminen on raskasta ja vie aikaa.

3 MOBIILILÄHTÖINEN SUUNNITTELU

Mobiililaitteet koetaan yleisesti alitehoisiksi versioiksi tietokoneista. Näin ei kuitenkaan ole, vaan ne itse asiassa ovat monin tavoin tehokkaampia kuin tietokoneet. Ihmisillä on älypuhelin aina mukana, aina päällä, ne ovat yhteydessä verkkoon ja ne voivat muun muassa tunnistaa sijainnin, liikkeen, orientaation ja ympäristön olosuhteet. Ei siis ole mikään ihme, että mobiililaitteille suunnittelemien on erilaista kuin työpöydälle suunnitteleminen. (Creative Bloq 2012.)

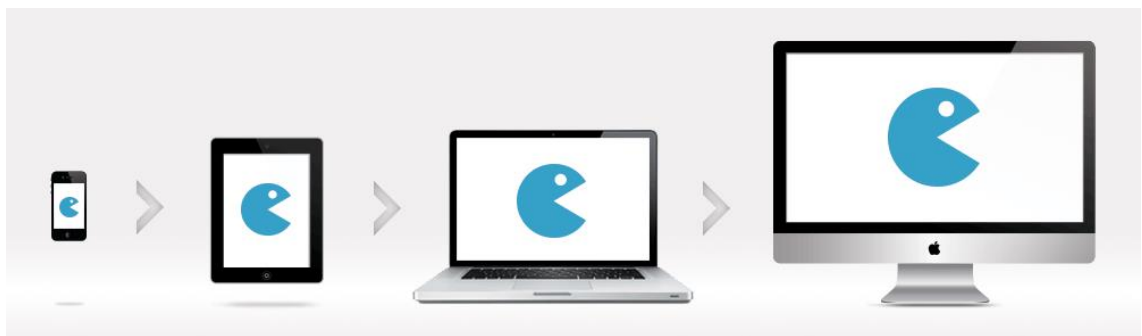
3.1 Graceful degradation ja progressive enhancement

Tyypillisesti sivuston suunnittelu on aloitettu työpöytäversiosta ja mobiilipuoli on jäänyt toissijaiseksi tavoitteeksi, joka on toteutettu myöhemmin. Responsiivisen web designin yleistyessä suunnittelijat ovat aloittaneet ”täysversiosta” ja lähteneet sen jälkeen työstämään pienemmille laitteille sopivaa sivustoa käyttäen apunaan media queryjä. (Code My Views 2015.) Tätä kutsutaan graceful degradation -periaatteeksi. Graceful degradation syntyi tarpeesta saada sivusto toimimaan mahdollisimman monella selaimella ja laitteella. Websuunnittelijat halusivat hyödyntää uutta teknologiaa jättämättä ulkopuolelle käyttäjiä, joiden laitteet eivät tukenneet kyseisiä toimintoja. Ratkaisuna oli siis luoda mahdollisimman hyvä käyttökokemus ensin. Kun selaimen valmiudet eivät enää riittäneet, toimintoja alettiin poistamaan siten että puutteista huolimatta sivusto olisi toimintakelpoinen. Responsiiviselle suunnittelulle tämä tarkoitti sitä että sisältöä ja toiminnallisuutta poistettaisiin. Täten mitä pienempi laitteen viewport, sitä yksinkertaisempi sivusto. (Fielding 2014, 67; Code My Views 2015.)



Kuva 4. Graceful degradation (Zurb 2015).

Graceful degradation haittaa myös sivuston suorituskykyä. Kuvassa 4 on esimerkkinä havainnollistettu saman kuvan koko verrattuna itse laitteen kokoon. Vaikka kuvan fyysinen koko on mahdollista sovittaa näytön kokoon, se ei pienennä kuvan kokoa tavuissa. Graceful degradation ylikuormittaa mobiililaitteet liialla informaatiolla, jolloin sivustojen latautumisaika on turhaututtavan pitkä (Zurb 2015).



Kuva 5. Progressive enhancement (Zurb 2015).

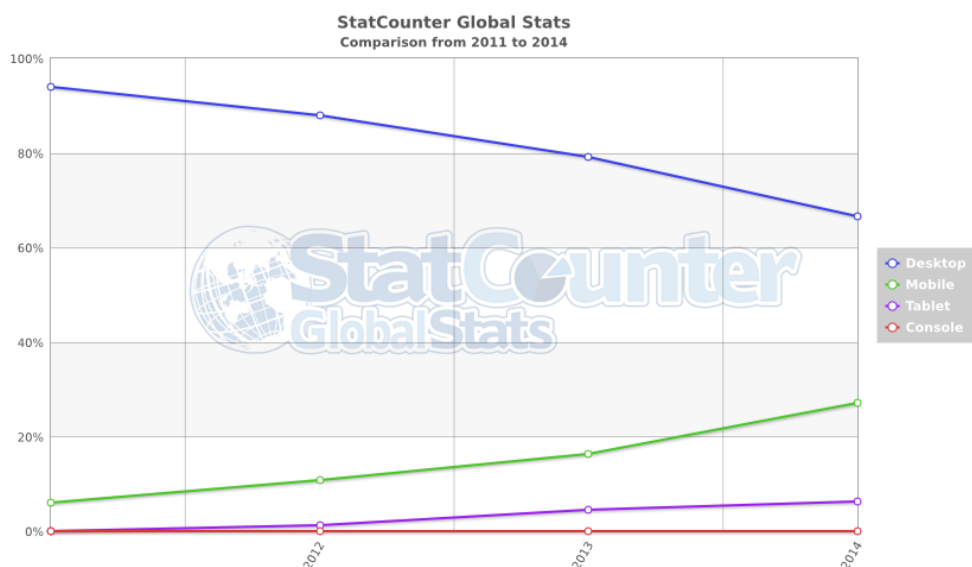
On olemassa kuitenkin toinen periaate, joka sopii paremmin responsiiviseen web designiin: progressive enhancement. Progressive enhancement -periaatteen mukaan sivusto suunnitellaan ensin selaimelle tai laitteelle, jolla on huonoimmat valmiudet. Tämän jälkeen sivustoa parannetaan progressiivisesti lisäämällä sisältöä ja toiminnallisuutta. (Fielding 2014, 67.) Kuvassa 5 on esitettyä progressive enhancement jossa kuvien koot on optimoitu jokaiselle laitteelle sopiviksi. Progressive enhancement -periaatteen avulla sivuston sisältö on suunniteltu toimivan mobiilissa paremmin kuin graceful degradation -periaatetta käytettäessä eli sivuston suorituskyky on parempi. Progressive enhancement -periaate yhdistettynä responsiiviseen web designiin luo responsiivisen mobile first web designin.

3.2 Mobile first

Mobile first on Luke Wroblewskin kehittämä tapa suunnitella sivustoja. Mobile first korostaa tarvetta aloittaa suunnittelu mobiililaitteille sopivaksi, jotta mobiilikäyttäjä saa parhaan mahdollisen käyttökokemuksen. Mobile first -periaatteen avulla

- sivustot tavoittavat enemmän ihmisiä
- suunnittelijoiden on pakko keskittyä tärkeimpään sisältöön ja toiminnallisuuteen
- suunnittelijat voivat hyödyntää mobiililaitteiden uusinta teknologiaa ja kehittää innovatiivisia ratkaisuja. (Frost 2011.)

Mobiilitietoliikenteen määrä on kasvanut räjähdysmäisesti viime vuosien aikana. Kuviossa 1 on esitetty tietoliikenteen määrää vuosien 2011 ja 2014 väliltä. Vuonna 2011 älypuhelimien osuus tietoliikenteestä vastasi vain noin kuutta prosenttia. Kolmessa vuodessa osuus kuitenkin kasvoi jo 27 %:iin ja tablettien osuus oli noin 6 %. Vastaavasti tietokoneiden osuus on laskenut jyrkästi 96 %:sta 67 %:iin. Toisin sanoen vuonna 2014 joka kolmas vierailu verkkosivulla suoritettiin mobiililaitteella. (StatCounter 2015.)



Kuvio 1. Tietoliikenteen määrä vuosina 2011-2014 (StatCounter 2015).

