

Base C6 -julkaisujärjestelmän käyttöliittymä uudistus

Henri Sutinen

Opinnäytetyö
Joulukuu 2015

Mediatekniikan koulutusohjelma
Tekniikan ja liikenteen ala



Tekijä(t) Sutinen, Henri	Julkaisun laji Opinnäytetyö, AMK	Päivämäärä 4.12.2015
	Sivumäärä 54	Julkaisun kieli Suomi
		Verkojulkaisulupa myönnetty: x
Työn nimi Base C6 -julkaisujärjestelmän käyttöliittymä uudistus		
Tutkinto-ohjelma Mediatekniikan koulutusohjelma		
Työn ohjaaja(t) Pasi Manninen		
Toimeksiantaja(t) Valu Digital Oy		
<p>Tiivistelmä</p> <p>Opinnäytetyön aiheena oli uudistaa Valu Digital Oy:n Base C6 -julkaisujärjestelmä, hyödyntäen Twitter Bootstrap 3 -sovelluskehystä. Opinnäytetyössä tarkasteltiin käyttöliittymäsuunnittelua ja käytettävyyttä mm. käyttäjäkokemusten ja käytettävyyssstandardien pohjalta. Käyttöliittymän teknisessä uudistuksessa hyödynnettiin teoriaa sekä valitun sovelluskehysten luokkia ja komponentteja. Järjestelmän kaikki eri moduulit ja näkymät käännettiin käyttämään Bootstrap-sovelluskehysten luokkia, käyttöliittymäelementtejä ja toiminnallisuuksia.</p> <p>Käyttöliittymän parantamisessa keskityttiin myös yksittäisten komponenttien toiminnallisuuksien parantamiseen. Järjestelmää muokattiin vastaamaan paremmin nykyaikaisia käytettävyyssstandardeja, kuten tekemällä käyttöliittymästä responsiivinen. Responsiivisuudessa keskityttiin erityisesti parantamaan tablet-näkymiä, sillä niiden käyttö on asiakkaiden keskuudessa lisääntynyt huomattavasti.</p> <p>Opinnäytetyössä myös vertailtiin toiminnallisuuksia vanhan ja uuden version välillä, jolla pystyttiin osoittamaan muutoksien hyödyt. Opinnäytetyössä toteutettu Base C7 -versio on käytettävyydeltään nykyaikaisempi ja parempi, kuin vanha versio. Käytettävyyden parantaminen tekemällä järjestelmästä responsiivisen, yhdenmukaistamalla komponentteja ja käyttämällä Bootstrap-sovelluskehysten tarjoamia työkaluja. Jatkokehityksenä julkaisujärjestelmää tullaan vielä ennen julkaisua viimeistelemaan mm. responsiivisuuden ja editorin osalta syvällisemmin.</p>		
<p>Avainsanat (asiasanat)</p> <p>Julkaisujärjestelmä, käytettävyyden, käyttöliittymä, käyttöliittymä uudistus, Valu Digital</p>		
Muut tiedot		

Description

Author(s) Sutinen, Henri	Type of publication Bachelor's thesis	Date 04.12.2015
	Number of pages 54	Language of publication: finnish
		Permission for web publication: X
Title of publication UI revision for Base C6 content management system		
Degree programme Media Engineering		
Supervisor(s) Manninen, Pasi		
Assigned by Valu Digital Oy		
<p>Description</p> <p>The thesis was assigned by Valu Digital Inc. The topic of thesis was to conduct a revision for Valu Digital's independent and closed content management system, Base C6' user interface, by utilizing Twitter Bootstrap 3 framework. The thesis was based on research that involving user interface design, usability, user experience and usability standards. Technical user interface revision exploits the made research while changing the old views and elements by using the new classes and components of the new framework, Bootstrap3. All system modules and views were translated into the new framework.</p> <p>The thesis also includes also improvements to the functionality of the user interface. The system was modified to suit modern UI standards better, and be responsive and usable on tablet computers. The usability changes included the use of modal windows instead of pop-ups, for example. Comparisons between old and new version were made.</p> <p>Usability was improved by unifying components and design and also taking advantage of Bootstrap 3's improved usability and the design of the user interface. The user interface is now responding to design standards of 2015 with sleeker look and design. The content management system is not completely ready. Future development will mainly focus on responsiveness and general finalizing of the Base C6 content management system. Large scale testing of the new version will also be performed in the future.</p>		
<p>Keywords (subjects)</p> <p>Content management system, usability, user interface, user interface revision, Valu Digital</p>		
Miscellaneous		

Sisältö

Käsitteet	6
1 Työn lähtökohdat	9
1.1 Taustat.....	9
1.2 Toimeksiantaja	9
1.3 Tavoitteet	10
2 Base C6 -julkaisujärjestelmä	10
2.1 Historia	10
2.2 Ominaisuudet ja toiminnot	11
2.2.1 Tietoturva.....	11
2.2.2 Käytettävyys.....	11
2.2.3 Toiminnot.....	12
2.2.4 Hakukoneoptimointi ja analytiikka	14
2.2.5 Versionhallinta	14
2.3 Uuden version tavoitteet ja lähtökohdat.....	15
2.4 Kehitysympäristö.....	15
3 Käyttöliittymäsuunnittelu	17
3.1 Käyttöliittymäsuunnittelu yleisesti	17
3.2 User Interface (UI).....	18
3.3 Käytettävyys	20
3.4 User experience (UX)	21
4 Käyttöliittymä uudistus	26
4.1 Tavoitteet	26
4.2 Twitter Bootstrap	27
4.3 Käyttöliittymäkomponentit.....	29
4.3.1 Yleistä	29
4.3.2 Kirjautumissivu.....	30
4.3.3 Navigaatio	31
4.3.4 Muokkausnäkyvä	32
4.3.5 Päänäkymä.....	33
4.4 Käyttöliittymäkirjastojen päivitys	34
4.5 Bootstrap ja teemat	35

5 Teknillinen toteutus	36
5.1 Moduulien päivitys.....	36
5.2 Käyttöliittymän toiminnallisuuksien muutokset.....	39
5.2.1 Modaali-ikkunat	39
5.2.2 Päivämäärävalitsinkomponentit	42
5.3 CSS-preprosessointi Compassilla ja SASS:lla	44
5.4 Responsiivisuus	45
6 Tulokset	47
7 Pohdinta	48
Lähteet	52
Kuviot	
Kuvio 1. Base-julkaisujärjestelmän näkymän rakenne.....	12
Kuvio 2. Yaml-tiedoston luonti PuPHPetilla.	17
Kuvio 3. Bootstrapin painiketyylit noudattavat perinteisiä käytettävyyssstandardeja.	18
Kuvio 4. Käyttäjäkokemus muodostuu useammasta pienemmästä osa-alueesta. (Harvey, Amy. 2013.)	22
Kuvio 5. Käyttäjäkokemuksen rakentuminen. (Roto, V., Law, E., Vermeeren, A. & Hoonhout, J. 2011.)	24
Kuvio 6. Käyttäjäkokemuksen aikajänne. (Roto, V., Law, E., Vermeeren, A. & Hoonhout, J. 2011.).....	25
Kuvio 7. Twitter Bootstrap kehitettiin alunperin Twitter-mikroblogipalvelua varten.....	27
Kuvio 8. Tabs-komponentit ja hakukomponentti	29
Kuvio 9. Base C6- ja C7-version kirjautumisnäkymät.	31
Kuvio 10. Yllä C6:n navigaatio ja alla uusi C7-navigaatio.	32
Kuvio 11. Muokkausnäkymän rakenne.	33
Kuvio 12. C6- ja C7-version päänäkymät vierekkäin.	34
Kuvio 13. Inspinia Bootstrap-teema.....	35
Kuvio 14. C6- ja C7 version viesti-ikkuna ja viestimodaali-ikkuna	40

Kuvio 15. Viestimodaali-ikkunan toiminta.	41
Kuvio 16. Vanha jQuery-datepicker ja uusi Bootstrap-datetimepicker-komponentti.....	42
Kuvio 17. C7-versio iPadilla.	46

KÄSITTEET

Datepicker

Päivämäärävalitsin (engl. *datepicker*) on käyttöliittymäkomponentti, jonka avulla käyttäjän on helpompi syöttää päivämäärätietoja. Esimerkiksi tapahtuman alkupäivämäärän syöttämiseen kannattaa käyttää päivämäärävalitsinta. Se avaa yleensä pienen kalenterin, josta käyttäjän on helppo selata haluttu päivämäärä.

Framework

Sovelluskehys, joka tarjoaa rungon esimerkiksi web-sovellukselle. Sovelluskehys sisältää toiminnallisuuksia ja valmiita komponentteja, jotka helpottavat ja auttavat kehitystyössä. Vähentää esimerkiksi ohjelman osien uudestaan kirjoitusta ja nopeuttaa kehitystyötä.

Iframe

Iframe on HTML-elementti, jonka avulla HTML-sivun sisään voidaan upottaa toinen HTML-sivu kehyksen sisään.

Käyttöliittymäkomponentti

Komponentti on pieni osa käyttöliittymää esimerkiksi painike, muropolku, sivutus, tekstikenttä tai valintaruutu.

Modaali-ikkuna

Web-sivua normaalisti selaillessa siirrytään sivulta toiselle, kun halutaan näyttää eri sisältöjä. Modaali-ikkunan avulla voidaan kuitenkin näyttää sisältöä sivua vaihtamatta. Modaali-ikkuna aukeaa sivun päälle, ja sen voi myös sulkea ja palata samalle sivulle. Modaali-ikkunassa voidaan esimerkiksi näyttää yhteydenottolomake tai avata kuva suuremmaksi.

MVC

Model-view-controller eli malli-näkymä-käsittelijä. Ohjelmistoarkkitehtuurityyli, jossa ohjelma jaetaan kolmeen osaan: malleihin, näkymiin ja käsittelijöihin. Malli hoitaa tietojen tallentamisen ja käsittelyn, näkymät määräävät ulkoasun, ja käsittelijät hoitavat käyttäjän toiminnot ja muuttavat malleja ja näkymiä käyttäjien tekemien toimintojen perusteella. Malleja, näkymiä ja käsittelijöitä eli ohjelman osa-alueita voi muokata, ylläpitää ja testata ilman, että ne olisivat toisistaan riippuvaisia.

SASS

Lyhenne tulee sanoista Syntically Awesome Stylesheets. SASS on CSS-preprosessointi työkalu, joka mahdollistaa esimerkiksi muuttujien käytön ja vähentää luokka- ja id-valitsijoiden toistoa CSS-tyylitiedostoissa. SASSilla luodaan joko SCSS tai SASS-tyylitiedostoja, jotka sitten käännetään perinteisiksi CSS-tyylitiedostoiksi Compass -työkalulla.

SVN

Subversion eli lyhennettynä SVN on versionhallintajärjestelmä. SVN mahdollistaa esimerkiksi usean henkilön työskentelyn saman projektin parissa ja samojen tiedostojen yhtäaikaista muokkaamista. Näin tiedostot pysyvät kaikilla muokkaajilla ajan tasalla eikä ylikirjoituksia tai muuta datan katoamista tapahdu.

Twitter Bootstrap

Avoimen lähdekoodin sovelluskehys (framework), jota voidaan käyttää hyödyksi rakentamassa web-sivustoja tai web-sovelluksia. Sisältää mm. komponenttikirjaston, responsiivisen grid-järjestelmän ja JavaScript-toiminnallisuuksia.

UI

UI eli user interface, käyttöliittymä.

UX

UX eli user experience, käyttäjäkokemus.

WYSIWYG

”What you see is what you get” eli mitä näet tekstieditorissa, on se, miltä sivu tulee näyttämään julkaistuna valmiina internetselaimessa.

Zend Framework

Avoimen lähdekoodin PHP-sovelluskehys, joka perustuu olio-ohjelmointiin ja MVC-arkkitehtuuriin. Zend Framework -sovelluskehystä voidaan käyttää web-sovellusten taustalla. Zend Frameworkia käytetään myös Base C6 -julkaisujärjestelmän taustalla.

1 TYÖN LÄHTÖKOHDAT

1.1 Taustat

Opinnäytetyö pohjautuu syksyn 2014 ja kevään 2015 välillä suoritettuun harjoitteluun jyvaskyläläisessä verkkopalveluyrityksessä nimeltä Media Cabinet Oy. Harjoittelun aikana päädyttiin ratkaisuun, että opinnäytetyö tehdään myös Media Cabinetille, nykyiselle Valu Digital Oy:lle. Yrityksestä ehdotettiin opinnäytetyön aiheeksi yrityksen itse kehittämänsä julkaisujärjestelmän, Base C6:n uuden version toteuttamista, joka on käytännössä käyttöliittymä uudistus sisältäen ulkoasun ja toiminnallisuuksien muutoksia.

1.2 Toimeksiantaja

Valu Digital Oy on vuonna 1997 perustettu jyvaskyläläinen verkkopalveluita tarjoava yritys. Valu Digital tarjoaa asiakkaille ratkaisut verkkoviestintään, verkkoliiketoiminnan pyörittämiseen ja sähköiseen asiointiin. Yritys tunnettiin aiemmin nimellä Media Cabinet Oy. Keväällä 2015 yritys kävi läpi brandin uudistuksen ja vaihtoi nimensä nykyiseen muotoon Valu Digital Oy:ksi. Samalla WordPress-toteutuksiin erikoistunut tytäryhtiö Versi Oy liitettiin osaksi Valu Digital Oy:ta. Mobiilisovelluksien parissa toimiva tytäryhtiö Showell Applications Oy muutti myös nimensä Valu Mobile Oy:ksi. Tällä hetkellä yrityksen toimisto sijaitsee Jyväskylässä, Technopoliksen Innova-rakennuksessa ja yrityksessä työskentelee 22 henkilöä. (Yritys n.d.)

Valu Digital tekee pääasiassa verkkosivustoprojekteja, mutta on lisäksi erikoistunut verkkovuosikertomusten tekoon, jotka julkaistaan verkkosivustona tavallisen PDF-version sijaan. Verkkosivuston sisällön voi halutessaan tulostaa kokonaan PDF-tiedostona, joka on generoitu kokonaan verkosta julkaistavasta sivustosta.

Valu Digital on myös panostanut suuresti omaan tuotekehitykseen, jonka päätuote on oma itsenäinen Base C6 -julkaisujärjestelmä. Base C6 on eritoten suuren luokan web-sivustoprojekteihin suunniteltu julkaisujärjestelmä, jonka käytettävyyteen ja tietotur-

vaan on panostettu. Lisäksi yritys käyttää myös muissa projekteissa avoimen lähdekoodin WordPress -julkaisujärjestelmää. (Teknologiat n.d.)

1.3 Tavoitteet

Opinnäytetyön aiheeksi valikoitui Valu Digitalin oman julkaisujärjestelmän Base C6:n uuden version käyttöliittymä uudistus. Uusi versio tulee olemaan nimeltään Base C7. Harjoittelun aikana päädyttiin tähän aiheeseen, koska se oli yritystä hyödyttävä ja opinnäytetyön tekijälle mielenkiintoinen. Yrityksen julkaisujärjestelmän C6-versio oli varsinkin ulkoasullisesti ja toiminnallisuuksien osalta jo vanhentunut. Yrityksessä nähtiin tarve käyttöliittymän uudistamiselle, jotta Base C6 -julkaisujärjestelmä vastaisi nykypäivän vaatimuksia ja standardeja. Yleisesti opinnäytetyössä oli tarkoitus käydä läpi käyttöliittymäsuunnittelua ja käytettävyyttä myös teorian kautta ja soveltaa hankittua tietoa teknisessä toteutuksessa.

Opinnäytetyössä tavoitteena oli rakentaa uusi julkaisujärjestelmän versio Twitter Bootstrap -sovelluskehiksen päälle. Bootstrap valikoitui käyttöliittymä uudistuksen pohjalle suosionsa, soveltuvuutensa ja hyvänlaisen yksinkertaisuuden vuoksi. Järjestelmän ulkoasu oli tarkoitus uudistaa Bootstrapin omilla komponenteilla. Esimerkiksi kaikki painikkeet, ikonit, tekstikentät yms. Komponentit vaihdettiin. Muita julkaisujärjestelmän muutoskohtia olivat CSS-preprosessointi Compass/SASS-tekniikoilla, käyttöliittymäkirjastojen päivittäminen uusimpaan tuettuun versioon (mm. jQuery), tuki käyttöliittymän teemoittamiselle ja teemoituksen dokumentoiminen. Lisäksi nykyisen version moduulien päivittäminen yhteensopiviksi kuului osaksi opinnäytetyötä.

2 BASE C6 -JULKAISUJÄRJESTELMÄ

2.1 Historia

Base C6 on Valu Digital Oy:n kehittämä julkaisujärjestelmä. Se sai alkunsa vuonna 2000, kun Valu Digital julkaisi iManager-nimisen tuotteen. Tuote kantoi iManager-nimeä ai-

na vuoteen 2005, jolloin julkaisujärjestelmän nimi muuttui nykyiseen muotoonsa: Base. Nykyinen versio julkaisujärjestelmästä on Base C6, joka julkaistiin vuonna 2009. Tavoitteena on saada versio C7 ulos alkuvuoden 2015 aikana. (Tolvanen 2010.)

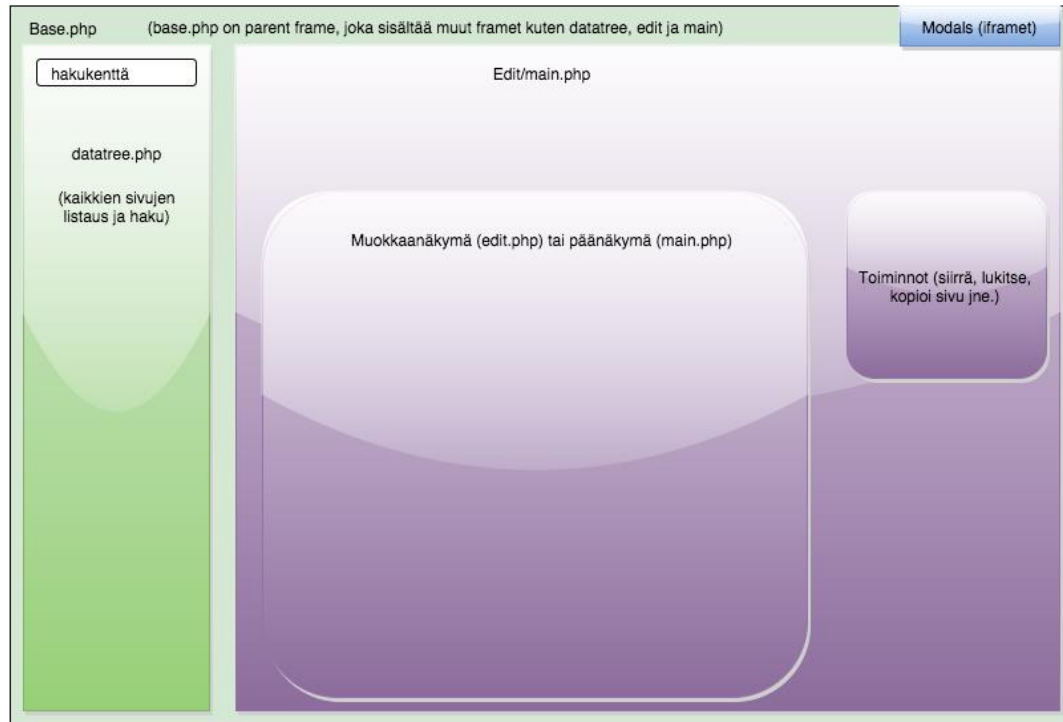
2.2 Ominaisuudet ja toiminnot

2.2.1 Tietoturva

Base C6 -julkaisujärjestelmä on kehitetty Zend Framework -sovelluskehityksen päälle. Tietoturvaan on panostettu mm. automaattisilla tietoturvapäivityksillä, Zend Frameworkin työkaluilla sekä ohjelmisto- ja rautapuolen palomuuereilla. Lisäksi suljetun järjestelmän suurimpia etuja on se, että kaikki lisäosat ovat itse kehitettyjä. Kun kaikki moduulit ja toiminnot ovat tällaisia, tiedetään paremmin, mitä mikäkin järjestelmän komponentti tekee. Tämä lisää tietoturvaa esimerkiksi verrattuna WordPressiin, jonka ulkoiset lisäosat voivat altistaa sivuston hyökkäyksille tai tehdä sivuston tietoturvaan reikiä, joista käyttäjät eivät tiedä mitään. WordPressissä voidaan esimerkiksi hyvinkin törmätä tilanteeseen, jossa lisäosan päivitys luo järjestelmään haavoittuvuuden, jota sitten hakkerit voivat hyväksikäyttää.

2.2.2 Käytettävyys

Käytettävyys on Base C6:ssa johtavassa osassa julkaisujärjestelmän kehitystä. Julkaisujärjestelmän käyttöliittymästä on tehty yksinkertainen, selkeä ja helposti omaksuttava. Komponentit, kuten haku ja navigaatio, on asemoitu ns. standardoiduille paikoille, jolloin järjestelmän omaksuminen on käyttäjälle paljon helpompaa ja nopeampaa. Base C6:ssa eri toimintojen käyttö on pyritty pitämään samanlaisena kaikissa järjestelmän eri näkymissä. Tällöin käyttäjä sisäistää helposti myös uusien ominaisuuksien käyttöperiaatteen. Base C6 perustuu MVC-arkkitehtuuriin eli järjestelmä koostuu erilaisista moduuleista, näkymistä ja kontrollereista. Base C6:n peruskäyttöliittymä koostuu kolmesta eri näkymästä: base frame eli äitikehys, datatree-kehys ja edit- tai main-kehys (ks. kuvio 1).



Kuvio 1. Base C6 -julkaisujärjestelmän näkymän rakenne.

Moduleja eli komponentteja on järjestelmässä useita erilaisia muun muassa FormBuilder, jolla voidaan luoda ja hallita lomakkeita. Uusien moduulienkin luominen on mahdollista asiakkaiden tarpeiden mukaisesti. Base C6 sisältää oletusmoduuleina sivut, kävijäseurannan, Filebankin (tiedostopankki kuville ja muille tiedostoille), FormBuilderin (lomakkeet), snippetit (jaetut sisällöt), artikkelit ja käyttäjähallintamoduulin.

2.2.3 Toiminnot

Base C6:n käyttöliittymä on tehty alun perin englannin kielellä, mutta käyttäjä voi valita, käyttääkö järjestelmää suomeksi vai englanniksi. Julkaisujärjestelmään on tehty tuki myös muillekin kieliversioille sekä laajasti eri merkistöille. Tämä mahdollistaa esimerkiksi venäjän ja kiinankieliset versiot sivustosta, joita pystyy hallitsemaan helposti yhdestä näkymästä. Uudessa C7-versiossa päädyttiin pitämään vanha kielivalintakomponentti, jossa käyttäjä voi valita haluamansa kielen kirjautumisen yhteydessä. Tämä helpottaa esimerkiksi testausta ja eri kielten välillä vaihtamista, kun kieltä ei esimerkiksi haeta suoraan selaimen asetuksista.

Base C6-julkaisujärjestelmässä on laaja versionhallinta. Käyttäjä voi luoda sivuista eri versioita ja tallentaa niitä versionhallintaan, ajastaa niiden julkaisun haluttuun ajankohtaan tai siirtää niitä roskakoriin. Roskakori-toiminnon avulla käyttäjä ei voi tehdä mitään peruuttamatonta, sillä sivut ovat palautettavissa roskakorista takaisin Sivut-moduulin käyttöön. Base C6 tarjoaa myös tiedostoille versionhallinnan, jonka avulla on helppo luoda mm. kuvista eri versiot erilaisiin käyttötarkoituksiin.

Base C6:ssa on käytössä avoimen lähdekoodin TinyMCE-editori, jota on muokattu julkaisujärjestelmän omiin tarpeisiin. Editori on niin sanottu WYSIWYG-editori, jolloin sivuja muokatessa ne näyttävät samalta, kuin miltä ne tulevat näyttämään sivustolla julkaistuna. Julkaisujärjestelmään voi luoda monta erilaista sivupohjaa eri tarpeisiin. Editori sisältää valmiiksi suuren määrän erilaisia komponentteja, kuten taulukot, palstoitukset, YouTube-videoiden upotuksen, Google Mapsin, erilaiset teksti- ja linkkityylit sekä lomakkeet. Käyttäjä voi lisätä näitä komponentteja sivuilleen suoraan editorin oikeasta reunasta löytyvästä valikosta.

Kuvien lisäämiseen ja niiden skaalaukseen on kiinnitetty paljon huomiota julkaisujärjestelmässä. Riippumatta siitä minkä kokoisen kuvan lisää, kuvat skaalautuvat oikeaan kokoon. Julkaisujärjestelmän editori tarjoaa mahdollisuuden muuttaa esimerkiksi kuvan mittoja, tasausta ja linkkejä. Julkaisujärjestelmän LiveEdit-toiminnolla valitun kuvan käyttäjä voi rajata kuvan haluamansa kuvasuhteeseen ja kokoon helposti suoraan julkaisujärjestelmässä.

Valu Digital Oy:n julkaisujärjestelmä sisältää laajan käyttäjähallinnan, joka on tärkeässä roolissa suurempien organisaatioiden verkkopalvelujen toteutuksessa. Käyttäjähallinnassa on useita tasoja, joilla käyttöoikeuksia voi määrittellä. Luonnollisesti käyttöoikeudet voi määrittää vapaasti myös henkilökohtaisesti. Käyttäjille tai käyttäjäryhmille voi antaa oikeuksia esimerkiksi eri moduuleihin, sivuihin ja artikkeleihin.

2.2.4 Hakukoneoptimointi ja analytiikka

Hakukoneoptimointi on nykypäivänä erittäin tärkeää. Sivustoja optimoidaan, jotta ne näkyisivät hakukoneissa mahdollisimman korkealla hakutuloksissa ja asiakas löytäisi yrityksen sivut ensimmäisenä ennen kilpailevia yrityksiä. Base C6:ssa koko sivuston ja yksittäisen sivun metatietoja voi hallita suoraan. Base C6:ssa on mahdollista esimerkiksi määrittää yksinkertaisempia sivupolkuja sivuille, riippumatta sivustohierarkiasta. Lisäksi käytössä on Googlen Analytics -työkalut, joilla sivustoon liittyviä erilaisia statistiikkoja on helppoa seurata.

2.2.5 Versionhallinta

Apache SVN on ohjelmistojen versionhallintaan kehitetty työkalu. Sillä voidaan päivittää muutoksia projektin versioon, jota saattaa työstää monta henkilöä samanaikaisesti. Perustoimintaperiaate on, että projektissa on ensiksi otettu käyttöön versionhallinta ja siitä on tehty ensimmäinen versio. Projektiin osallistuvat henkilöt lataavat tämän version omille paikallisille kehitysympäristöille ja voivat tämän jälkeen tehdä muutoksia tiedostoihin.

Kun muutokset suoritetaan versionhallintaan, muut projektiin osallistuvat henkilöt voivat päivittää muutokset omaan paikalliseen versioonsa. SVN:n käyttämiseen yrityksessä on selvät syyt. Se helpottaa yrityksen useamman henkilön työskentelyä yhtäaikaaisesti saman projektin parissa ja varmistaa, että kaikkien tiedostot ovat ajan tasalla, eikä ylikirjoituksia tapahdu. Lisäksi SVN on ollut aiemminkin käytössä vanhemmissa versioissa, joten yrityksessä ei ole nähty syytä luopua sen käytöstä tai vaihtaa sitä toiseen. (Collins-Sussman, Fitzpatrick & Pilato. n.d.)

Base asennettiin ympäristöön, joka rakennettiin kehitystyötä varten. Basesta oli aiemmin tehty versionhallintaan 2.0 alpha -versio, vanhan C6-version pohjalta. Lisäksi kaikista Basen moduuleista oli tehty myös 2.0 alpha -versiot. SVN-versionhallintaa hyödynnettiin heti kehitystyön alusta lähtien, sillä se helpotti suuresti muun muassa opinnäytetyön ohjaajaa seuraamaan projektia.

2.3 Uuden version tavoitteet ja lähtökohdat

Yrityksen johto ja tuotekehityksen johtohenkilöt halusivat lähteä uudistamaan Basea, koska se ei täyttänyt enää nykyajan vaatimuksia käytettävyydeltään ja ulkoasultaan. Baseen edellinen C6-versio julkaistiin vuonna 2009, joten oli luonnollista, että muutoksia tarvittiin järjestelmän nykyaikaistamiseksi. Suurimpina ongelmakohtina nähtiin käyttöliittymän ulkoasun vanhanaikaisuus, toimintojen aukeaminen uusiin selainikkunoihin ja teemoituksen puuttuminen. Myös epäyhtenäiset komponentit eri moduuleissa haluttiin korjata. Uuden version toteuttamisessa haluttiin panostaa eritoten käyttöliittymän ulkoasuun ja käytettävyyteen.

Ulkoasu päädyttiin uudistamaan vaihtamalla käyttöliittymän taustalle Twitter Bootstrap 3 -sovelluskehys. Se tarjoaa kehittäjille erilaisia komponentteja, toiminnallisuuksia ja CSS-tyyliluokkia. Sen päälle voidaan rakentaa web-sivun tai web-sovelluksen käyttöliittymä. C6-version ulkoasu nähtiin vanhanaikaisena, eikä se vastannut nykypäivän standardeja. Ulkoasusta haluttiin nykyaikaisempi, terävämpi ja kevyempi.

Toiminnallisuuden osalta suurin huomio kiinnitettiin uusiin ikkunoihin aukeaviin toimintoihin. Vanhassa versiossa ne nähtiin ongelmallisina, sillä nykypäivänä asiakkaat voivat esimerkiksi päivittää sivujansa tablet-laitteilla, eikä uusi ikkuna ole tällöin käyttäjäystävällisin vaihtoehto. Bootstrap-sovelluskehys tarjosi vaihtoehtoisen tavan esittää sisältöä modaali-ikkunoiden avulla. Sisältö pystyttiin esittämään samassa ikkunassa avaamatta uusia ikkunoita tai välilehtiä. Lisäksi uuden version yksi tavoite oli yhtenäistää järjestelmää ulkoasultaan ja toiminnaltaan. Aiemmassa C6-versiossa käytössä saattoi olla esimerkiksi useaa erilaista päivämääräkomponenttia jQuery UI datepicker -komponentista pelkkään tekstikenttään. Kuvauksen mukaiset epäyhteneväisyydet pyrittiin poistamaan kokonaan uudesta C7-versiosta.