Tietokoneet ja tabletit ovat yleensä yhteydessä WiFi-verkkoon, mutta älypuhelin-ten yhteys voi vaihdella vahvasta WiFi-signaalista heikkoihin 3G- ja EGDE-verk-koihin. Vuonna 2009 Akamai Technologiesin tilaaman ja Forrester Consultingin tekemän tutkimuksen mukaan 47 % käyttäjistä kuitenkin odottaa sivuston latau-tuvan maksimissaan kahdessa sekunnissa. Tutkimuksen mukaan 40 % käyttä-jistä poistuu sivustolta, jos se ei lataudu kolmessa sekunnissa. (Akamai Techno-logies 2009.)

Samana vuonna Compuwaren teettämän tutkimuksen mukaan 58 % käyttäjistä toivoo sivuston latautuvan mobiilissa yhtä nopeasti tai jopa nopeammin kuin tie-tokoneella. Kaksi vuotta myöhemmin osuus oli noussut jo 71 %:iin, ja 74 % käyt-täjistä oli valmiita odottamaan maksimissaan viisi sekuntia ennen sivulta poistu-mista. Tutkimuksen mukaan myös huono kokemus mobiilisivuston käytössä vai-kuttaa siihen kuinka todennäköisesti käyttäjät palaavat sivustolle. (Compuware 2011.)

Yksi hyvä syy aloittaa suunnittelu mobiililähtöisesti on CSS-tiedosto itsessään. CSS-tiedostot luetaan ylhäältä alaspäin eli tyylimäärittelyt voidaan korvata, jos niille annetaan myöhemmin uudet arvot. Jos suunnittelu aloitetaan työpöytäver-siosta, kaikki työpöytäversioon tehtyt muutokset täytyy erikseen käydä korvaa-massa mobiiliversioon. Esimerkiksi linkit ovat oletusarvoisesti alleviivattuja. Jos alleviivaus halutaan poistaa työpöytäversiosta mutta säilyttää mobiilissa, CSS-tiedoston alussa täytyy ensin määritellä alleviivauksen poisto ja myöhemmin me-dia queryssä lisätä alleviivaus uudelleen. Tämä lisää koodin määrää ja taistelee DRY-periaatetta (don't repeat yourself, älä toista itseäsi) vastaan. Lisäksi desktop first -tyylillä toteutettujen sivujen latausaika mobiilissa on pitkä, sillä ensin lada-taan kaikki työpöytäversion tyylit ja vasta tämän jälkeen mobiilille tarkoitetut tyy-littelyt. (Baumann 2015.)

Mobile first pakottaa suunnittelijoita miettimään, mikä sisältö on liiketoiminnalle ja asiakkaille kaikkein tärkeintä. Kaikki hienot efektit kuten flash-animaatiot ja hover-tilalla toteutetut pudotusvalikot eivät toimi, joten suunnittelijat pystyvät keskitty-mään miten esittää sivuston sisältö kaikkein järkevimmällä tavalla. Älypuhelimien näyttö on noin 80 prosenttia pienempi kuin tietokoneen. Tilaa tulee siis käyttää

säästeliäämmiin kuin työpöytäversiota suunniteltaessa. Pieni näyttö myös rajoittaa kuinka paljon informaatiota voidaan esittää kerralla, ennen kuin näyttöä täytyy selata alaspäin tai siirtyä toiselle sivulle. Mobiilisivuston rakenne on yleensä lineaarinen. Sisällölle täytyy siis kehittää hierarkia eli tärkein sisältö tulisi esittää ylimpänä ja vähiten tärkein alimpana. Työpöytäversiossa kaksi sisältöaluetta voivat olla tasa-arvoisesti vierekkäin, mutta mobiilissa toinen on aina ennen toista. (Metamonks 2015; Zurb 2015.)

Pienikokoisen näytön ja vaihtelevien verkkoyhteyksien lisäksi mobiililaitteet asettavat muitakin haasteita. Kosketusnäyttö asettaa omat rajoitteensa sivustoa suunniteltaessa. Perinteisesti tietokonetta ohjataan hiirellä ja mobiililaitteita kosketuksella, vaikkakin tämä raja on hälvenemässä, kun markkinoille tulee yhä enemmän kosketusnäyttöisiä tietokoneita. Kosketusnäytöille suunnitteleminen on hankalampaa, koska niillä ei muun muassa ole kaikkia samoja valmiuksia kuin kursorilla ohjattavilla laitteilla. CSS:n hover-tila saadaan aktivoitua, kun kursori viedään jonkin elementin päälle klikkaamatta sitä. Kosketusnäytöt eivät tue hover-tilaa, joten sen käyttö ei ole kannustettavaa. Kosketusnäytön koskettaminen vastaa klikkaamista, joten hover-tila ei koskaan aktivoidu kosketusnäytöllä. Pelkästään ulkonäöllisten muutosten suorittaminen, kuten taustaväriin vaihto, hoverilla ei siis pystytä toteuttamaan. Monilla sivustoilla pudotusvalikot kuitenkin toteutetaan hoverilla. Valikko on normaalitilassa piilotettu, mutta kursorin vieminen valikkolinkin päälle tuo sen esiin. Useimmat kosketusnäytöt pystyvät mukautumaan muuttamalla valikon käyttäytymistä. Ensimmäinen kosketus toimii kuten hover ja tuo valikon esiin. Vasta toinen kosketus valitsee linkin. (Peterson 2014, 234-237.)

Sormi ei ole yhtä tarkka osoitin kuin kursori. Kursorin pystyy helposti kohdistamaan haluamaansa paikkaan, mutta sormi on isompi kuin kursori ja sormella on vaikeampaa osua pieniin kohteisiin. Yksittäiseen linkkiin on helpompaa osua ja vaikka sormi koskettaisikin useaa pikseliä näytöllä, älypuhelimet pystyvät yleensä päättämään mitä käyttäjä halusi koskettaa. Jos linkkejä on kuitenkin useampi lähellä, on todella helppoa vahingossa osua linkkiin, joka vie käyttäjän väärään paikkaan. Kosketuksen kohteen koolle (touch target size) onkin määritelty

useita eri ohjeita ja suositeltavaa on, että lähekkäisten kohteiden leveys olisi ainakin 44 pikseliä ja niille olisi varattu enemmän tyhjätillaa. (Peterson 2014, 238.)



Kuva 6. Kosketusnäytön käytön helppous (Aarabi 2013).

Suurin osa ihmisistä on oikeakätisiä ja käyttää oikeaa peukaloaan älypuhelimensa käyttöön. Kuvasta 6 ilmenee, että oikealla peukalolla helpointa on ylettyä näytön oikeaan alareunaan, kun taas vaikeinta on ylettyä vasempaan yläreunaan. Täten olisi järkevintä sijoittaa tärkeimmät painikkeet, kuten navigaatiopalkki, näytön alareunaan. Mobiilisovelluksissa tämä onkin yleisesti käytetty käytäntö, mutta sivustoissa on totuttu sijoittamaan navigaatio yläreunaan. (Peterson 2014, 241.)

Vaikka mobiililähtöinen suunnittelu asettaa omat haasteensa, tuo se myös mukanaan etuja. Ihmisillä on älypuhelin aina mukana ja mobile first -suunnittelu pystyy hyödyntämään mobiililaitteiden ominaisuuksia kuten GPS-navigointia. Mobiililaitte pystyy paikallistamaan käyttäjän sijainnin paremmin kuin tietokone ja esittämään olennaista tietoa ympäristöstä. Muita mobiililähtöisessä suunnittelussa hyödynnettyjä ominaisuuksia ovat muun muassa puhelintoiminnot ja ääni- ja videonytöt. (Byers 2015b; Zurb 2015.)

4 SIVUSTOPOHJAN TOTEUTUS

4.1 R2-sisällönhallintajärjestelmä

R2 on Mobiilimarkkinointi Roudalle räätälöity sisällönhallintajärjestelmä. R2-järjestelmällä luotavat sivut koostuvat erilaisista widgeteistä eli sisältöalueista. Tekstialueita varten tarkoitetut widgetit ovat eniten käytettyjä ja moniin eri käyttötarkoituksiin sopivia. Tekstialue-widgettejä ovat

- sisältöalue editorilla
- vakiosisältö editorilla
- koodiwidget.