2.4 Kehitysympäristö

Ennen projektin teknisen toteuttamisen aloittamista, valittiin aloituspalaverissa kehitysvaiheessa käytettävät työkalut. Työkaluja tarkasteltiin perusteellisesti, sillä ne vaikuttivat

paljon käyttöliittymä uudistuksen toteuttamisen sujuvuuteen ja jouhevuuteen. Yrityksen työntekijöiden kanssa päädyttiin suosimaan aiemmin hyväksi havaittuja työkaluja. Ne helpottivat kehitystyötä ja uuteen versioon tehtyjen muutosten siirtämistä toisille koneille. Tällä haettiin takaa mahdollisuutta, että muiden työntekijöiden olisi helppoa päivittää muutokset omille työpöytäilleen ja testata tehtyjä muutoksia tarvittaessa nopeasti.

Tärkein osa kehitystyökalujen valintaa oli valita, mihin Base-julkaisujärjestelmän kehitysversio asennettiin ja missä sen kehitystyö tapahtui. Valinta kohdistui virtuaalisten kehityspalvelimien pystytykseen tarkoitettuun Vagrant-työkaluun. Vagrant valittiin käyttöön, koska sillä pystyttiin luomaan nopeasti suorituskykyinen ja helposti muunneltava virtuaalipalvelin. Vagrantilla voidaan luoda helposti virtuaalipalvelimia, jotka ovat keveitä, liikuteltavia ja helppokäyttöisiä. Se toimii yhdessä joko Oraclen Virtualboxin, VMwaren tai AWS:n kanssa.

Vagrantin etu on sen käyttämä yaml-tiedosto, jonka perusteella kehitysympäristö määriteltiin. Tiedosto luotiin ilmaisella PuPHPet-työkalulla, jolla voitiin määritellä kaikki virtuaalikoneeseen halutut asetukset (Ks. kuvio 2). Tällä tiedostolla määritettiin muun muassa virtuaalikoneessa käytetty käyttöjärjestelmä, tietokanta, tietokannan salasanat ja käyttäjätunnukset, käytettävä web-palvelin ja millä virtualisointityökalulla kehitysympäristöä pyöritetään. Vagrant huolehtii koko kehitysympäristön pystyttämisestä tämän tiedoston avulla. (Getting started 2015.)

Local VM Details

Deploy your virtual machine on your local PC.

Deploy to Local Host
The virtual machine will live inside your local PC. [Click here for instructions.](#)

Provider

<input checked="" type="checkbox"/> VirtualBox <small>Windows, OS X & Linux Performance: ★</small>	<input type="checkbox"/> VMWare Fusion <small>OS X only Performance: ★★★</small>
<input type="checkbox"/> VMWare Workstation <small>Windows & Linux Performance: ★★★</small>	<input type="checkbox"/> Parallels Desktop <small>OS X only Performance: ★★★</small>

Distro

<input type="checkbox"/> CentOS 6 x64 <small>PHP 5.6, 5.5, 5.4, HHVM</small>	<input type="checkbox"/> Debian Wheezy 7 x64 <small>PHP 5.6, 5.5, 5.4, HHVM</small>
<input checked="" type="checkbox"/> Ubuntu Trusty 14.04 LTS x64 <small>PHP 7, 5.6, 5.5, HHVM</small>	<input type="checkbox"/> Ubuntu Trusty 14.04 LTS x32 <small>PHP 5.6, 5.5, HHVM</small>
<input type="checkbox"/> Ubuntu Precise 12.04 LTS x64 <small>PHP 5.5, 5.4, HHVM</small>	

IP Address <input type="text" value="192.168.96.101"/>	Hostname <input type="text" value="local.base.fi"/>
Memory <input type="text" value="1024"/>	CPUs <input type="text" value="2"/>

Forwarded Ports

Host Port <input type="text" value="7200"/>	VM Port <input type="text" value="22"/>
---	---

Kuvio 2. Yaml-tiedoston luonti PuPHPetilla.

Muut projektiin osallistuvat henkilöt pystyivät käyttämään samaa tiedostoa rakentaakseen täysin identtisen kehitysympäristön. Vagrantilla pystytetty kehitysympäristö oli riippumaton käyttöjärjestelmästä, käytettiin sitä sitten Mac OSX:lla, Windowsilla tai Linuxilla. Tällöin kaikki projektiin osallistuvat henkilöt tekivät työtä samanlaisella kokoonpanolla ja samoilla riippuvuuksilla.

3 KÄYTTÖLIITTYMÄSUUNNITTELU

3.1 Käyttöliittymäsuunnittelu yleisesti

Käytettävyys (engl. usability) on apuvälineen tai muun valmistetun esineen, palvelun tai ympäristön helppokäyttöisyyttä tietyn tavoitteen saavuttamiseksi. Käytävyydellä voidaan myös viitata helppokäyttöisyyttä mittaaviin menetelmiin sekä

oppiin niistä periaatteista joita soveltamalla tuotteesta, palvelusta tai ympäristöstä saadaan helpokäyttöisempi. (Käytettävyys 2015.)

Käyttöliittymäsuunnittelun ja käytettävyyden teorian tutkiminen oli iso osa opinnäytetyötä, kun suunniteltiin uutta versiota julkaisujärjestelmästä käyttöliittymä uudistuksen perspektiivistä. Käytettävyydellä tarkoitettiin tässä yhteydessä Base-julkaisujärjestelmän helpokäyttöisyyttä. Esimerkiksi miten helposti asiakas löytäisi järjestelmästä haluamansa toiminnot tai miten loogisia niiden asetukset graafisessa käyttöliittymässä (GUI) olisivat. Käyttöliittymäsuunnitteluun liittyy valtavasti huomioonotettavia asioita ja seikkoja, jotta ohjelmasta tai järjestelmästä saadaan helposti käytettävä, nopeasti omaksuttava, looginen ja yleisesti omaksuttuja standardeja noudattava. Yksinkertaisimmillaan käytettävyysstandardi voi olla vaikka värisignaali käyttäjälle: vihreä merkkää onnistunutta toimintoa ja punainen väri taas poistamista tai peruuttamista (Ks. kuvio 3).



Kuvio 3. Bootstrapin painiketyylit noudattavat perinteisiä käytettävyysstandardeja.

Käytettävydestä ihmisen ja tietokoneen välisessä vuorovaikutuksessa käytetään termiä HCI eli Human Computer Interaction. HCI on tutkimusalue, jossa tutkitaan ihmisen ja tietokoneen välistä interaktiota erilaisilla osa-alueilla, joita ovat kognitiivinen psykologia, tietojenkäsittelytiede, käyttöliittymät, käytettävyys, käyttäjäkokemus ja ohjelmistot. Tässä opinnäytetyössä paneudutaan tarkemmin varsinkin käyttöliittymään (UI), käytettävyyteen (usability) ja käyttäjäkokemukseen (UX).

3.2 User Interface (UI)

User Interface, lyhennettynä yleensä UI, tarkoittaa käyttöliittymää. UI on tuotteen, ohjelmiston tai vastaavan kokonaisuuden osa, jolla käyttäjä hallitsee tuotteen toimintaa. Base C7 -tapauksessa UI:lla tarkoitetaan julkaisujärjestelmän näkymää, jossa käyttäjä voi luoda uusia sivuja, julkaista artikkeleita tai määrittää käyttäjäasetuksia. Osa käyttöliitty-

mää on myös tapa, jolla käyttäjä tai asiakas syöttää tietoja tai antaa käskyjä julkaisujärjestelmälle (hiiri ja näppäimistö). UI koostuu interaktioiden eli toimintojen suunnittelusta, visuaalisesta suunnittelusta ja tiedon rakenteesta.

Hyvän UI:n suunnittelussa on monia asioita, jotka tulee ottaa huomioon. Yksi tärkeimmistä on käyttöliittymän komponenttien suunnittelu visuaalisesti ja toiminnallisesti. Käyttäjät ovat nykypäivänä tottuneet yleisesti hyväksytyihin standardeihin, mitä erilaiset symbolit, painikkeet ja värit merkitsevät. Niiden huomioon ottaminen ja käyttäminen käyttöliittymää suunnitellessa helpottaa eri toimintojen suorittamista. Tämä lisää käyttäjätyytyväisyyttä ja parantaa käytön tehokkuutta.

Käyttöliittymän suunnittelussa kannattaa kiinnittää huomiota etenkin komponenttien valintaan ja niiden hyödyntämiseen. Käyttöliittymä sisältää erilaisia komponentteja, jotka voi yleensä lajitella seuraavasti:

- Tiedonsyöttämiseen käytettävät komponentit (painikkeet, valintapainikkeet, radiopainikkeet, vetovalikot, kytkimet, päivämääräkomponentit)
- Navigaatioelementit (muropolut, liukuvalikot, hakukentät, sivutuselementit, aihe sanat ja ikonit)
- Informaaliset elementit (ikonit, latauspalkit, ilmoitukset, viestilaatikat, modaali-ikkunat)
- Komponenttien järjestely loogisesti käyttöliittymän näkymissä. Esimerkiksi haku löytyy yleensä sivun ylälaidasta navigaation kanssa.

Lisäksi on monia ohjenuoria, joita kannattaa hyödyntää käyttöliittymäsuunnittelussa. Tärkeää on pitää käyttöliittymä riittävän yksinkertaisena, luoda siitä johdonmukainen ja käyttää edellä mainittuja yleisiä käyttöliittymäkomponentteja. Tällöin käyttäjät käyttävät tehokkaasti ohjelmaa tai sivustoa. Johdonmukaisuudella tarkoitetaan, että esimerkiksi sivusto toimii kaikilla eri sivuilla tai näkymissä saman logiikan mukaan. Kun käyttäjä on sisäistänyt käyttöliittymän toiminnot yhdellä sivulla, hän pystyy toimimaan samalla logiikalla myös muilla sivuston sivuilla.

3.3 Käytettävyys

”ISO-standardin (ISO9241-11 1998) määritelmässä käytettävyys tarkoittaa vuorovaikutteisen tuotteen tai järjestelmän käytön tarkoituksenmukaisuutta, tehokkuutta, ja miellyttävyyttä” (Hyysalo, S. Oulasvirta, A. Väänänen-Vainio-Mattila, K. 2011). Käytettävyydellä tarkoitetaan esineen, palvelun tai esimerkiksi ohjelmiston helppokäyttöisyyttä tai niiden käytön helpottamista. Käytettävyydellä voidaan myös tarkoittaa palvelun tai ohjelmiston muokkaamista vastaamaan paremmin käyttäjien tarpeita. Käytettävyydellä pyritään vaikuttamaan palvelun tehokkuuteen, efektiivisyyteen ja tyytyväisyyteen.

Käytettävyystutkimuksilla pyritään ymmärtämään käyttäjien toimintaa ja tutkimaan käyttäjien tarpeita, kun he käyttävät järjestelmää tai tuotetta. Iso osa käytettävyyttä, varsinkin web-projektien kanssa, on käyttöliittymäsuunnittelu. Tunnetuin käyttöliittymäsuunnittelun tutkija Jakob Nielsen on jakanut käytettävyyden viideksi eri kokonaisuudeksi. Opittavuus määrittelee, kuinka käyttäjä omaksuu järjestelmän käytön ja kuinka kauan häneltä kuluu aikaa päästä sinuiksi järjestelmän kanssa.

Opittavuus on tärkeää, koska ilman sitä käyttäjä ei pysty tekemään nopeasti haluamiansa toimintoja, hän voi turhautua ja lopettaa käytön. Järjestelmän tai esimerkiksi web-sovelluksen ollessa helposti omaksuttava, käyttäjältä ei kulu kauan aikaa opetteluun. Käyttäjä pystyy käyttämään sivustoa ilman suurempia ongelmia tai opettelukynnystä. Opittavuutta voidaan parantaa käyttämällä esimerkiksi yleisiä käyttöliittymäsuunnittelustandardeja toteutuksessa tai lisäämällä järjestelmään ensimmäisellä käyttökerralla eteen tulevia apuilmoituksia. (Hyysalo, S. Oulasvirta, A. Väänänen-Vainio-Mattila, K. 2011, 103)

Toinen kokonaisuus on tehokkuus. Tehokkuudella mitataan, kuinka nopeasti käyttäjä suoriutuu tehtävistä ja toiminnoista, jotka hän haluaa suorittaa. Tehokkaasti toimivassa järjestelmässä käyttäjä on tietoinen koko ajan, mitä hän on tekemässä ja pystyy ennustamaan järjestelmän toimintaa, mitä seuraavaksi tapahtuu. Tehokkuuteen vaikuttavat esimerkiksi tunnistettavien ja standardien mukaisien ikonien käyttö web-sivuilla, sivujen ja osioiden looginen nimeäminen, hyvä navigaatorakenne ja hyvin jäsennelty sisältö.

Tehokkuutta voidaan lisätä esimerkiksi tarjoamalla edistyneille käyttäjille pikavalintoja ja pikanäppäimiä. (Hyysalo, S. Oulasvirta, A. Väänänen-Vainio-Mattila, K. 2011, 103)

Muistettavuus on kolmas osa käytettävyyden osakokonaisuus Jakob Nielsenin mukaan. Muistettavuudella tarkoitetaan käyttäjän taitoa muistaa järjestelmän käyttö, kun käyttäjä palaa järjestelmän pariin pidemmän tauon jälkeen. Käyttötaidon säilyminen oppimisen jälkeen voi vaihdella sen mukaan, kuinka helppokäyttöinen tai monimutkainen tuote on. (Kurkkia, T. 2007.)

Virheettömyys määrittää taas sen, että käyttäjän pitäisi suoriutua järjestelmän käytöstä mahdollisimman hyvin ilman, että hän tekee virheitä. Julkaisujärjestelmä pitäisi suunnitella siten, että käyttäjä ei tekisi vahingossa virheitä. Esimerkiksi peruskäyttäjällä ei pitäisi olla oikeuksia poistaa pysyvästi mitään. Esimerkiksi sivuja tai artikkeleita poistettaessa ne siirtyvät vain roskakoriin, josta ne on mahdollista palauttaa takaisin. Hyvä tapa ehkäistä virhealttiutta on käyttää varmistavia toimintoja, kun käyttäjä on tekemässä peruuttamatonta toimintoa. Tällöin hyvän käytettävyyden mukaista olisi kysyä käyttäjältä vielä, halutaanko toiminto varmasti suorittaa.

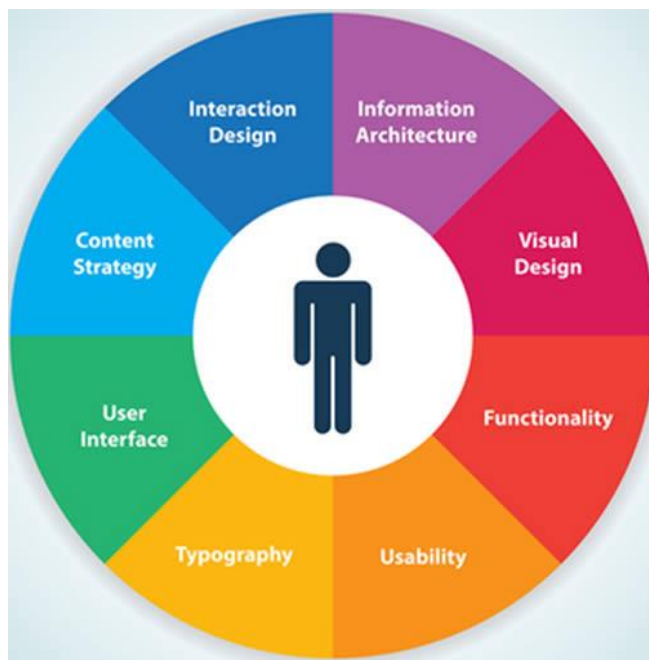
Viimeinen viidestä osakokonaisuudesta on käyttäjätyytyväisyys. Se mittaa, kuinka tyytyväinen käyttäjä on käyttäessään tuotetta. Jos tuote ei ole käytettävyydeltään looginen, helposti omaksuttava, nopea tai helppo, käyttäjä turhautuu ja muuttuu tyytymättömäksi. Lopulta tyytymättömyys voi johtaa tuotteen käytön lopettamiseen.