Sisältöalue on tavallinen tekstieditori, johon voidaan esimerkiksi kirjoittaa otsikoita ja leipätekstiä, lisätä kuvia, linkkejä ja taulukoita sekä luoda lomakkeita. Tekstieditori muuttaa sisällön automaattisesti HTML-koodiksi, mutta antaa myös käyttäjän tehdä muutoksia koodiin. Vakiosisältö-widget on samanlainen tekstieditori kuin sisältöalue, mutta sen käyttötarkoitus on eri. Vakiosisältöä tulisi käyttää, kun vähintään kahdella sivulla halutaan näyttää täsmälleen sama sisältö. Esimerkiksi etusivulle ja yhteydenottosivulle voidaan haluta samanlainen tarjouspyyntö-lomake. Vakiosisällön avulla ei ole tarvetta luoda samaa lomaketta kahteen kertaan. Lisäksi vakiosisältö-widgettiin tehdyt muutokset päivittyvät molemmille sivuille. Koodiwidget ei ole yhtä yleisesti käytetty kuin sisältöalue- ja vakiosisältö-widgetit. Koodiwidget ei sisällä tekstieditoria, vaan on tarkoitettu HTML-koodin lisäämiseen.

Tekstialue-widgettien lisäksi R2-järjestelmässä on tiettyihin käyttötarkoituksiin erikoistettuja widgettejä. Näitä widgettejä ovat muun muassa kuvakaruselli, kuvagalleria, Google-kartta, navigaatio, kielivalinta, logo ja uutismoduuliin kuuluvat widgetit. Jokaisella erikoistetuilla widgeteillä on omat asetuksensa ja eri tavat lisätä sisältöä. Kuvassa 7 on esitelty kuvakarusellin asetuksia. Kuvakaruselliin voidaan lisätä kuvia tiedostonhallinnan kautta, kuville voidaan antaa otsikko ja kuvateksti ja kuvien järjestystä voidaan muuttaa. Karusellille voidaan muun muassa

antaa leveys ja korkeus, siirtymäefekti voidaan valita luettelosta ja kuvien vaihtovälille voidaan määritellä jokin aika.

Asetukset

Käynnistä banneri automaattisesti

Pause onMouseOverilla

Näytä sivutus

Näytä Seuraava/Edellinen painikkeet

Leveys (px)

Korkeus (px)

Easing

Siirtymäefekti

Siirtymänopeus (s)

Vaihtoväli (s)

Kuvan rajausta

Bannerikohteet

Kohde 1

Osoite

Pikkukuvan osoite

Pikkukuvan otsikko

Kohteen tiedot

Otsikko

Teksti

Linkin osoite

Linkin teksti

Kohde 2

Osoite

Pikkukuvan osoite

Kuva 7. Kuvakaruseelin asetukset.

R2-järjestelmällä luodut sivut käyttävät HTML5-kieltä ja niiden perusrakenne on aina sama. Widgetit sijoitetaan perusrakenteeseen valitsemalla niille sijaintialue ja järjestys kyseisen sijaintialueen sisällä. Sijaintialueet järjestyksessä ylhäältä alas ovat

- head
- header
- main
- column1
- column2
- footer
- scripts.

Head- ja scripts-alueet ovat erilaisia komentosarjoja ja seurantakoodeja varten. Header sijaitsee sivun yläosassa ja footer alaosassa. Sivun varsinainen sisältö sijoitetaan main-, column1- ja column2-alueisiin. Riippuen sivuston ulkoasusta main, column1 ja column2 voidaan sijoittaa vierekkäin tai allekkain.

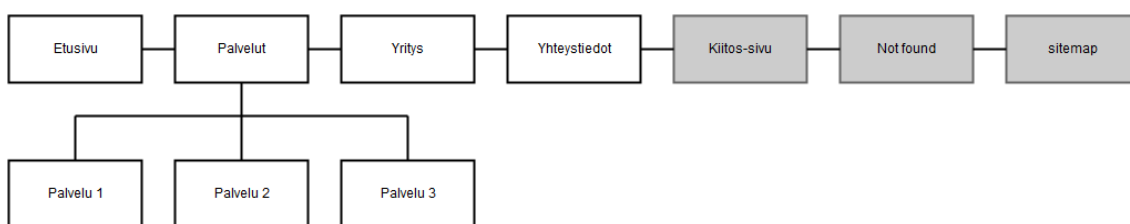
Header- ja footer-alueiden sisältö on yleensä muuttumaton kaikissa sivuston sivuissa. Esimerkiksi logo on yleensä sivuston jokaisella sivulla samassa kohtaa. Sivuston ylläpidon kannalta ei kuitenkaan ole käytännöllistä että jokaisella yksittäisellä sivulla on oma logo-widget. Jos logo myöhemmin haluttaisiin vaihtaa, se pitäisi vaihtaa jokaiselle sivulle erikseen. Tätä varten R2-järjestelmässä voidaan luoda yksi tai useampi sivupohja. Kaikki sivupohjassa olevat widgetit tulevat automaattisesti jokaiselle sivulle, jolloin ylläpito helpottuu.

4.2 Suunnittelu

Sivustopohjan suunnittelu aloitetaan sen tavoitteiden, sisällön ja perusrakenteen määrittelystä. Sivustopohjan tavoitteena on toimia alustana, jonka pohjalta on helppo luoda sivustoja eri yrityksille. Sen rakenne ja ulkonäkö tulee olla helposti muokattavissa, jottei kaikista sivustopohjalla tehdyistä sivustoista tule suurin piirtein samannäköisiä.

Sivustopohjan sisällön suunnittelu on haastava osio ja tulee miettiä huolella. Siinä käytettävä teksti tulee suurimmaksi osaksi koostumaan testaustekstistä sekä ohjeista sivustopohjan käyttöön. Ohjeiden tarkoitus on auttaa Roudan kotisivusuunnittelijoita käyttämään sivustopohjaa esittelemällä sen uusia ominaisuuksia ja eroja desktop first -periaatteella toteutettuihin sivustopohjiin. Sisältö ei tule kuitenkaan koostumaan pelkästä leipätekstistä, vaan haasteena on, että se muistuttaa mahdollisimman paljon valmiin sivuston sisältöä. Tekstin lisäksi sivustopohjaan tulee muita elementtejä ja widgettejä muun muassa lomakkeita, haitari (accordion), nostoja ja kuvagalleria. Mobiilimarkkinointi Roudan kotisivusuunnittelijoiden toiveena oli, että sivustopohjaan tulisi täysin räätälöity lomake. Lomakkeeseen kerättäisiin yleisesti käytettyjä syötteitä (input) ja niille määriteltäisiin toimivat tyylit.

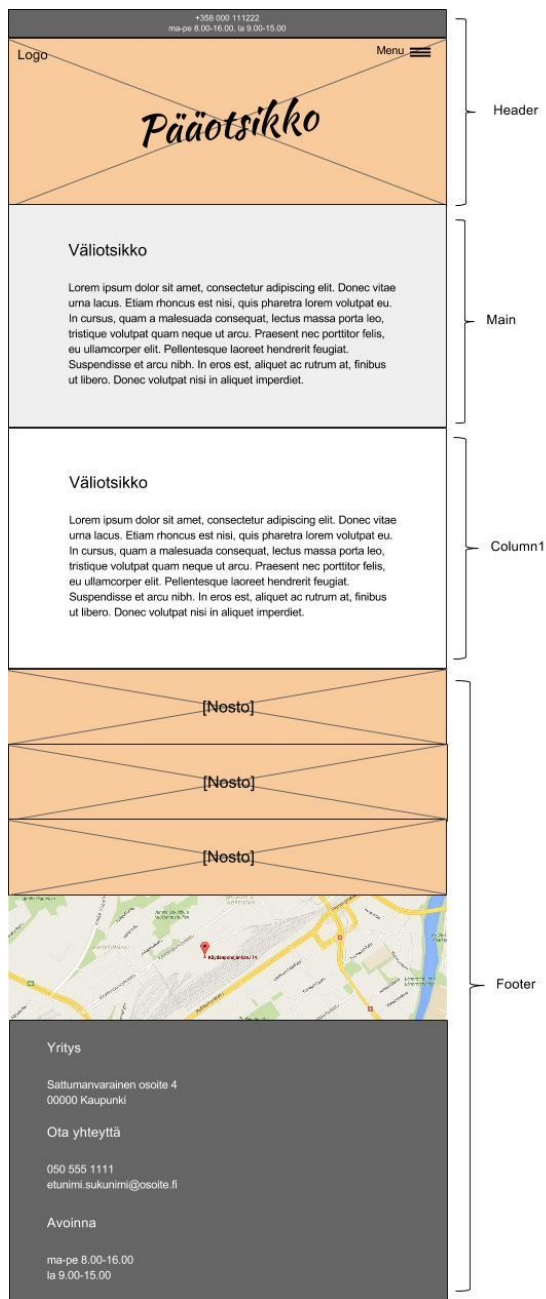
Sisällön suunnittelun jälkeen sivustopohjalle määritellään sivukartta ja sivujen rakenne. Kuviossa 2 on esitelty sivustopohjan sivukartta. Päätasoon sivuja ovat Etusivu, Palvelut, Yritys ja Yhteystiedot. Palvelut-sivulla on lisäksi kolme alisivua. Kiitos-sivu, Not found ja sitemap ovat myös päätasoon sivuja, mutta ne ovat piilotettuja. Piilotetut sivut eivät näy navigaatiopalkissa. Sivuston lomakkeet ohjautuvat Kiitos-sivulle, jossa käyttäjää kiitetään yhteydenotosta. Not found -sivu ilmoittaa käyttäjälle, että hänen hakemaansa sivua ei löydy. Sitemap on sivuston hierarkiapuun esittävä XML-dokumentti.