3.4 User experience (UX)

All aspects of the end-user's interaction with the company, its services, and its products. The first requirement for an exemplary user experience is to meet the exact needs of the customer, without fuss or bother. Next comes simplicity and elegance that produce products that are a joy to own, a joy to use. True user experience goes far beyond giving customers what they say they want, or providing checklist features. In order to achieve high-quality user experience in a company's offerings there must be a seamless merging of the services of multiple disciplines, including engineering, marketing, graphical and industrial design, and interface design. (Nielsen, J. N,d.)

Käyttäjäkokemuksella tarkoitetaan laajempaa kokonaisuutta, jolla käyttäjän kokemusta järjestelmää käytettäessä voidaan mitata. Käyttäjäkokemus koostuu useammasta pienestä kokonaisuudesta, jotka yhdessä muodostavat käyttäjäkokemuksen. Nämä pienemmät osakokonaisuudet ovat

- Informaatioarkkitehtuuri
- Visuaalinen suunnittelu
- Toiminnallisuudet
- Käytettävyys
- Typografia (fontit, fonttikoko, värit)
- Käyttöliittymä
- Sisältöstrategia
- Interaktion suunnittelu (Ks. kuvio 4.)

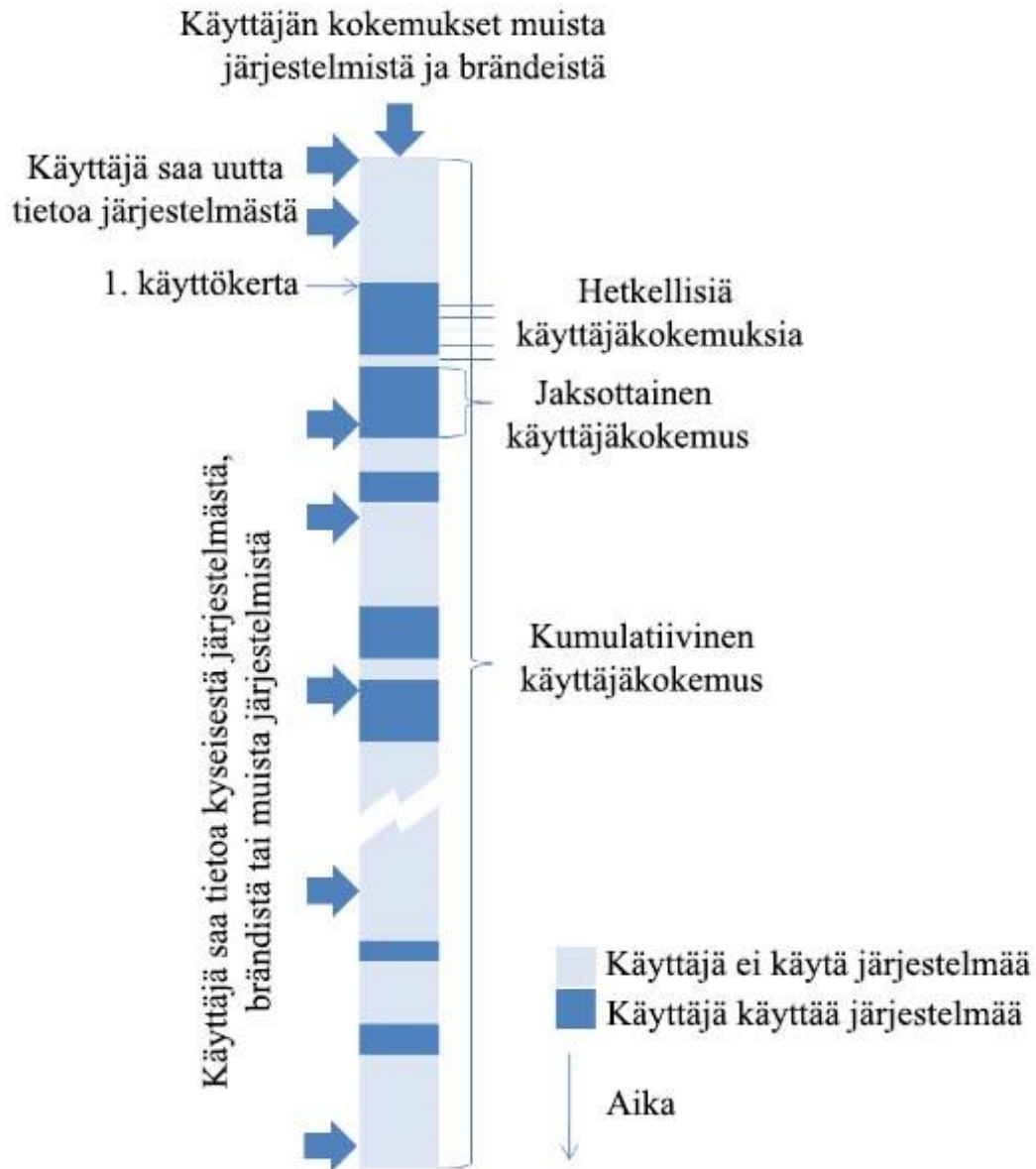


Kuvio 4. Käyttäjäkokemus muodostuu useammasta pienemmästä osa-alueesta. (Harvey, Amy. 2013.)

Käyttäjäkokemus huomioidaan yleensä käyttäjien toimesta vasta, jos kokemus on huono. Parhaat käyttäjäkokemukset ovat niin hyvin suunniteltuja, sulavia ja ohjaavia, että

käyttäjinä emme ajattele käyttäjäkokemusta juurikaan. Hyvä käyttäjäkokemus voi olla esimerkiksi web-sivun tapauksessa jonkin tiedon etsiminen web-sivuilla. Jos tietoa löytyy helposti ja nopeasti, käyttäjäkokemus on onnistunut. Jos taas käyttäjä joutuu painamaan useampaa linkkiä, menemään takaisin ja yrittämään uudelleen löytämättä etsimäänsä, käyttäjäkokemus on huono ja epäonnistunut.

Yrityksen kannalta käyttäjäkokemus on äärimmäisen tärkeä. Jos käyttäjäkokemus epäonnistuu asiakkaan kannalta, tämä voi vaikuttaa pitkään asiakkaan toimintaan. Asiakas turhautuu ja harmistuu ja voi pahimmassa tapauksessa alkaa käyttää kilpailevan yrityksen tuotetta tai palvelua. Käyttäjäkokemussuunnittelu on kuitenkin nähty yleensä vaikeana ja resursseja paljon vievänä asiana. Hyvän käyttäjäkokemussuunnittelijan pitää osata samaistua peruskäyttäjään, jolla ei välttämättä ole samanlaista tieto- tai käyttötaitoa kuin alan ammattilaisilla. Lisäksi pitää osata käytettävyyttä, ihmisen ja tietokoneen vuorovaikutusta. Tärkeää on myös tuntee tuote, jonka kanssa työskentelee, läpikotaisin. (Virta, Paavo. 2014.)(Ks. kuvio 5.)

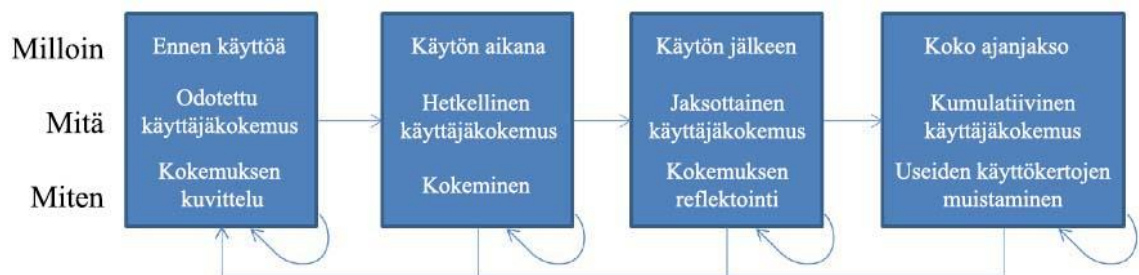


Kuvio 5. Käyttäjäkokemuksen rakentuminen. (Roto, V., Law, E., Vermeeren, A. & Hoonhout, J. 2011.)

Käyttäjäkokemusprosessi rakentuu neljästä eri osasta. Ennen käyttöä käyttäjällä on odotettu käyttäjäkokemus. Tässä vaiheessa käyttäjällä on omia odotuksia järjestelmästä. Miten käyttäjä ajattelee järjestelmän toimivan tai mitä toimintoja hän ajattelee siitä löytyvän. Tähän vaikuttavat myös mm. kokemukset muista vastaavista järjestelmistä, joita käyttäjä on mahdollisesti käyttänyt aiemmin (ks. kuvio 5). Ennen käyttöä käyttäjä luo siis

itselleen tietynlaisen mielikuvan tuotteesta omien odotuksien pohjalta, jotka voivat perustua aiempiin kokemuksiin tai mielipiteeseen brändistä. (Virta, P. 2014.)

Tuotteen käyttöönotossa alkaa myös hetkellinen käyttäjäkokemus. Se jatkuu aina kun käyttäjä käyttää tuotetta. Julkaisujärjestelmän tapauksessa käyttöliittymällä on suuri merkitys. Se vaikuttaa ihmisen ja tietokoneen vuorovaikutukseen eli ns. HCI-kokemukseen, millainen suhde käyttäjän ja järjestelmän välille syntyy. Tämä käyttäjä-tuote -suhde jatkaa kehittymistään ja muodostumistaan jaksottaisessa käyttäjäkokemuksessa jokaisen käyttökerran jälkeen. (Ks. kuvio 6.) Kumulatiivinen eli kokonainen käyttäjäkokemus muodostuu näistä kolmesta muusta osa-alueesta. (Virta, P. 2014.)



Kuvio 6. Käyttäjäkokemuksen aikajänne. (Roto, V., Law, E., Vermeeren, A. & Hoonhout, J. 2011.)

Käyttäjäkokemus voidaan jakaa kolmeen eri osa-alueeseen: kontekstiin, käyttäjään ja tuotteeseen. Kontekstilla tarkoitetaan, miten tuotetta tai palvelua käytetään. Käytetäänkö tuotetta ryhmässä vai yksin ilman muita. Tällä tarkoitetaan sosiaalista kontekstia. Lisäksi kontekstissa on fyysinen aspekti: käytetäänkö tuotetta töissä vaiko junassa. Web-sivuston esimerkissä hyvä fyysinen konteksti olisi responsiivinen sivusto, joka parantaisi junassa käytön kokemusta, kun saatavilla ei ole tietokonetta. Tehtäväkonteksti määrittää, onko samalla aikaa muita suoritettavia tehtäviä, jotka voivat vaikuttaa käyttöön. Tekninen konteksti sisältää tekniset aspektit, jotka voivat vaikuttaa käyttöön (esim. internet- ja sähköverkot).

Käyttäjän toimintaan ja käyttötapaan vaikuttavat mm. seuraavat seikat.

- Käyttäjän odotukset tuotetta kohtaan

- Fyysinen ja henkinen puoli
- Motivaatio käyttää tuotetta

Tärkein osa käyttäjäkokemusta on jo tuote tai palvelu itsessään. Tuotteen ominaisuudet, ulkonäkö, kestävyys, ergonomia ja mielikuva tuotteesta kuten tuotteen brändi muodostavat käyttäjälle kokonaiskuvan tuotteesta.

4 KÄYTTÖLIITTYMÄUUDISTUS

4.1 Tavoitteet

Base-julkaisujärjestelmän C6-versio on otettu käyttöön vuonna 2010. C6-versio ei tule toiminnallisuuksien osalta muuttumaan suuresti C7-versioon nähden. Valulla tehdään Basen kanssa ns. jatkuvaa kehitystä. Uusia toiminnallisuuksia ja ominaisuuksia pyritään lisäämään julkaisujärjestelmään tarpeen ilmetessä tai asiakkaan tarvitessa. Tätä lähtökohtaa ei lähdetty muuttamaan C7-version kanssa, vaan päädyttiin pitämään päivitys lähes täysin käyttöliittymä uudistuksena, joka parantaa käytettävyyttä ja etenkin käyttöliittymän ulkoasua. Tärkeimmät tavoitteet ovat selkeyttää ja yksinkertaistaa käytettävyyttä, uudistaa käyttöliittymän komponentit Twitterin Bootstrapin mukaisiksi, korvata esimerkiksi kaikki kuvaikonit glyphicon-vektorigrafiikkaikoneilla, jotka ovat osa Bootstrap-sovelluskehystä ja uudistaa käyttöliittymän ulkoasu vastaamaan nykyaikaa terävämällä ja modernimmalla ilmeellä.

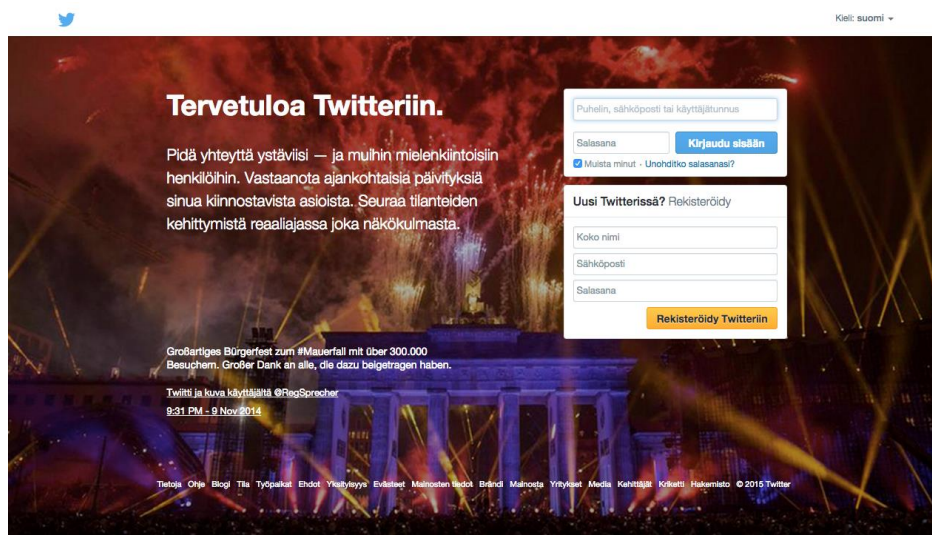
Uudessa C7-versiossa tavoitteena oli myös parantaa responsiivisuutta, jotta julkaisujärjestelmän käyttö onnistuisi ongelmitta ainakin tablet-laitteilla vaakatilassa. Tätä seikkaa silmällä pitäen julkaisujärjestelmästä haluttiin poistaa kaikki pop up-ikkunat, jotka ovat käytettävyydeltään huonoja ajatellen tabletin käytettävyyttä.

Käyttöliittymän ulkoasun uudistus tapahtui pitkälti päivittämällä kaikki mahdolliset käyttöliittymän osat Twitter Bootstrapin mukaisiksi. Käyttöliittymään päätettiin ottaa käyttöön myös teemoitus, jolloin järjestelmän värimaailmaa, fontteja ja muita tyyliseikkoja

voidaan muokata. Tämä nähtiin tarpeelliseksi, sillä yhdellä asiakkaalla käytössä voi olla useampi järjestelmä. Tällöin voi joskus aiheutua hämmennystä missä järjestelmässä ollaan kirjautuneena ja minne ollaan tekemässä muutoksia. Käyttöliittymän teemoitus ratkaisee ongelman esimerkiksi määrittämällä eri järjestelmiin erilaiset väriteemat.