Kuvio 2. Sivustopohjan sivukartta.

Sivujen perusrakenteen suunnittelussa mietitään, mitkä sivujen elementit laitetaan mihinkin sijaintialueeseen. Kuvassa 8 on rautalankamalli (wireframe) sivustopohjan etusivusta älypuhelimelle suunniteltuna. Header-alueeseen on tarkoitus sijoittaa sivuston logo, navigaatio sekä hero-kuva ja pääotsikko. Koska sivuston mobiilikäyttäjä todennäköisesti etsii yrityksen perustietoja, header-alueen alkuun

on tarkoitus lisätä pieni alue, jossa on yrityksen puhelinnumero ja aukioloajat. Tätä aluetta ei kuitenkaan näytetä työpöytäversiossa, koska sen tärkeys ei ole yhtä suuri. Main-, column1- ja column2-alueisiin tulee sivun varsinainen sisältö esimerkiksi tekstit ja kuvat. Footer-alueeseen sijoitetaan nostot, kartta ja yhteystiedot.



Kuva 8. Hahmotelma templatien etusivusta älypuhelimelle suunniteltuna.

Header-alueen yhteystiedot, logo ja navigaatio sekä footer-alueen kartta ja yhteystiedot ovat jokaisella sivustopohjan sivulla sisällöltään samat. Siksi ne tullaan sijoittamaan sivupohjaan. Kaikki muu sisältö on jokaisella sivulla erilaista. Sijaintialueita ei myöskään käytetä jokaisella sivulla etusivun tapaan. Esimerkiksi hero-kuvaa ei tulla käyttämään kaikilla sivuilla ja nostot ovat vain etusivulla.

Kuten aikaisemmin todettiin, mobiililaitteiden näytön koko on paljon pienempi kuin tietokoneiden eli tilaa tulee käyttää säästeliäästi. Hero-kuva tulee tietokoneella olemaan koko viewportin levyinen ja korkuinen. Kuva toimii kuitenkin vain visuaalisena elementtinä, joten mobiilissa sen korkeus tulee olemaan paljon pienempi. Näin mobiilikäyttäjä pystyy helpommin keskittymään olennaiseen eli varsinaiseen tekstiin. Kaikki sisältö ei kuitenkaan ole tärkeää mobiilikäyttäjälle. Esimerkiksi etusivun column2-sijaintialueeseen tulee kuva. Kuva kuitenkin näkyy vain työpöytäversion käyttäjille, koska sen näyttäminen mobiilikäyttäjille ei ole välttämätöntä.

Koska mobiilikäyttäjille sivustopohjan rakenne on lineaarinen, sen sisällölle on kehitetty hierarkia. Kuvasta 8 voidaan havaita, että main-sijaintialue tulee ennen column1:tä. Jos column2:ssa olisi sisältöä, se tulisi column1:n ja footerin väliin. Tästä voisi päätellä, että tärkein sisältö tulisi aina sijoittaa main-alueeseen ja vähemmän tärkeä column1-alueeseen. Column2- ja footer-alueita käytettäisiin tarpeen mukaan. Tässä kohtaa tulee kuitenkin ottaa huomioon main-, column1- ja column2-alueiden keskinäinen sijoittelu työpöytäversiossa. Main-sijaintialue tulee aina ennen column1- ja column2-alueita, ja se on koko näytön levyinen. Työpöytäversiossa column1- ja column2-alueet ovat kuitenkin vierekkäin. Jos siis näistä kolmesta sijaintialueesta sisältöä halutaan lisätä vain kahteen, main-alue jätetään tyhjäksi.

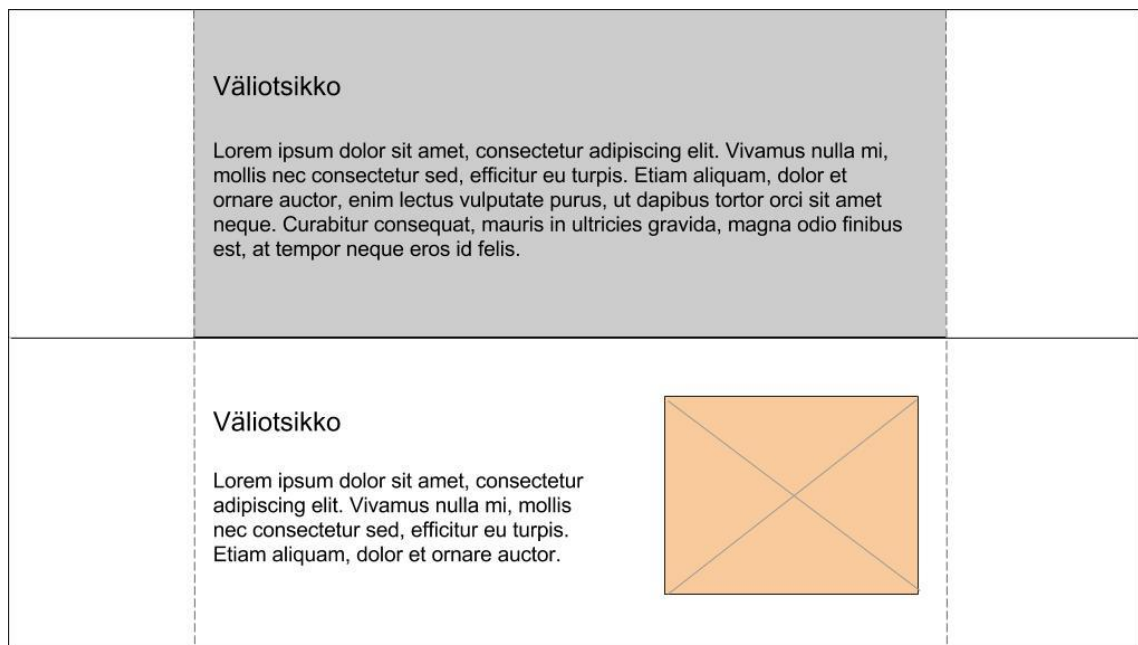
4.3 Sisällön toteutus R2-järjestelmään

Sivustopohjan toteutus aloitetaan luomalla uusi sivusto. Uutta sivustoa luotaessa sille annetaan nimi ja se määritellään sivustopohjaksi normaalin sivuston sijaan. Uudelle sivustopohjalle annetaan myös jokin vanhoista sivustopohjista pohjaksi.

Osa R2-järjestelmän sivustopohjista on nimetty aakkosjärjestyksessä. Päätin jatkaa tätä kaavaa, joten sivustopohjan nimeksi tuli L-Template.

Sivustopohjan luomisen jälkeen määritellään sen sivut. Etusivu ja piilotetut sivut luodaan automaattisesti, mutta muut sivut täytyy itse lisätä. Sivuille annetaan nimi, järjestysnumero, sijainti joko päätasolle tai jonkin toisen sivun alle, näkyvyys ja sivupohja. Näkyvyys voi olla normaali, sivukartassa, piilotettu tai kokonaan suljettu.