4.2 Twitter Bootstrap

Twitter Bootstrap on sovelluskehys, joka helpottaa ja nopeuttaa toimivien käyttöliittymien rakentamista. Se kehitettiin alun perin Twitterin käyttöliittymän toteutukseen Twitterin työntekijöiden Mark Otton ja Jacob Thorntonin toimesta. Alun perin Thornton ja Otto kehittivät Bootstrapia yhtenäistämään Twitterin sisäisiä työkaluja, mutta nykyään se on täysimittainen sovelluskehys ja käyttöliittymäkirjasto (kuvio 7).



Kuvio 7. Twitter Bootstrap kehitettiin alunperin Twitter-mikroblogipalvelua varten.

Bootstrap on hyvin yleiskäyttöinen ja sen voi ottaa käyttöön jo käynnissä olevassa projektissa, kuten tässä Base C6:n käyttöliittymä uudistuksessa. Toinen vaihtoehto on käyttää Bootstrapia heti projektin alusta alkaen. Molemmat käyttöönotto vaihtoehdot onnistuvat aivan yhtä helposti. Bootstrap tarjoaa ison kirjaston valmiiksi tyyliteltyjä käyttöliittymäkomponentteja ja erilaisia rakennustyökaluja web-sivuston tai web-sovelluksen toteuttamiseen. Työkaluja ovat esimerkiksi responsiivinen grid-järjestelmä, lomakkeiden

elementit, ikonit ja navigointikomponentit. Twitterin käyttöliittymä on rakennettu kokonaan sen omalla Bootstrap-kirjastolla. Nykyään se on kaikkien saatavilla ilmaiseksi.

Käyttöliittymän uudistus aloitettiin lisäämällä esimääritettyjä elementtejä, joiden sisään voidaan sitten lisätä HTML-elementtejä. HTML-elementtejä kuten otsikoita tai painikkeita voitiin muuttaa Bootstrapin mukaiseksi lisäämällä niille Bootstrapissa määritettyjä luokkamääreitä (Korpela, J. 2012).

Twitter Bootstrap sisältää myös paljon JavaScript-toiminnallisuuksia. Bootstrapin virallisessa dokumentaatiossa näitä kutsutaan plugineiksi eli lisäosiksi. Näiden ominaisuuksien käyttämiseksi HTML-dokumenttiin pitää linkittää jQuery-kirjasto ja Bootstrapin oma JavaScript-tiedosto alla olevan ohjelmoinnin mukaisesti.

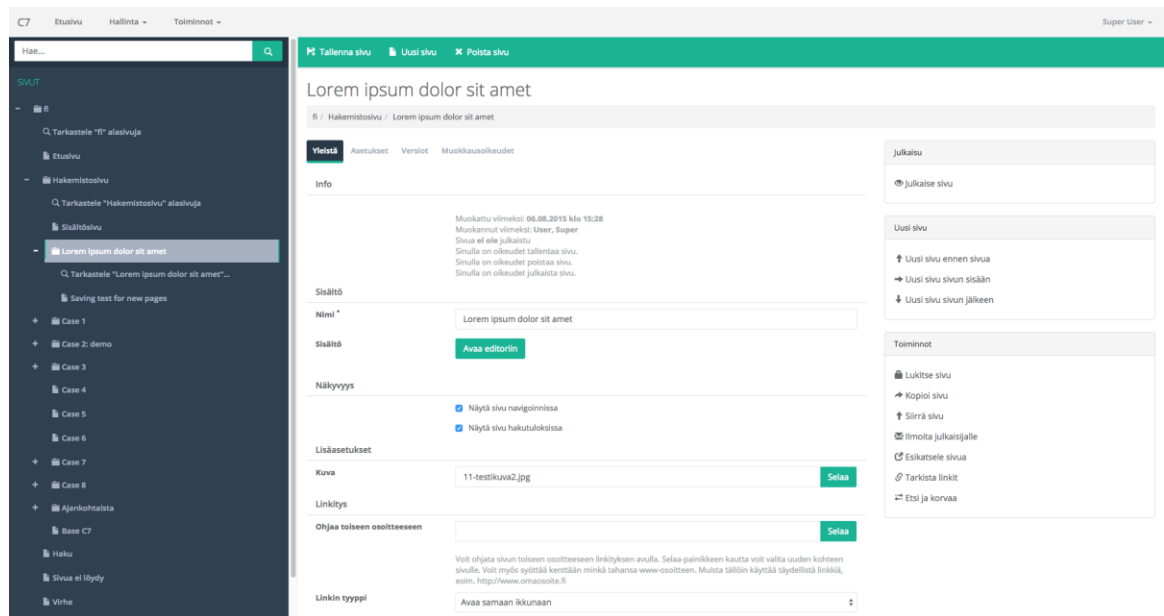
```
<link rel="stylesheet" href="/css/bootstrap.min.css">
<script src="js/jquery.min.js"></script>
<script src="/js/bootstrap.min.js"></script>
```

JavaScript-toiminnallisuuksia on useita mm. Bootstrapin välilehti-komponentti. Tätä käytettiin esimerkiksi Base C7-version pages-moduulin muokkausnäkyvässä. Sillä mahdollistetaan sisällön esittäminen välilehdissä. Esimerkiksi alla olevassa tapauksessa yleiset tiedot, asetukset, version hallinta ja oikeudet ovat omilla välilehdillään (Kuvio 8.). Tämä parantaa käytettävyyttä merkittävästi, koska suuri määrä sisältöä saadaan sijoitettua fiksumusti pieneen tilaan ja niiden välillä liikkuminen on vaivatonta. (Korpela, J. 2012).

Luokkamääryyksillä pystytään muuttamaan esimerkiksi painikkeiden ulkonäköä (esimerkki alla).

```
<button type="submit" name="Submit" value="Search" class="btn btn-
primary">
    <span class="glyphicon glyphicon-search text-center"
    style="vertical-align:middle; top: 0;"></span>
</button>
```

Tässä esimerkissä luotiin Bootstrapin luokilla hakupainike. Buttonille annettiin luokka ”btn”, joka määrittää peruspainiketyylit. Toinen luokka btn-primary määrittää kyseessä olevan ensisijaisen painikkeen, jolle tulee esimerkiksi teeman turkoosi väri. Painikkeen sisälle on määritetty span-elementti, joka sisältää keskitetyn glyphicon-hakuikonin. (Kuvio 8).



Kuvio 8. Tabs-komponentit ja hakukomponentti

4.3 Käyttöliittymäkomponentit

4.3.1 Yleistä

Base C6 -julkaisujärjestelmässä on suuri määrä käyttöliittymäkomponentteja, kuten hakukenttiä, painikkeita ja sisältökenttiä. Twitter Bootstrap tarjoaa käyttöliittymäkomponenttikirjaston, josta löytyy valmiiksi tyyllitellyt yhdenmukaiset komponentit erilaisiin käyttöliittymän tarpeisiin. Lisäksi Bootstrap-kirjasto sisältää glyphicon-ikonikirjaston, joka vähentää omien ikoneiden tarvetta ja mahdollistaa yhtenäisen ulkoasun muiden käyttöliittymäkomponenttien kanssa.

Iso osa opinnäytetyötä oli käyttöliittymäkomponenttien vaihtaminen Bootstrapin mukaisiksi. Seuraavassa esitellään eri käyttöliittymäosien muutokset ja vertailun C6- ja C7-versioiden välillä ja avataan käyttöliittymämuutoksia teoriankin puolesta.

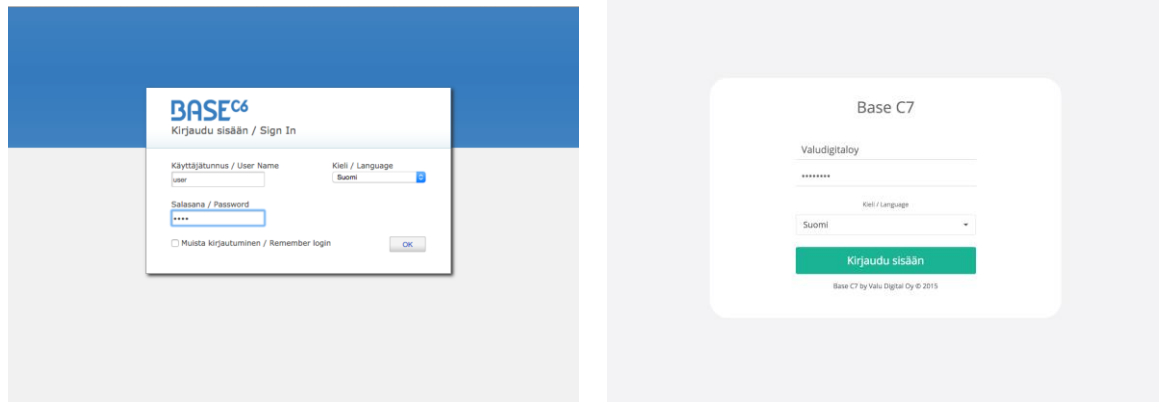
4.3.2 Kirjautumissivu

Base C6 -julkaisujärjestelmää käytettäessä ensimmäisenä tullaan kirjautumissivulle. Kirjautumissivulta käyttäjä luonnollisesti kirjautuu julkaisujärjestelmään.

Uudella kirjautumissivulla pyrittiin yksinkertaisuuteen. Päädyttiin poistamaan ”muista minut tällä tietokoneella” -checkbox, koska nykyiset selaimet osaavat hoitaa tämän asian itsenäisesti eikä enää ole tarvetta käyttää ”muista minut” -valintaa ja evästeitä. (Dan Singerman. 5.8.2010). Kuitenkin päätettiin pitää kielivalinta samanlaisena kuin ennen. Tähän päädyttiin, koska kehittäjien on tällöin helpompi vaihtaa kieliversioiden välillä, jos jokin tehtävä tai testaustarve sitä vaatisi.

Kirjautumissivulla oli ennen myös käytössä pop-up-ikkunoiden eston tunnistaminen. Tämä paransi vanhassa versiossa käytettävyyttä varmistettaessa, että käyttäjältä on poistettu käytöstä ko. esto. Näin aukeavat editori- tai muut vastaavat toimintoikkunat eivät jääneet käyttäjältä piiloon. Kuitenkin C7-versiossa ei nähty enää tarvetta pop up – tarkistukselle, koska uudessa versiossa käytetään modaali-ikkunoita pop up -ikkunoiden sijaan.

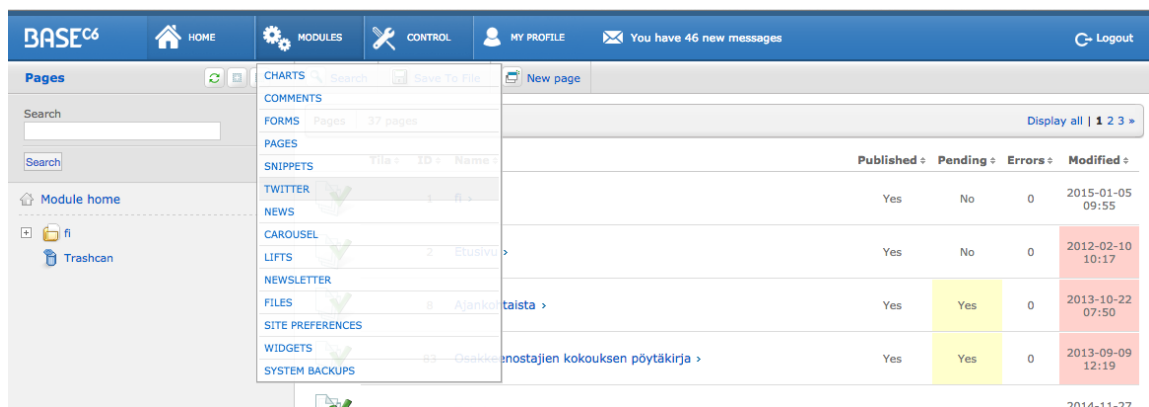
Lopuksi muutettiin kaikki täytettävät tekstikentät ja kirjautumispainike käyttämään Bootstrapin komponentteja antamalla niille tarvittavat luokat (Kuvio 9).

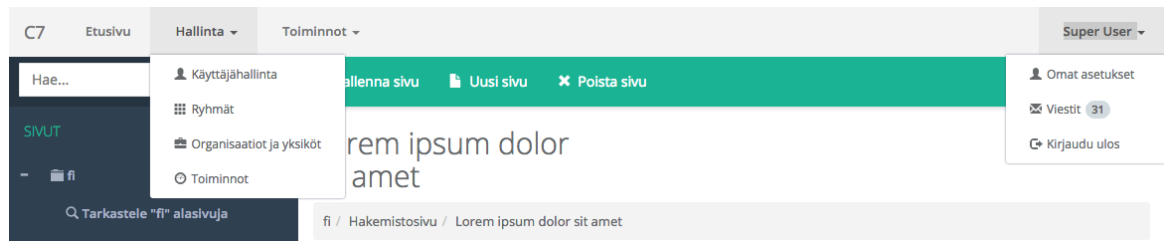


Kuvio 9. Base C6- ja C7-version kirjautumisnäkymät.

4.3.3 Navigaatio

Navigaatiota muutettiin uuteen versioon tiivistämällä elementtejä pudotusvalikoihin ja näin tekemällä navigaatiosta yksinkertaisempi. Uutena C7-versioon luotiin profiilivalikko. Profiilivalikot ovat nykyaikainen tapa tuoda kaikki käyttäjäkohtaiset valinnat ja asetukset helposti löydettäväksi samaan paikkaan. Samanlaista valikkoa käytetään esimerkiksi Twitterissä. Profiilivalikosta löytää oman käyttäjätilinsä asetukset, järjestelmän sisäiset viestit ja ulos kirjautumisen. Valikkopalkissa on aina näkyvissä käyttäjätilin nimi, joten käyttäjä on aina perillä siitä, millä tilillä hän on kirjautunut järjestelmään. Viestitoiminnossa hyödynnettiin Bootstrapin badge-toiminnallisuutta, jolla pystyttiin näyttämään uusien lukemattomien viestien lukumäärä suoraan pudotusvalikossa. Lisäksi valikoissa olevien valintojen havainnollistamiseen käytettiin glyphicon-ikoneita. Ne elävöittävät valikkoa ja lisäävät ymmärrystä mitä mistäkin toiminnosta tapahtuu. (Kuvio 10.)





Kuvio 10. Yllä C6:n navigaatio ja alla uusi C7-navigaatio.

4.3.4 Muokkausnäky

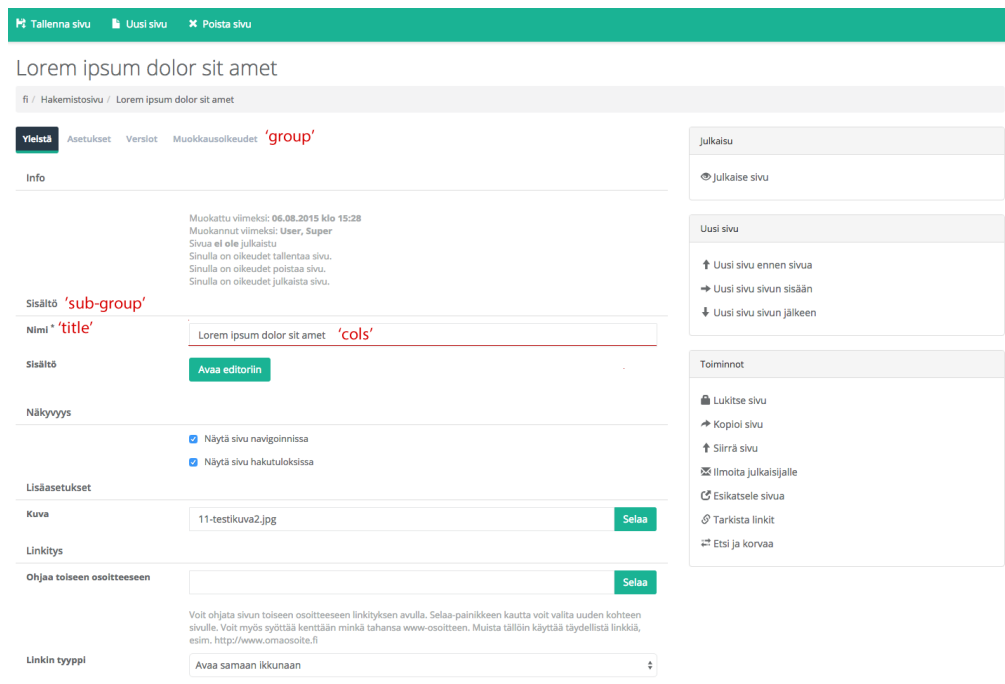
Muokkausnäky löytyy Base C6 -julkaisujärjestelmässä jokaisesta moduulista. Muokkausnäky luodaan moduulikohtaisesti edit.php-tiedostossa. Muokkausnäkyä käytetään esimerkiksi muokatessa tai luodessa uutta sivua, tapahtumaa tai artikkelia. Muokkausnäky rakentuu otsikosta, muropolusta ja lomakkeen kentistä. Muokkausnäkyt voivat olla hyvinkin erilaisia riippuen moduulista. Esimerkiksi tapahtumat-moduulista löytyy päivämäärä- ja aikavalitsimet, kun taas sivut-moduulista löytyy esimerkiksi linkitys mahdollisuus toiselle sivulle.

Muokkausnäkyssä on hyödynnetty moduulista riippuen erilaisia käyttöliittymäkomponentteja, jotka parhaiten soveltuvat käytettäväksi kyseisessä tarkoituksessa. Yleensä muokkausnäkyssä on moduulista riippumatta hyvin paljon eri kenttiä mihin voi syöttää tietoa. Tämän suuren tietomäärän järkevään esittämiseen tarvittiin tiedon jäsentelyä ja ryhmittelyä. Tässä hyödynnettiin käyttöliittymäkomponenttien ryhmittelyä alaryhmiin ja Bootstrapin välilehti-toiminnollisuutta. Muokkausnäkyssä käyttöliittymäkomponentit ovat suuren lomakkeen kenttiä, jotka luodaan edit.php-tiedostossa alla olevan ohjelmoinnin mukaisesti.

```
$tmp_formdata['group3'][] = Array(
    'subGroup' => $T->T('Content'),
    'title' => $T->T('Name'),
    'cols' => 9,
    'required' => false,
    'field' => "<input... .../>"
```

);

Sen avulla pystyttiin kätevästi näyttämään suuri määrä tietoa pienemmässä tilassa viisaasti jäsennehtynä. Group 3 määrittää välilehden mille kenttä sijoitettiin, subgroup alaosikon ja cols-attribuutti leveyden antamalla div-elementille Bootstrapin col -tyyliluokan. (Kuvio 11).



Kuvio 11. Muokkausnäytön rakenne.

4.3.5 Päänäkymä

Base C6 -julkaisujärjestelmässä jokaisella moduulilla on aina main- eli päänäkymä, joka luodaan moduulikohtaisessa main.php-tiedostossa. Kun julkaisujärjestelmässä siirrytään moduulista toiseen käyttäjä näkee päänäkymän ensimmäisenä. Päänäkymän vasemmalta puolella on puurakenteinen hakemisto esim. sivuista jos valittuna on sivut-moduuli. Päänäkymä koostuu muutamasta tärkeästä käyttöliittymäkomponentista ja listauksesta.

Muutoksina C7-versioon haku muutettiin yläpalkista sivutus-komponentin viereen ja siitä tehtiin suodatus-toiminto. Sen tyylittelyssä hyödynnettiin Bootstrapin dropdown-toiminnallisuutta. Lisäksi sivutus-komponentti vaihdettiin käyttämään Bootstrapin pagination-tyyliä. Listausta ei haluttu muuttaa sarakkeiden tai sisällön osalta. Vanhan version

listaus sisälsi kaiken tarvittavan tiedon mitä käyttäjä tarvitsee eikä sen käytettävyyttä nähty ongelmallisena. C7-versioon listauksen ulkoasua haluttiin kuitenkin päivittää. Ensimmäisen sarakkeen kuvat vaihdettiin glyphicon-ikoneiksi, jotka muuttuvat sisällön mukaan. Listauksessa lisättiin myös rivin korkeutta, jolla mahdollistettiin tiedostoarkistossa suuremman thumbnail-kuvan käyttäminen glyphicon-ikonin sijasta kuvatiedostojen kanssa. Tehostevärien käytöstä päätettiin myös luopua, koska se rikkoi väriteemaa ja ulkoasua. Lisäksi myös glyphicon-ikonit esimerkiksi sivut-moduulissa ilmasivat jo julkaistu tilan, joten kahta ilmaisinta ei nähty tarpeelliseksi (Kuvio 12).

Tila	ID	Nimi	Julkaistu	Odottaa	Virheitä	Muokattu
✓	1	fi	Kyllä	Ei	0	2015-01-05 09:55
✓	2	Etusivu	Kyllä	Ei	0	2012-02-10 10:17
✓	8	Ajankohtaista	Kyllä	Kyllä	0	2013-10-22 07:50
✓	83	Osakkeenostajien kokouksen pöytäkirja	Kyllä	Kyllä	0	2013-09-09 12:19
✓	10	Toiminnot	Kyllä	Ei	0	2014-11-27 14:16
✗	14	Sivuston asetukset	Ei	Ei	0	2012-02-10 14:54
✓	15	Tiedotteet	Kyllä	Ei	0	2012-02-10 14:54
✗	82	Suomi2	Ei	Ei	0	2013-09-06 16:35
✗	16	Etusivun karsuelli	Ei	Ei	0	2012-02-10 14:54
✗	17	Etusivun nebot	Ei	Ei	0	2012-02-10 14:54
✗	18	Tiedostot	Ei	Ei	0	2012-02-10 14:55

Kuvio 12. C6- ja C7-version päänäkymät vierekkäin.

4.4 Käyttöliittymäkirjastojen päivitys

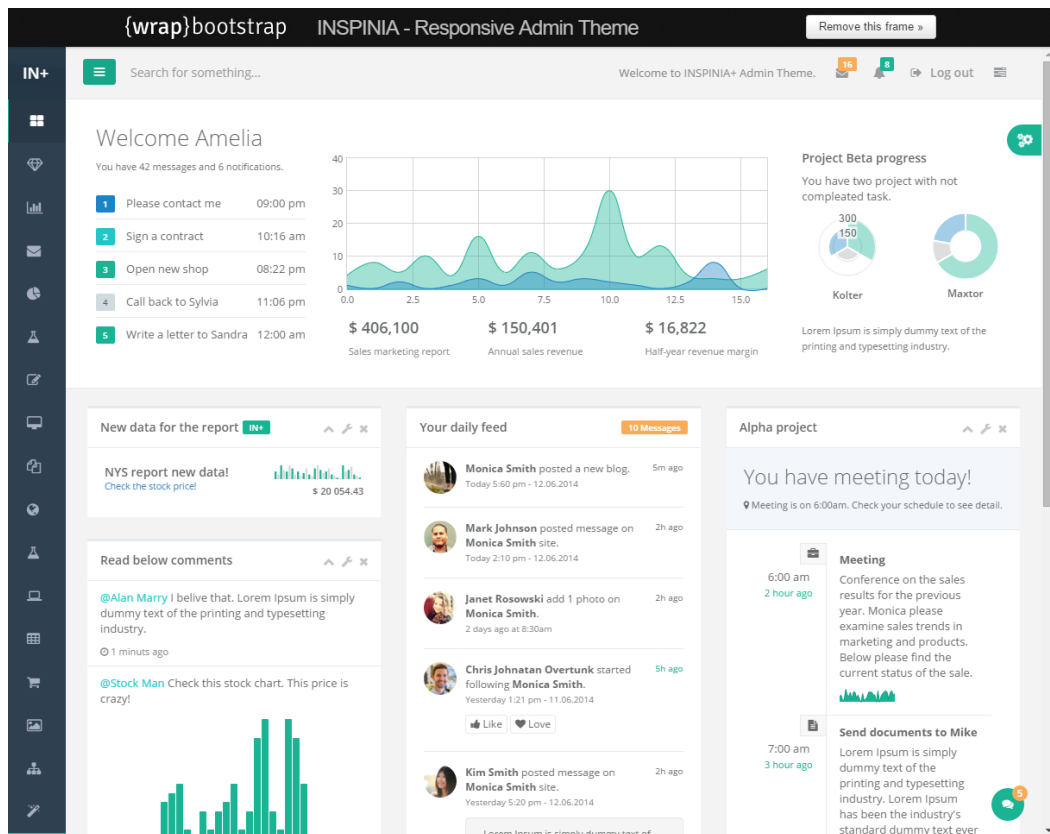
Base C6 -julkaisujärjestelmä sisältää erilaisia moduuleja, joista jotkin tarvitsivat toimintaansa jQuery-kirjastoja. Vanhassa Base C6 -järjestelmässä nämä tarvittavat jQuery-kirjastot olivat jokaisen moduulilla omansa. Tämä tarkoitti käytännössä sitä, että vanhasta julkaisujärjestelmästä löytyi useampi kappale samaa tiedostoa eri hakemistoista ja moduulit saattoivat käyttää eri versioita 1.4.3- ja 1.8.3-version välillä.

Base C7-julkaisujärjestelmää uudistettiin siten, että kaikkien tarvitsevien moduulien jQuery-kirjasto haettiin samasta paikasta ja poistettiin tarve erillisille moduulien sisäisille jQuery-kirjastoille. Tähän ratkaisuun päädyttiin, koska haluttiin vähentää paketin kokoa kokonaisuudessaan hieman ja tehdä jQuery-kirjastojen hakemisesta loogisempaa ja te-

hokkaampaa. Lisäksi tällöin jokainen moduuli käyttää samaa jQuery 1.11-versiota. Samalla tavalla haettiin jokaiseen modaaliin myös Bootstrapin omat tyyli- ja JavaScript-tiedostot.

4.5 Bootstrap ja teemat

Bootstrap on yksi maailman yleisimmistä sovelluskehysistä, joten on luonnollista, että siihen on saatavilla erilaisia lisäosia ja teemoja. Teemojen käyttö perustuu Bootstrapin standardoituihin luokkiin ja elementteihin. Teemat sisältävät yleensä omat tyyli-tiedostot, jotka sitten ylikirjoittavat Bootstrapin omat oletustyyli. Yrityksessä päädyttiin tähän ratkaisuun, koska huomattiin että markkinoilla on useita moderneja ja hyvännäköisiä Bootstrap admin-teemoja, joita voitaisiin hyödyntää osana projektia. Yrityksessä päädyttiin valitsemaan teemaksi Inspinia (Ks. kuvio 13).



Kuvio 13. Inspinia Bootstrap-teema.

Teemoituksen onnistumisen kannalta tärkeintä oli, että aiempi komponenttien, elementtien ja näkymien käännoistyö Bootstrapin mukaiseksi oli tehty huolella. Jos luokat eivät olleet kunnossa tai joku komponentti oli rakennettu Bootstrapin standardien vastaisesti, siitä seurasi ongelmia. Teemoitusta käyttöönottaessa tyylit eivät olleet täysin kunnossa, ja tyylejä jouduttiin muokkaamaan järjestelmään sopiviksi mm. muokkaamalla SCSS-muuttujat sisältävää SASS-tiedostoa. Esimerkiksi värit eivät menneet oikeille elementeille, tekstien värit olivat vääriä ja osa elementeistä oli osittain toistensa päällä. Teeman käyttöönotto vaati paljon työtä ja se osoitti missä komponenteissa ja osioissa oli kääntämisen ja luokkien osalta menty pieleen.

Valmiin teeman käyttöön päädyttiin yrityksessä ajan säästämiseksi ja siksi, että teemoja pystyi muokkaamaan omien tarpeidensa mukaisiksi. Yrityksessä ei nähty tarpeelliseksi käyttää resursseja tyylittelyyn paljoa, vaan pohjana päätettiin käyttää Inspinia-teemaa. Teeman mukana tulleita SCSS-tiedostoja pystyttiin käyttämään teemoituksen pohjana. Teeman tehtiin muutoksia, jotta teema sopi paremmin Base C7 -julkaisujärjestelmän kanssa käytettäväksi ja siitä saatiin tehtyä omanlaisensa.

5 TEKNILLINEN TOTEUTUS

5.1 Moduulien päivitys

Base C6 -julkaisujärjestelmässä käytetään paljon moduuleita. Moduulien tarkoituksena on mahdollistaa erilaiset toiminnallisuudet ja toiminnot, joita web-sivustoilla tarvitaan. Esimerkkeinä Base C6 -julkaisujärjestelmästä löytyy tapahtumat, jolla käyttäjä voi luoda, muokata ja hallita tapahtumia esimerkiksi toistumisen, ajankohdan ja keston osalta. Käytännön tasolla Base C6 -julkaisujärjestelmän moduulit vastaavat WordPress-julkaisujärjestelmän lisäosia.

C7-versiota varten moduuleita tuli päivittää, jotta ne saatiin toimimaan uuden PHP 5.6 -version kanssa. Kehitysversiossa ei ollut mukana aluksi kuin muutama tärkein moduuli, kuten sivut, artikkeli ja tiedostoarkisto-moduuli. Ennen päivittämistä asennettiin uudet

tarvittavat moduulit, joita olivat muun muassa kaaviot, lomakkeet, kontaktit ja tapahtumat. Moduulit asennettiin lataamalla versionhallinnasta kehitysversiota varten luodut 2.0a-versiot ja ajamalla niiden asennustyökalu, jolla rekisteröitiin moduulit käyttöön julkaisujärjestelmässä sekä luotiin tietokantoihin tarvittavat tietokantataulut.

Päivitettäviä moduuleja oli useita erilaisia ja suurin osa muutoksista tehtiin käsin eri tiedostoihin, joista moduulit rakentuvat. Pääosin moduulit päivitettiin muokkausnäkyssä ja päänäkyssä. Moduulit rakentuvat PHP-tiedostoista, joilla luodaan moduulin eri näkymien komponentit. Moduulin eri näkymiä muokattiin vastaavista PHP-tiedostoista. Esimerkiksi sivun muokkausnäkyä pystyttiin muokkaamaan pages-moduulin edit.php-tiedostolla. Moduulin päivittämiseksi käytiin moduulin edit.php- ja main.php-tiedostot läpi ja muutettiin elementtien rakennetta ja luokkia niin, että ne vastaavat Bootstrap-sovelluskehityksen standardeja.