Ennen kuin ehdin luomaan kaikkia sivustopohjan sivuja, huomasin virheen suunnitelmassani. Virhe olisi tullut vastaan työpöytäversion ulkoasua tyyliteltäessä. Ideana oli, että työpöytäversiossa main-sijaintialue olisi ollut koko näytön levyinen. Column1- ja column2-alueet olisivat olleet vierekkäin. Näiden kolmen sijaintialueen sisältö keskitettäisiin näytön keskelle 1200 pikselin levyiselle alueelle. Kuvan 9 katkoviivat kuvastavat tätä keskitystä. Jotta tämä keskitys voidaan tehdä, main-, column1- ja column2-alueita ympäröivälle <div>:lle annetaan tyyli-määrittelyt *max-width:1200px;* ja *margin:0 auto;*. Tämä johtaa siihen että kuvan harmaa main-alue ei enää olekaan koko näytön levyinen. Main-alueen sisältö on kohdistettu oikein sivun keskelle, mutta harmaan alueen pitäisi jatkua koko sivun levyiseksi.



Kuva 9. Työpöytäversion ongelma keskityksen kanssa.

Ratkaisuna tähän ongelmaan on column1- ja column2-alueiden sijoittaminen allekkain, ei vierekkäin. Halusin kuitenkin säilyttää alkuperäisen suunnitelman rakenteen, joten siirsin sisältöjä sijaintialueesta toiseen. Etusivulla tämä tarkoittaa sitä, että kuva, joka ennen oli column2-alueessa, siirretään column1-alueeseen. Column1-alueen teksti ja kuva sijoitetaan vierekkäin käyttämällä joustavaa ruudukkoa. Lisäksi footer-alueen nostot siirretään column2-sijaintialueeseen.

Kun olin korjannut virheen suunnitelmassani, ryhdyin lisäämään sisältöä sivustopohjaan. Päätin aloittaa sivupohjan widgeettien määrittelyistä. Näin sivustopohjaan saadaan runko, josta on helppo lähteä rakentamaan yksilöllisiä sivuja. Kuten jo aikaisemmin todettua sivupohjaan sijoitetaan muun muassa logo, navigaatio, kartta ja yhteystiedot. Sivupohjassa on myös erilaisia seurantaskriptejä sekä sivustopohjassa käytetyt jQuery-koodit.

Kun sivupohja oli määritelty, siirryin lisäämään sisältöä sivuille. Päätin aloittaa etusivusta, koska olin tehnyt tarkan suunnitelman siitä. Ulkoasun muokkaaminen tulisi vasta myöhemmässä vaiheessa, joten tässä vaiheessa lisäsin sivulle vain tekstit ja kuvat. Muista sivustopohjan sivuista en ollut tehnyt rautalankamalleja, mutta olin etukäteen miettinyt mitä widgeettejä ja minkälaista sisältöä jokaiselle sivulle tulisi leipätekstin ja otsikoiden lisäksi. Esimerkiksi Yritys-sivulle tulisi haitari ja Palvelu 3 -aläsivulle kuvagalleria. Lisäksi Palvelut-sivulle ja sen aläsivuille tulisi toivottu räätälöity lomake. Lomakkeeseen määriteltäisiin syötteet tekstile, numeroille, puhelinnumerolle, sähköpostille, tekstikentälle, pudotusvalikolle sekä valintaruuduille ja -napeille. Syötteiden ulkoasu voitaisiin lisäksi määritellä niin että kaksi tai useampi syöte olisivat rinnakkain. Kuvassa 10 on esiteltynä lopullinen versio tästä lomakkeesta.

Tilauslomake

Lisävarusteet:

<input type="checkbox"/>	kpl näyttösuojakalvo 19,95 €	<input type="checkbox"/>	kpl suojakotelo 44 €
<input type="checkbox"/>	kpl näppäimistö 99 €	<input type="checkbox"/>	kpl USB-kaapeli 49 €

Valitse merkki:

Acer

Nimi	Puhelin	Sähköposti
<input type="text"/>	<input type="text"/>	<input type="text"/>

Viesti

Toimitus:

Kotiinkuljetus

Paketti

Nouto myymälästä

Haluan vahvistuksen:

Sähköpostitse Puhelimitse

Kuva 10. Sivustopohjassa käytetty räätälöity tilauslomake.

Koska kyseessä on sivustopohja eikä varsinainen sivusto, sisällön ei tarvitse muodostaa yhtenäistä kokonaisuutta. Tarkoituksena on pikemminkin testata melkein kaikkia R2-järjestelmän widgettejä sekä yleisesti käytettyjä HTML-elementtejä. Käytetyille widgeteille ja elementeille on täten helppo antaa tyylimäärittelyt, koska ne nähdään heti selaimessa. Widgettien ja elementtien tyylimäärittelyt mukautuvat tietysti sivustopohjan yleisulkoasua, mutta ne toimivat silti hyvänä pohjana aloittaa rakentamaan uutta sivustoa.

4.4 Sivustopohjan ulkoasun toteutus

R2-sisällönhallintajärjestelmässä tyylimäärittelyt perustuvat CSS3:een ja LESSiin. LESS on CSS:n esiprosessori, joka mahdollistaa sisäkkäiset luokat sekä muuttujien ja mixinien määrittelyn (Less.org 2016). Sisäkkäiset luokat lyhentävät koodin määrää, ja muuttujien ja mixinien ansiosta sivuston ylläpito helpottuu huomattavasti. Esimerkiksi jos tietyillä sivuston elementeillä on yhteinen värisävy, niille voidaan määrittellä muuttuja ”@nimi” ja antaa muuttujalle jokin arvo. Jos värisävy halutaan myöhemmässä vaiheessa vaihtaa, vain muuttujalle täytyy antaa

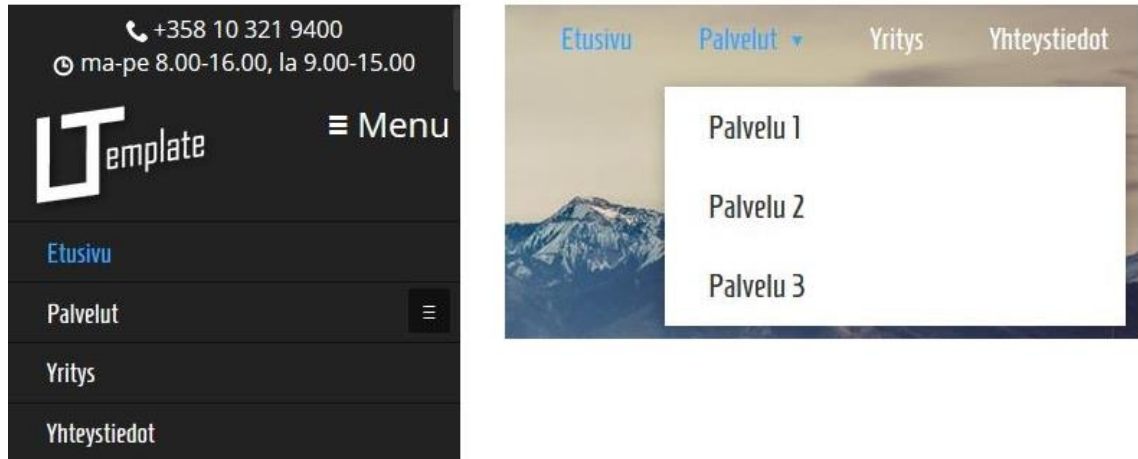
uusi arvo. CSS:ssä uusi värisävy pitäisi käydä vaihtamassa erikseen jokaisessa elementissä, jossa sitä on käytetty.

R2-järjestelmän muissa sivustopohjissa muuttujia on käytetty juurikin värien määrittämiseen. Myös L-Templatessa värit on määritelty muuttujissa, mutta niiden määrää on vähennetty huomattavasti. Muuttujia on vain kuusi noin kahdenkymmenen sijaan. Näin vältetään turhaa toistamista, sillä kahdella tai useammalla muuttujalla on voinut olla sama arvo. Kuudesta väristä kaksi ovat ns. räätälöityjä eli sivuston päävärit. Loput neljä ovat leipätekstin väri, otsikoiden oletusväri, valkoinen sekä tumma väri.

Sivustopohjan päätyylitiedostossa style.less:ssä on määritelty melkein kaikkien widgettien ja HTML-elementtien ulkonäkö. Koska L-Templaten pohjana on käytetty toista sivustopohjaa, tämä asettaa omat etunsa ja haasteensa. Etuna on se, että sivustopohjaan on valmiiksi määritelty perusrakenne esimerkiksi joustava ruudukko, fonttien koot ja tyhjätilan määrä elementtien ympärillä. Valmiiksi määriteltyjen mixinien avulla oli vaivatonta esimerkiksi vaihtaa L-Templateen sopivat fontit. Muuttujien avulla olisi ollut helppo vaihtaa sivustopohjan värimaailma. Päätin kuitenkin käyttää vain kuutta muuttujaa, joten jouduin määrittelemään uudet muuttujat ja käymään vaihtamassa uudet nimet vanhojen tilalle. Haasteena ovat tietenkin tyylimäärittelyt, jotka koskevat vain tätä sivustopohjaa. Suurin haaste on kuitenkin kääntää vanhan, desktop first -periaatteella toteutetun sivustopohjan tyylimäärittelyt mobiililähtöisiksi käyttäen mobile first -periaatetta.

Sivustopohjan tyylimäärittelyjen muuttaminen mobile first -periaatteen mukaisiksi luulisi olevan melko helppoa. Ei kuitenkaan riitä että perustyyli siirretään media queryyn ja media queryssä olleet määrittelyt perusmäärittelyiksi. Varsinkin navigaation muuttaminen mobiililähtöiseksi aiheutti paljon ongelmia. Mobiiliversiossa navigaatio meni aluksi osittain yläpalkin yhteystietojen ja logon päälle. Pääsivujen alisivut myös näkyivät, vaikka niiden pitäisi näkyä vasta kun ne erikseen klikataan auki. Työpöytäversiossa Palvelut-sivun alisivut olisivat pitäneet tulla esiin, kun hiiri viedään Palvelut-linkin päälle. Alisivut eivät kuitenkaan tulleet näkyviin.

Kuvasta 11 näkee miten navigaatioiden olisi pitänyt toimia. Ratkaisuna näihin ongelmiin oli käydä koodia kohta kohdalta läpi ja etsiä mitkä tyylit sopivat vain mobiiliversioon ja mitkä työpöytäversioon.



Kuva 11. Mobiili- ja työpöytäversion navigaatio.

L-Templatessa kaikkien media queryjen media expressionina on käytetty minimileveyttä. Toisin sanoen viewportin täytyy olla tietyn arvon levyinen, ennen kuin media queryn sisältämät tyylimäärittelyt tulevat voimaan. Nämä arvot eli breakpointit on valittu sen mukaan missä kohtaa alkuperäinen mobiiliversio sivustopohjasta ei enää näyttänyt hyvältä. Esimerkiksi joustavalla ruudukolla toteutetut elementit ovat mobiiliversiossa kaikki allekkain, mutta elementtejä aletaan sijoittaa vierekkäin viewportin leveydessä 570 pikseliä.



Kuva 12. Nosto Palvelut-sivulle testaustekstillä.

Hover-tilan puuttuminen kosketusnäytöllisistä mobiililaitteista oli toinen haaste tehdä sivustopohjasta mobiililähtöinen. Tämä ongelma tuli vastaan erityisesti etusivun nostojen kanssa. Nostojen ideana on toimia kuvallisina linkkeinä L-Temp-laten muille sivuille. Kuvan keskelle tulee sivun nimi. Työpöytäversiossa kun hiiren vie kuvan päälle, kuvan alareunaan ilmestyy tekstiä, sivun nimi siirtyy hiukan ylemmäs ja kuvan päälle tulee läpikuultava liukuvärjäys kuvan 12 mukaan. Mobiililaitteilla nämä efektit eivät tule toimimaan hover-tilan puuttumisen vuoksi. Nostot voitaisiin suunnitella niin, etteivät mobiilikäyttäjät koskaan tulisi näkemään näitä piilotettuja osia. En kuitenkaan halunnut, että mobiilikäyttäjät jäisivät mistään paitsi. Ratkaisuna oli siis suunnitella nostot niin että perustyleissä kaikki elementit ovat näkyvissä. Vasta media queryissä osa elementeistä tehtäisiin läpinäkyviksi ja hover-tilan avulla taas näkyviksi.

4.5 Uutiset-moduulin toteutus ja tyylittely

Räätälöity lomake oli yksi Roudan kotisivusuunnittelijoiden toiveista. Toinen oli Uutiset-moduulin widgettien valmiit tyylimääritykset. Uutiset-moduuli ei ole valmiina mukana R2-järjestelmällä tehdyllä sivustolla, mutta se voidaan siihen erikseen asentaa. Uutiset-moduulin alkuperäisenä käyttötarkoituksena on blogityylinen esitellä yrityksen ajankohtaisia uutisia. Moduulia voidaan kuitenkin käyttää muihinkin tarkoituksiin esimerkiksi tarjous- tai tuoteluettelonä. Päätin käyttää Uutiset-moduulia näihin kolmeen eri käyttötarkoitukseen.

Uutiset-moduulia varten loin uuden testisivuston L-Templaten pohjalta ja asensin moduulin sivustolle. Jos olisin asentanut moduulin suoraan L-Templateen, se olisi tullut mukaan kaikkiin sivustoihin, jotka käyttävät pohjanaan L-Templatea. Uutiset-moduulissa voidaan luoda uusia kategorioita ja uutisia sekä muokata ja poistaa vanhoja. Kategorioiden avulla uutisia voidaan jaotella, tässä tapauksessa ajankohtaisiin, tuotteisiin ja tarjouksiin. Kuvassa 13 on esitelty yksi tuotteet-kategorian uutisista. Uutiselle annetaan otsikko, osoite, kategoria, julkaisuaika, näkyvyys ja leipäteksti. Uutiselle on mahdollista myös antaa päättymisaika, lähde ja kuva.

Otsikko
iPad Air 32 GB Wi-Fi + Cellular

Osoite
ipad-air-32-gb-wi-fi-cellular

Liitä uutinen ryhmään
 Ajankohtaista Tarjoukset Tuotteet

Ajankohta
 Julkaisuaika: 30.11.2015 17:42
 Päättymisaika

Näkyvyys

Lähde/kirjoittaja

Kuva
 tuotekuvat/tabletti.jpg

Teksti

569,00 € (norm. 589,00 €)

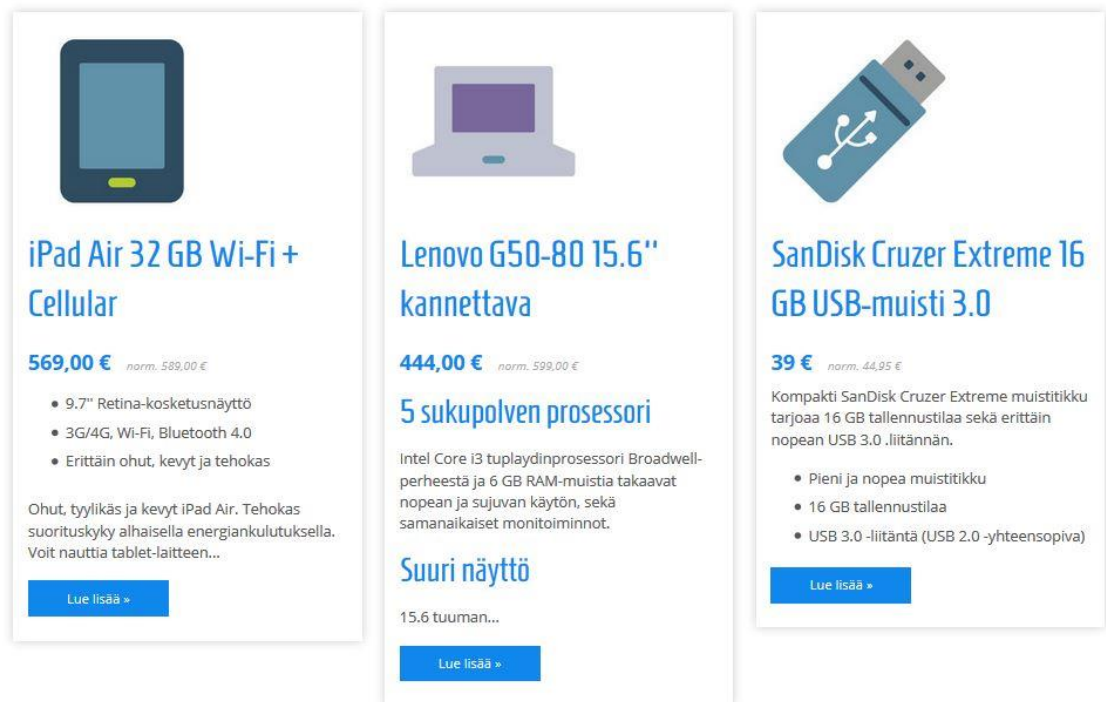
- 9,7" Retina-kosketusnäyttö
- 3G/4G, Wi-Fi, Bluetooth 4.0
- Erittäin ohut, kevyt ja tehokas

Ohut, tyylikäs ja kevyt iPad Air. Tehokas suorituskyky alhaisella energiankulutuksella. Voit nauttia tablet-laitteen monipuolisuudesta jopa 10 tuntia. Laitteessa on kestävä alumiininen runko ja kirkas 9,7" Retina-näyttö sormenjälkiä hylkivällä pinnoitteella.