Esimerkiksi artikkelit-moduulissa julkaisuajankohta-komponentti muodostettiin aiemmin, kuten alla.

```

$elements['publishTime'] = array (
    'subGroup' => $groupTitle,
    'title' => $T->T('Publish time'),
    'required' => false,
    'field' => '<input type="text" name="publishDate" size="10"
maxlength="10" value="'. $date. '" class="articles_date" /> '.
        '<input type="text" name="publishTime" size="5"
maxlength="5" value="'. $time. '" class="articles_time"
/> ',
    'help' => $T->T('help_publish_time')
);

```

Uudessa versiossa komponentti muutettiin käyttämään Bootstrap päivämäärävalitsinta ja lisäksi siihen lisättiin käytettävyyttä selventävät kalenteri- sekä kelloikonit span-elementeillä. Lisäksi käytettiin Bootstrap-kehityksen input-ryhmää, jolla saatiin asetettua painikkeet input-kenttien sisään.

```

$elements['publishTime'] = array (
    'subGroup' => $groupTitle,
    'title' => $T->T('Publish time'),
    'required' => false,
    'cols'      => 9,
    'field' => '<div class="row">'.
    '<div class="col-xs-6">'.
    '<div class="input-group" id="publishDate"><input
type="text" name="publishDate" size="10" maxlength="10"
value="'. $date. '" class="articles_date form-control" /> ' .
    '<label for="publishDate" class="input-group-addon
btn"><span class="glyphicon glyphicon-
calendar"></span></label></div>'.
    '</div>'.
    '<div class="col-xs-3">'.
    '<div class="input-group has-feedback has-input-feedback">'.
    '<input type="text" name="publishTime" id="publishTime"
size="5" maxlength="5" value="'. $time. '"
class="articles_time form-control" /> '.
    '<div class="input-feedback"><span class="glyphicon glyphi-
con-time form-control-feedback"></span></div>'.
    '</div>'.
    '</div>'.
    '</div>',
    'help'      => $T->T('help_publish_time')
);

```

Samalla tavalla käytiin läpi kaikki yksittäiset lomakekentät moduulikohtaisesti ja tehtiin niihin Bootstrapin mukaiset muutokset. Esimerkiksi perus input-kentät oli helppo kääntää vain cols-attribuutin leveysmäärittelyllä ja antamalla elementille form-control-tyyliluokka. Useammasta erilaisesta pienemmästä komponentista koostuvat elementit, kuten tiedoston selaus- tai monivalintaelementit, vaativat taas enemmän työtä kääntämisen osalta.

5.2 Käyttöliittymän toiminnallisuuden muutokset

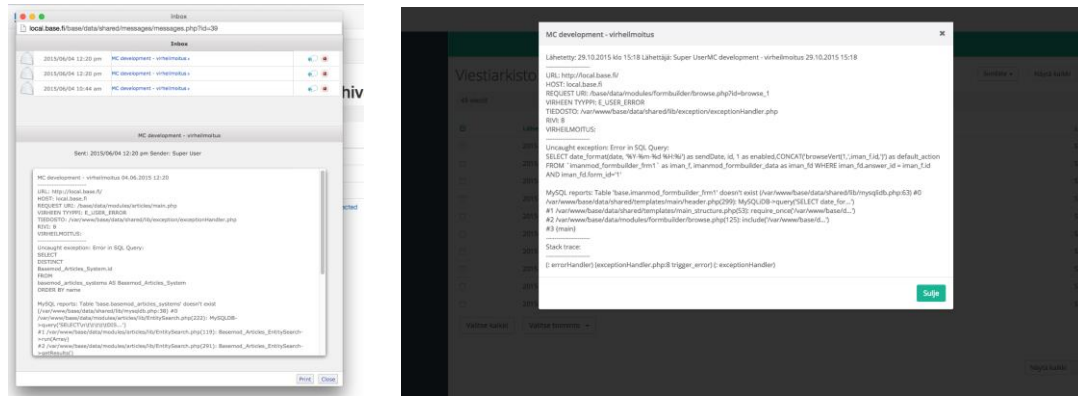
5.2.1 Modaali-ikkunat

Vanhassa Base C6 -julkaisujärjestelmässä useat toiminnallisuudet sisälsivät uuteen selainikkunaan aukeavia asetuksia ja valintoja, joilla muun muassa voitiin lukita sivu tai avattiin järjestelmän lähettämät virheviestit. Tällaiset popup-ikkunat eivät ole käytettävyydeltään optimaalisia. Yrityksessä nähtiin tarpeelliseksi luopua uusiin ikkunoihin avautuvista toiminnoista. Ongelmana nähtiin esimerkiksi tablettikäytettävyys, jossa popup-ikkunat eivät toimi. Ne ovat käytettävyydeltään kömpelöitä jos ne esimerkiksi jäävät piiloon pop-up-eston takia tai jäävät piiloon toisen ikkunan alle.

Bootstrap tarjoaa hyvän vaihtoehdon pop-up-ikkunoille modaali-ikkunoilla. Modaali on sisällöllinen ikkuna, joka on jo sivulla piilotettuna. Modaali tuodaan näkyviin esimerkiksi jotain käyttöliittymän painiketta painettaessa, jolloin modaali sisältö aukeaa sivun päälle ja tummentaa sivun taustan sen avaamatta uutta ikkunaa. Tällöin modaalin oma sisältö saadaan hyvin esille ja käyttäjän on helpompi havaita se ja suorittaa toimintoja sen kautta.

Kaikissa moduuleissa, missä hyödynnettiin modaali-ikkunoita, käytiin myös läpi modaalin alaosassa sijaitsevat painikkeet. Painikkeiden osalta hyödynnettiin tutkittua teoriaa ja huomattiin että vanhassa versiossa painikkeiden järjestys saattoi vaihdella, eli toisessa peruuta-painike saattoi olla vasemmalla. Toisissa modaaleissa taas hyväksymis-painike oli taas vasemmalla. Käytettävyyden osalta tärkeintä oli yhteneväisyys modaalien välillä. Tämänlaisia pieniä käytettävyyden asioita korjattiin uuteen C7-versioon modaali-ikkunoiden yhteydessä.

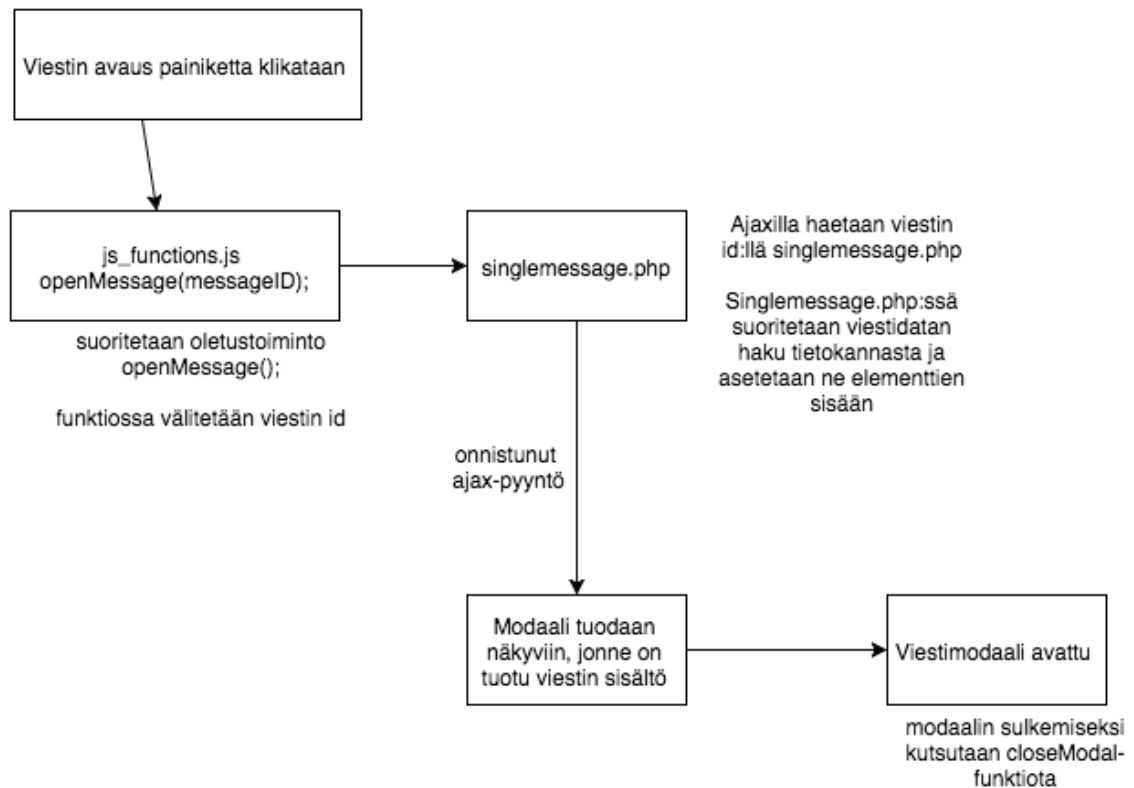
Viestiarkistossa hyödynnettiin modaali-ikkunoita ja parannettiin toiminnallisuutta lisäämällä viestiarkisto päänäkymään, kun taas Base C6-versiossa viestiarkisto ja avattu viesti olivat samassa pienessä popup-ikkunassa (kuviot 14).



Kuvio 14. C6- ja C7 version viesti-ikkuna ja viestimodaali-ikkuna

Popup-ikkunassa oli ylälaidassa viestien listaus ja alla luettavissa valittu viesti. Ikkuna oli kooltaan myös hyvin pieni, joten esimerkiksi järjestelmän lähettämät virheviestit eivät sopineet kokonaan esille. Uuteen versioon muutettiin toimintaa siten, että yläkulmasta viestejä klikatessa aukeaa viestiarkiston oma sivu. Yrityksessä päädyttiin tähän ratkaisuun, koska silloin on helpompi tarkastella viestejä, kun ne ovat selvästi taulukkonäkyssä.

Viestiarkistossa käyttäjälle tarjottiin mahdollisuus valita viesti minkä hän haluaa lukea. Viesti aukeaa tällöin erilliseen modaali-ikkunaan uuden ikkunan sijasta. Viestimodaalin sisältö haettiin Ajax-tekniikalla viestin id:n perusteella. Ajax-pyyntö onnistuessa viestin sisältö haettiin modaalin sisältöelementtiin ja modaali tuotiin näkyviin. (kuvio 15).



Kuvio 15. Viestimodaali-ikkunan toiminta.

Muiden moduulien toiminnoissa, kuten Sivut-moduulin sivun lukituksessa tai kuvan selaamisessa ei tuotu Ajax-pyyntöllä dataa HTML-sisältönä, vaan PHP-tiedoston sisältö haettiin modaali-ikkunan sisällä olevaan iframe-komponenttiin. PHP-tiedosto, joka haettiin iframen src-attribuuttiin, koostui yhdestä tai useammasta iframe-komponentista. Basen äitikehykseen luotiin modaalille oma div-elementit Bootstrapin modaalin mukaan ja sen sisään luotiin iframe-komponentti. Php-tiedostolla haettiin haluttu sisältö kehyksen sisään. Sisältö saattoi esimerkiksi olla tiedostoarkisto tiedoston lataus-näkymä tai sivupuu, josta pystyttiin valitsemaan sivun uudelleenohjaus.

Modaalille luotiin funktio, jolla on kolme attribuuttia, joita se ottaa vastaan: iframen lähde, iframen korkeus ja modaalin otsikko. Modaalin otsikko lokalisoitiin JavaScriptin avulla, joten sen sisältö vaihtuu käytettävän kielen mukaan. Artikkelimoduulissa se tapahtui määrittämällä käännettävät tekstit taulukkomuuttujaan EditArticleSpecs. Alla olevassa esimerkissä on määritetty sivun linkitys modaali-ikkunan otsikoksi haetaan "Link address", jonka suomenkielinen käänös määriteltiin käänöstiedostoon.

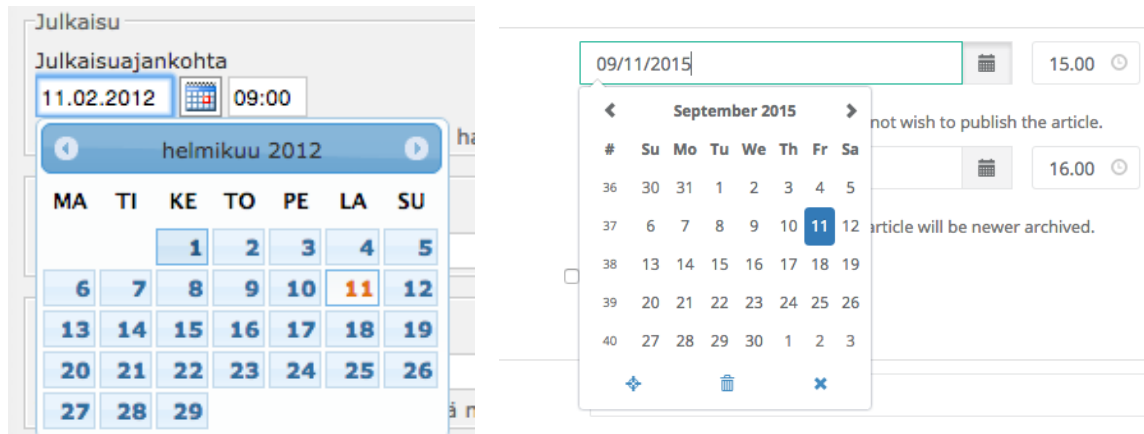
```

Var EditArticleSpecs = {
  TR : {
    browseLinkTitle : '<?=$T->JS('Link address') ?>'
  };
  $TR = array(
    'Link address' => 'Linkitä artikkeli toiseen sivuun'
  );
}

```

5.2.2 Päivämäärävalitsinkomponentit

Base C6 -julkaisujärjestelmässä käytetään moduuleita, jotka sisältävät aika- ja päivämäärätietoja. Päivämäärä- ja aikatiedolla voitiin määrittää esimerkiksi artikkelin julkaisuajan, tapahtuman alku- ja loppumisajan sekä sivun viimeisimmän muokkauksen päiväys ja aika. Aiemmassa versiossa käytettiin päivämäärien valitsemisessa jQuery-kirjaston datepicker-komponenttia (kuvio 16).



Kuvio 16. Vanha jQuery-datepicker ja uusi Bootstrap-datetimepicker-komponentti.

Uudessa C7-versiossa vaihdettiin jQuery:n päivämäärävalitsimet Bootstrap 3:n datetimepickereihin. Tähän päädyttiin, koska komponenttien ulkoasu haluttiin pitää yhtenäisenä ja Bootstrapin mukaisena koko järjestelmässä. Lisäksi esimerkiksi Bootstrapin teemojen käyttö onnistuu paljon helpommin, jos kaikki komponentit käyttävät Bootstrapin luokkia ja komponentteja. Bootstrap-sovelluskehys ei sisällä omia päivämäärävalinta-komponentteja, vaan jQuery:n komponentit korvattiin Bootstrapin datetimepicker-lisäosalla, joka on rakennettu käyttäen Bootstrapin tyylejä ja luokkia, joten sen valinta

käyttöön nähtiin hyväksi yrityksessä jo edellä mainitun teemoituksenkin kannalta.(Bootstrap 3 Datetimepicker v4 Docs, N.d.)