Kuva 13. Esimerkki tuotteet-kategoriaan kuuluvasta uutisesta.

Sivujen päivityksen puolella Uutiset-moduuli koostuu neljästä eri widgetistä: Uutisnostosta, Uutislistauksesta, Uutisen näytöstä ja Uutiskategorioiden listauksesta. Uutislistaus listaa kaikki tietyn kategorian tai kategorioiden uutisten otsikot. Uutislistauksen asetuksista voidaan myös esimerkiksi valita näytetäänkö uutisesta lyhennelmä, kuva ja päivämäärä. Kun yhtä listauksen uutisista klikataan, Uutisen näyttö -widgetti näyttää kyseisen uutisen kokonaisuudessaan. Uutisnosto toimii samalla tavalla kuin Uutislistaus. Niiden erona on se, että Uutisnostossa voidaan valita kuinka monta uutista näytetään kaikkien sijaan. Uutiskategorioiden listaus on yksinkertainen linkkilista, johon voidaan valita halutut kategoriat.

Koska Uutiset-moduulilla on useita eri käyttötarkoituksia, myös niiden tyyllittelyt ovat erilaisia. Ajankohtaista-kategorian Uutislistaus ja Uutisnosto ovat hyvin yksinkertaisia. Yksittäiset uutiset ovat allekkain ja Uutislistauksessa uutiskuvat ovat uutisen oikealla puolella. Tuotteet-kategorian tyyli on verrattuna jo paljon monimutkaisempi. Työpöytäversiossa tuotteet on aseteltu vierekkäin siten että yhdellä rivillä on kolme tuotetta. Kuvassa 14 on kolme ensimmäistä tuotetta. Tarjous- ja normaalihintojen tyyllittelyyn apuna on käytetty jQueryä. Tekstieditorissa hinnat on määritelty muodossa *|tarjoushinta (normaalihinta)|*. JQueryn avulla ensin etsitään pystyviivat. Pystyviivojen välinen sisältö sijoitetaan ``-tagien sisään ja tagille annetaan luokka `price`. Tämän jälkeen etsitään sulkeet. Myös niiden välinen sisältö sijoitetaan ``-tagien sisään, ja tagille annetaan luokka `norm`. Hintojen tyyllittely on hoidettu JQueryn avulla, jotta sivuston loppukäyttäjän eli Mobiilimarkkinointi Roudan asiakkaan olisi mahdollisimman helppo käydä itse päivittämässä tuoteluettelo.



Kuva 14. Tuotteet-listan kolme ensimmäistä tuotetta.

Tarjoukset-kategorian tyylit muistuttavat pitkälti tuotteet-kategorian tyylejä. Tarjouksien tyylimäärittelyillä on kuitenkin kovemmat vaatimukset. Tarjoukset on aseteltu samalla tavalla vierekkäin kuin tuotteetkin. Kuvasta 14 havaitaan että tuotteiden korkeudet vaihtelevat. Kaikilla tarjouksilla tuli kuitenkin olla sama korkeus kuvan 15 mukaan. Verrattuna toisiinsa tarjousten kuvat ovat kaikki erikokoisia. Osa kuvista on vaakasuuntaisia, osa pystysuuntaisia. Kuvat tuli kuitenkin esittää siten, että ne ovat kaikki samankokoisia. Lisäksi Tutustu-painike piti sijoittaa tarjouksen oikeaan alanurkkaan.

Lappi, Suomi

129 € norm. 161 €

Elämyksiä - kokemuksia jotka puhuttelevat kaikkia aisteja ja tunteita jättäen pysyviä muistoja, ei tarvitse Lapissa etsimällä etsiä. Ainutlaatuisen sääolojen ja luonnon voima herkistää aistit, ja omat...

[Tutustu »](#)

Miami Beach, Yhdysvallat

2798 € norm. 2918 €

- Miami Beachin lämpö ja aurinko houkuttelevat viettämään rentoa lomaa
- Rannat, erityisesti pitkä South Beachin ranta vetää puoleensa auringonpalvoja
- Ostokset kannattaa tehdä Lincoln Roadilla
- Miami Beachin...

[Tutustu »](#)

Lontoo, Iso-Britannia

1040 € norm. 1218 €

Joku matkustaa Lontooseen nähdäkseen näyttelyn Tate Modernissa, toinen saadakseen maistaa tuttua olutta. Täältä löytyy jalkapalloa ja nostalgialta, tavarataloja ja historiaa.

Nähtävyydet

Historiallisia...

[Tutustu »](#)

Kuva 15. Tarjoukset-kategorian uutiset ovat yhtä korkeita.

Tarjouksien tekeminen yhtä korkeiksi hoitui helposti jQueryllä. Funktion avulla etsitään korkein tarjous ja sen arvo annetaan muille tarjouksille. Tutustu-painikkeen sijoittaminen oikeaan alanurkkaan onnistui osittain antamalla sille tyylimäärityt

- `position:absolute;`
- `bottom:10px;`
- `right:10px;`

Painike jäi kuitenkin hieman yllä olevan tekstin päälle. Ongelmana oli komentosarja, jolla määritetään tarjouksien korkeus. Koska painikkeen `position` oli `absolute`, sen korkeutta ei enää laskettu mukaan tarjouksen korkeuteen. Ratkaisuna tähän ongelmaan oli poistaa `position:absolute;` painikkeen tyylimäärityistä ja siirtää se skriptiin. Tällöin komentosarja pystyy laskemaan tarjouksen todellisen korkeuden ja vasta sen jälkeen sijoittamaan painikkeen alanurkkaan.

Kuvien rajaaminen samankokoisiksi onnistui CSS:n avulla. Kuvan sisältävälle `<div>`-tagille annetaan kiinteä korkeus ja sen ylivuoto piilotetaan. Itse kuvalle annetaan sadan prosentin leveys. Lopuksi kuva vielä keskitetään siten, että piilotettua ylivuotavaa osaa on yhtä paljon ylä- ja alaosassa.

4.6 Sivustopohjan testaus

L-Template on ensisijaisesti suunniteltu moderneille selaimille. Responsiivisuus jo itsessään vaatii nykyaikaisen selaimen, sillä vanhat selaimet eivät tue media queryjä (W3schools 2016b). Lisäksi L-Template käyttää muita tekniikoita, jotka toimivat kunnolla vain moderneilla selaimilla.

L-Templatessa on kuitenkin pyritty ottamaan myös vanhemmat selaimet huomioon mahdollisuuksien mukaan. Esimerkiksi etusivun koko viewportin kokoinen taustakuva ei toimi oikein kaikilla selaimilla. Taustakuvan sisältävälle <div>-tagille on määritelty korkeus 100vh eli koko viewportin korkeus. Vanhat selaimet kuten Internet Explorer 8 eivät kuitenkaan tue vh-mittayksikköä (W3schools 2016c). Tässä tapauksessa on käytetty varasuunnitelmaa eli <div>-tagille on lisäksi annettu minimikorkeus 600 pikseliä.

Kuten aiemmin mainittiin, L-Templanten tyylimäärittelyissä on käytetty CSS3:a (ks. luku 4.4). Vanhat selaimet eivät kuitenkaan tue kaikkia CSS3:n ominaisuuksia ollenkaan tai vain osittain. Jos selain tukee jotain ominaisuutta vain osittain, ominaisuuden eteen pitää kirjoittaa selainkohtaiset etuliitteet (prefix). Esimerkiksi *transform* on ominaisuus, jota vanhat selaimet tukevat vain etuliitteen avulla. Jos jotakin elementtiä haluttaisiin kääntää kymmenen astetta, sille tulisi antaa taulukon 1 määrittelyt.

Taulukko 1. Esimerkki tyylimäärittelyissä käytettävistä etuliitteistä.

Määrittely	Selain
<code>-moz-transform:rotate(10deg);</code>	Firefox 3.5-15.x
<code>-ms-transform:rotate(10deg);</code>	Internet Explorer 9.0
<code>-o-transform:rotate(10deg);</code>	Opera 10.5-14.x
<code>-webkit-transform:rotate(10deg);</code>	Chrome 4.0-35.x, Safari 3.2-8.x, Opera 15.0-22.x
<code>transform:rotate(10deg);</code>	Modernit selaimet

Eri selaimien lisäksi sivustopohjan responsiivisuutta on testattu eri laitteilla. Testausvaiheessa käytettiin kolmea eri laitetta: Android-pohjaista älypuhelinta, iOS-pohjaista tablettia sekä Windows-pohjaista kannettavaa tietokonetta ja siihen yhdistettyä isompaa näyttöä. Aloitin testauksen tietokoneella käyttämällä selaimen laitetilaa (device mode). Laitetilan avulla pystyin näkemään miltä sivustopohja näyttää yleisimmillä laitteiden viewportien koilla. Tietokoneella testattaessa sivustopohjassa ei ilmennyt ongelmia, kuten ei myöskään puhelimella testattaessa. IPadilla tuli kuitenkin ongelma vastaan nostojen kanssa. Nostojen kuvat venyivät liikaa vaakasuorassa. Ongelman korjaus oli onneksi helposti ja nopeasti hoidettu ja tämän jälkeen sivustopohja toimi taas virheettömästi.

YHTEENVETO

Tämän opinnäytetyön tavoitteena oli syksyn 2015 aikana suunnitella responsiivinen sivustopohja ja käyttää sen toteuttamiseen mobile first -periaatetta. Aikataulussa pysyttiin ja sivustopohja otettiin käyttöön marraskuussa 2015.

Opinnäytetyön tuloksena syntyi uusi, mobiililähtöinen sivustopohja R2-sisällönhallintajärjestelmään. Sivustopohjan sisällön ja ulkoasun toteutuksessa otettiin toimeksiantajan asettamat toiveet huomioon. Mobile first -periaatteen käyttäminen ulkoasun määrittelyyn tuntui aluksi työläämmältä kuin perinteisempi desktop first -menetelmä. Uuden ajattelutavan omaksuminen kuitenkin kannattaa, koska tyylimäärittelyjä ja toiminnallisuutta on aina helpompi alkaa lisäämään kuin poistamaan.

L-Template sai esittelytilaisuudessaan toimeksiantajalta positiivista palautetta sekä muutaman parannusehdotuksen, jotka korjattiin sivustopohjaan. Jatkokehitysideana on parantaa sivustopohjaa tuomalla siihen uusia käyttöliittymäelementtejä ja toiminnallisuutta sekä korjata sen käytössä mahdollisesti ilmenneitä ongelmia.

LÄHTEET

Aarabi, P. 2013. 5 tips for creating great mobile app user interfaces. VentureBeat. Viitattu 10.10.2015 <http://venturebeat.com/2013/04/08/5-tips-for-creating-great-mobile-app-user-interfaces/>

Akamai Technologies 2009. Akamai Reveals 2 Seconds As The New Threshold Of Acceptability For Ecommerce Web Page Response Times. Viitattu 3.10.2015 <https://www.akamai.com/us/en/about/news/press/2009-press/akamai-reveals-2-seconds-as-the-new-threshold-of-acceptability-for-ecommerce-web-page-response-times.jsp>

Barsby, A. 2015. Why Google Prefers Responsive Web Design. Xanthos Blog. Viitattu 26.9.2015 <http://www.e-xanthos.co.uk/blog/why-google-prefers-responsive-web-design/>

Baumann, D. 2015. Why you've got to start practicing mobile-first development. Flywheel. Viitattu 4.10.2015 <https://getflywheel.com/layout/start-practicing-mobile-first-development/>

Byers, J. 2015a. Mobile Responsive Design 101. Copyblogger. Viitattu 23.08.2015 <http://www.copyblogger.com/mobile-responsive-design-101/>

Byers, J. 2015b. Three Reasons a "Mobile First" Philosophy is Critical to Achieving Your Business Goals. Copyblogger. Viitattu 10.10.2015 <http://www.copyblogger.com/mobile-first/>

Code My Views 2016. Mobile First Design: Why It's Great and Why It Sucks. Viitattu 30.1.2016 <https://codemyviews.com/blog/mobilefirst>

Compuware 2011. New Study Reveals the Mobile Web Disappoints Global Consumers. Viitattu 3.10.2015 <http://investor.compuware.com/releasedetail.cfm?releaseid=592528>

Cousins, C. 2014. Make the Most of Your Design with a Hero Header. Designmodo. Viitattu 30.1.2016 <http://designmodo.com/hero-headers/>

Creative Bloq 2012. The 10 principle of mobile interface design. Viitattu 1.12.2014 <http://www.creativebloq.com/mobile/10-principles-mobile-interface-design-4122910>

Fielding, J. 2014. Beginning Responsive Web Design with HTML5 and CSS3. New York City: Apress.

Frost, B. 2011. Mobile-First Responsive Web Design. Viitattu 3.10.2015 <http://bradfrost.com/blog/web/mobile-first-responsive-web-design/>

Gharazian, A. 2014. The Pros and Cons of Responsive Web Design vs. Mobile Website vs. Native App. Designmodo. Viitattu 12.9.2015 <http://designmodo.com/responsive-design-vs-mobile-website-vs-app/>

Google-optimointi.com 2016. Hakukoneoptimointi kasvattaa kävijämäärää. Viitattu 30.1.2016 <http://www.google-optimointi.com/>

jQuery.com 2016. What is jQuery? Viitattu 30.1.2016 <https://jquery.com/>

Kajaanin ammattikorkeakoulu 2015. Toimintatutkimus. Viitattu 11.10.2015 <https://www.kamk.fi/opari/Opinnaytetyopakki/Teoreettinen-materiaali/Tukimateriaali/Toimintatutkimus>

Knight, K 2011. Responsive Web Design: What It Is And How To Use It. Smashing Magazine. Viitattu 26.9.2015 <http://www.smashingmagazine.com/2011/01/guidelines-for-responsive-web-design/>

Kuopion yliopisto 2016. Verkkohjelmointi. Viitattu 30.1.2016 <http://www.cs.uku.fi/~mhasine/VOH/Luennot/lu7.html>

Less.org 2016. Getting Started. Viitattu 30.1.2016 <http://lesscss.org/>

Makino, T. & Phan, D. 2015. Rolling out the mobile-friendly update. Google Webmaster Central Blog. Viitattu 26.9.2015 <http://googlewebmastercentral.blogspot.fi/2015/04/rolling-out-mobile-friendly-update.html>

May, A. 2015. How Google's mobile-friendly update will affect you (and what you can do about it). Webdesigner Depot. Viitattu 26.9.2015 <http://www.webdesignerdepot.com/2015/04/how-googles-mobile-friendly-update-will-affect-you-and-what-you-can-do-about-it/>

Metamonks 2015. Why 'mobile-first' is the new 'responsive'. Viitattu 4.10.2015 <http://metamonks.com/mobile-first-vs-responsive/>

Peterson, C. 2014. Learning Responsive Web Design. Sebastopol: O'Reilly Media, Inc.

Robinson, J. 2015. Mobile SEO: Responsive Design vs. Separate Mobile Site vs. Dynamic Serving. ClickSeed. Viitattu 12.9.2015 <http://www.clickseed.com/seo/responsive-design-vs-separate-mobile-site-vs-dynamic-serving>

Rocheleau, J. 2015. Responsive Web Layouts for Mobile Screens: Intro, Tips and Examples. Hongkiat. Viitattu 12.9.2015 <http://www.hongkiat.com/blog/responsive-for-mobile-screens/>

Sharkie, C. & Fisher, A. 2013. Jump Start Responsive Web Design. Collingwood: SitePoint Pty Ltd.

StatCounter 2015. Comparison from 2011 to 2014. Viitattu 4.10.2015 <http://gs.statcounter.com/#all-comparison-ww-yearly-2011-2014>

W3schools 2016a. Responsive Web Design – The Viewport. Viitattu 30.1.2016 http://www.w3schools.com/css/css_rwd_viewport.asp

W3schools 2016b. CSS3 @media Rule. Viitattu 30.1.2016 http://www.w3schools.com/cssref/css3_pr_mediaquery.asp

W3schools 2016c. CSS Units. Viitattu 30.1.2016 http://www.w3schools.com/cssref/css_units.asp

Zurb 2015. Mobile First. Viitattu 3.10.2015 <http://zurb.com/word/mobile-first>