Päivämäärä- ja aikakomponentit rakennettiin moduulin edit.php-tiedostossa määrittämällä HTML-elementit ja niiden luokat. Field-attribuuttiin kirjoitettiin HTML-sisältöä, jolla luotiin div-elementit ja tarvittavat luokat. Lisäksi määritettiin komponenttiin id, jolla määritettiin missä päivämääräkomponenttia haluttiin käyttää. Moduulin JavaScript-tiedostossa kutsuttiin id:n perusteella funktiota, jolla komponentti otettiin käyttöön.

```
'field' => '<div class="row">'.
'<div class="col-xs-9">'.
'<div class="input-group" id="publishDate"><input type="text"
name="publishDate" size="10" maxlength="10" value="'.$.date.'"
class="articles_date form-control" /> ' .
'<span class="input-group-addon btn"><span class="glyphicon glyph-
icon-calendar"></span></span>
</div>'.
'</div>'.
'</div>'
```

Bootstrapin päivämääräkomponentille pystyttiin määrittämään useita eri asetuksia, joiden avulla pystyttiin vaikuttamaan painikkeisiin, päivämäärämuotoon ja kalenteriviikkojen esittämiseen. Yrityksessä päädyttiin ratkaisuun, jossa päivämäärä ja kellon aika ovat omissa kentissään, koska aiemmassa järjestelmässä ne olivat myös aseteltu samalla tavalla. Näin koodia ei tarvinnut muuttaa niin radikaalisti. Komponentti olisi tarjonnut hyvät mahdollisuudet lisätä päivämäärän ja kellonajan samaan elementtiin, mutta nähtiin selkeämmäksi pitää ne erillään päivämäärävalitsimen toimiessa paremmin, kun päivämäärälle ja ajalle oli määritetty omat kentät. Lisäksi aiempaan versioon eroten kellonajan valinnassa otettiin käyttöön myös ajanvalintakomponentti, normaalin kirjoittamisen sijaan. Muuttuja määritettiin alla olevan ohjelmoinnin mukaisesti.

```

var dateTimeOptions = {
    format: '<?=$T->JS('cal_date_format') ?>',
    locale: '<?=$T->JS('date_locale') ?>',
    calendarWeeks: true, //show calendar weeks in calendar
    showTodayButton: true, //button to go today
    showClear: true, //button to reset
    showClose: true } //button to close

```

Asetuksien avulla voitiin myös kätevästi määrittää päivämäärän formaatti ja komponentin kieli käytössä olevan kielen perusteella.

5.3 CSS-preprosessointi Compassilla ja SASS:lla

CSS-preprosessoinilla tai esikäännöllä tarkoitetaan työkaluja, joilla tyylitiedostojen kirjoittamista helpotetaan ja tehdään tehokkaammaksi. CSS-preprosessointityökaluja on olemassa useampia, mutta yrityksessä on aiemmin päädytty käyttämään SASS-preprosessointia.

SASS tarjoaa monipuoliset mahdollisuudet helpottaa CSS-tyyliä kirjoittamista. Kielellisesti se ei eroa kovinkaan paljon normaaliin CSS:n verrattuna. SASSilla on mahdollista hyödyntää tyylien kirjoituksessa esim. muuttujia ja mixeneitä. Näiden avulla on helppo esimerkiksi vaihtaa muuttujan kautta väri koko sivustoon sen sijaan, että käytäisiin normaali CSS-tiedosto läpi ja vaihdettaisiin värikoodi moneen eri paikkaan. SASSin suurin hyöty tulee esille CSS-määritteiden ”nestaamisessa”. Sen avulla tyyliä ei tarvitse toistaa alusta alkaen, vaan valitsimia voidaan kirjoittaa sisäkkäin kuten alla.

```

//CSS-esimerkki
.myDiv {
    width: 200px;
}
.myDiv h1 {

```

```

        font-size: 24px;
    }
    .myDiv h1 span {
        color: green;
    }

    //sama SASS:lla
    .myDiv {
        width: 200px;
        h1 {
            font-size: 24px;
            span {
                color: green;
            }
        }
    }
}

```

CSS-preprosessoinissa SASSilla tuotetaan SCSS-päätteisiä tyylitiedostoja, joita selaimet eivät tue. Tähän työvaiheeseen tarvitaan Compassia. Se on työkalu, jonka avulla käyttäjä voi kääntää SCSS-tiedostot selainten ymmärtämään normaaliin CSS-muotoon. Lisäksi Compass helpottaa SCSS-tiedostojen kirjoittamista sisältämällä kirjaston, jossa on ennalta määritettyjä tyyliä helpottamaan esimerkiksi selainkohtaisten prefix-määritysten kirjoittamista. (Loons, C. 2012).

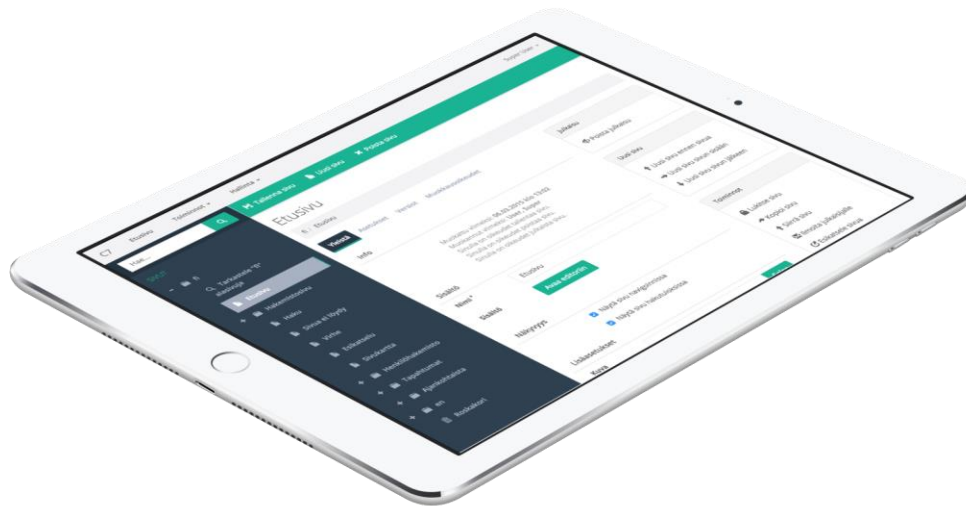
Base C7:ssä päätettiin ottaa käyttöön hyvin toimivaksi todettu SASS-preprossessori, jota on aiemmin käytetty yrityksen verkkoprojekteissa. Tämä nähtiin tarpeelliseksi esimerkiksi Bootstrapin teemoitusta ajatellen. Teeman väritystä on helppo muokata vain yhdellä variables.scss-tiedostolla, joka sisältää kaikki Basen käyttämät SCSS-muuttujat. Näillä muuttujilla määritetään julkaisujärjestelmän värikoodit ja sitä kautta voitiin helposti muokata myös teemoituksen ulkoasua. Myös esimerkiksi fontteja on helppo hallita samalla tavalla.

5.4 Responsiivisuus

Kun yrityksessä pohdittiin asioita, mitkä eivät vanhassa julkaisujärjestelmäversiossa toimi, nousi esiin monia asioita. Suurimpana ongelmakohtana nähtiin se, ettei julkaisujär-

jestelmän käyttö ole jouhevaa tai sujuvaa perustoimintojenkaan osalta mobiililaitteilla. Käyttäjän oli vaikeaa julkaista esimerkiksi pelkästään artikkeleita tai sivuja. Suurin syy tähän oli se, miten vanha järjestelmä oli suunniteltu toimimaan. Esimerkiksi kuvaa lisätessä tai sivua julkaistaessa, vanha järjestelmä avasi uuteen ikkunaan sisältöä, joka ei toimi esimerkiksi tabletilla. Tabletilla uudet ikkunat aukeavat uusiin välilehtiin ja peruskäyttäjälle sivujen minkäänlainen päivittäminen ilman tietokonetta oli hyvin vaikeaa.

Ongelma päädyttiin ratkaisemaan Bootstrapin modaali-ikkunoilla, joilla ikkunoihin aukeava sisältö pystyttiin näyttämään suoraan sivun päällä, ilman uusia selainikkunoita tai välilehtiä. Tällöin käyttö on jouhevampaa tietokoneella ja varsinkin tabletilla. Lisäksi myös kaikki edit- ja main-näkymät rakennettiin läpikotaisin käyttäen Bootstrapista löytyvää grid-järjestelmää. Kaikki käyttöliittymän komponentit ovat div-elementtien sisässä, joille annettiin tarpeen mukaiset luokkamääritykset. Näin julkaisujärjestelmän näkymät saatiin skaalautumaan myös vaaka orientoituun tablettiin. (Ks. kuvio 17).



Kuvio 17. C7-versio iPadilla.

Yrityksessä ei nähty tarvetta viedä julkaisujärjestelmän responsiivisuutta pidemmälle. Julkaisujärjestelmässä on esillä niin paljon dataa, komponentteja, kuvia, ikoneita ja muita elementtejä, joten sen mahduttaminen tablettia pienemmälle laitteelle olisi vaikeaa. Käyttökokemuksesta tuskin saataisiin kovinkaan hyvää ja sujuvaa. Jatkokehityksessä res-

ponsiivisuutta joudutaan vielä hiomaan vielä jonkin verran, jotta responsiivinen käyttökokemus pääsisi lähelle tyydyttävää tai lähes samalle tasolle tietokoneen käyttökokemusta. Opinnäytetyön puitteissa responsiivisuuden osalta pyrittiin tekemään perustoinnallisuudeltaan toimiva ja käytettävä. Tällä tarkoitetaan käytännön tasolla sitä, että käyttäjän on mahdollista kirjoittaa sivuille uutisia, luoda sivuille sisältöä ja kirjoittaa uusia blogikirjoituksia tai verkkolehden artikkeleita.

6 TULOKSET

Yleisesti ottaen ja lopputulosta tarkasteltaessa voitiin todeta opinnäytetyön päässeen suurimmalta osin tavoitteisiin. Vanha C6-versio onnistuttiin muuttamaan Bootstrap-sovelluskehiksen mukaiseksi. Käyttöliittymä uudistuksen tavoitteena ollut parempi ulkoasu ja parannetut toiminnallisuudet saavutettiin. Käyttöliittymä on ulkoasultaan parempi ja käytettävyydeltäkin on menty hieman eteenpäin. Teemoituksen osalta ei päästy aivan tavoitteisiin asti, mutta jatkokehityksessä tullaan keskittymään teemoituksen loppuun saattamiseen ja dokumentoimiseen.

Uuteen C7-versioon jäi vielä jatkokehitystoimenpiteitä vaativia seikkoja, kuten editorin muokkaaminen Bootstrapin mukaiseksi. C6-version käyttämä TinyMCE -editori on käytössä myös uudessa C7-versiossa, mutta jatkokehityksessä editorin modaalissa toiminen ja TinyMCE editorin tyyllittely Bootstrapin mukaiseksi tai sen vaihtaminen toiseen vaihtoehtoon ovat etusijalla.

Opinnäytetyössä saavutettuun lopputulokseen oltiin tilaajan puolesta tyytyväisiä. Erityisen tyytyväisiä oltiin modaali-ikkunoiden toimintaan ja ulkoasuun. Opinnäytetyön puitteissa tuotettu Base C7 -versio on melkein julkaisuvalmis Base-julkaisujärjestelmän versio. C7-version suurimmat jatkokehitystoimenpiteet liittyvät testaukseen, modaalieditoriin, teemoitukseen ja yleiseen viimeistelyyn. Teoriatutkimuksen pohjalta tehtyjä havain- toja ja oppeja hyödynnettiin käyttöliittymän muutoksissa mitä tehtiin C7-versioon. Saa- dun palautteen perusteella muutoksiin oltiin tyytyväisiä ja ne olivat hyvin käyttökelpoi-

sia. Teorian ja tutkimuksen luotettavuutta tai käyttökelpoisuutta ei tarvinnut kyseenalaistaa.

Opinnäytetyön työn lopputuloksena oli lähes julkaisuvalmis C7-versio julkaisujärjestelmästä. Järjestelmä tarjoaa asiakkaille paljon entistä nykyaikaisemman käyttöliittymän ulkoasun. Ulkoasussa hyödynnettiin erillistä Bootstrap 3:n admin-teemaa, jonka käyttöönotto onnistui hyvin, kun käyttöliittymän komponentit, näkymät ja toiminnallisuudet oli huolellisesti käännetty Bootstrap standardien mukaisiksi. Ulkoasusta tehtiin tyyliään modernimpi ja se näyttää paljon uudemmalta verrattuna C6-versioon, joka näytti selvästi vanhanaikaiselta ja päivittämättömältä.

Opinnäytetyöstä on hyötyä yritykselle. Base C7-versio tullaan päivittämään muutamalle asiakkaalle testauksen jälkeen käyttöön heti alkuvuonna 2016. Yritys on ollut tyytyväinen uuden version muutokseen ulkoasun, käyttöliittymän ja toiminnallisuuksien osalta. Yritys on halunnut uudistaa Base C6 -julkaisujärjestelmänsä jo pidemmän aikaa, mutta yrityksessä ei ole ollut oikein aikaresursseja tehdä asialle mitään.

Yrityksessä suurimpina hyötyinä uudessa versiossa nähtiin parantunut tabletti-käyttö, yhtenäisyys ja uudistunut ulkoasu. Yrityksen asiakkaiden on nyt helpompi käyttää järjestelmää esimerkiksi iPadilla ja myös tietokoneella. Yritys pystyy nyt kilpailemaan paremmin kilpailijoitaan vastaan uudistetulla ulkoasulla, paremmalla toiminnallisuudella ja eteenpäin menneellä responsiivisuudella.

7 POHDINTA

Käyttöliittymäkomponenttien kääntämisessä oli tavoitteena muuttaa kaikki julkaisujärjestelmän käyttöliittymässä käytetyt komponentit Bootstrap 3 -sovelluskehiksen mukaisiksi. Tämä tapahtui muokkaamalla vanhojen komponenttien HTML-rakennetta, jos käytettiin esimerkiksi Bootstrapin syöttö- tai painikeryhmiä. Elementille piti määritellä Bootstrapin responsiivisen grid-järjestelmän mukaiset leveydet ja muut tyyli CSS-luokkien avulla. Myös eri moduulien näkymiä muutettiin Bootstrapin mukaisiksi. Esimer-

kiksi sivut-moduulissa listauksia muutettiin tyylien, järjestyksen ja ikonien osalta. Kaikki vanhat kuvat olivat järjestelmässä GIF-muotoisina, mutta uudessa versiossa oli tavoitteena päästä kokonaan näistä eroon ja hyödyntää glyphicon-vektorigrafiikkaikoneita.

Käyttöliittymäkomponenttien kääntämisessä onnistuttiin hyvin. Tietyt moduulit aiheuttivat enemmän vaikeuksia luokkien lisäämisessä ja monimutkaisemmalla HTML-rakenteella tai sillä, että myös backend-koodia jouduttiin muuttamaan. Esimerkiksi profiilivalikon kanssa jouduttiin kirjoittamaan valikon koodia uudelleen, kun oli tarve siirtää kirjaa ulos -menu, luoda viesteille Bootstrapin badge ja hakea käyttäjänimi valikon otsikoksi. Osa moduuleista, joissa paljon perussisällönsyöttökenttiä oli taas helppo muuntaa Bootstrapin mukaisiksi.

Opinnäytetyön puitteissa lähes kaikki komponentit saatiin käännettyä onnistuneesti. Jatkokehityksessä kaikki luokat tullaan käymään läpi ja testataan, että ne toimivat hyvin yhteen teemoituksen kanssa. Julkaisujärjestelmän edellisessä C6-versiossa, oli useissa järjestelmän toiminnallisuuksissa muutettavaa. Esimerkiksi päivämääräkomponenteissa oli käytössä useampia erilaisia komponentteja: jQuery UI:n päivämäärävalitsimesta pelkään tekstikenttään.

Tavoitteena oli opinnäytetyön osalta pyrkiä yhtenäistämään komponentit, joita käytettiin useassa eri paikassa julkaisujärjestelmässä. Lisäksi toiminnollisuuksista haluttiin helpommin käytettävämpiä. Esimerkiksi pop-up-ikkunat olivat tavoitteena vaihtaa Bootstrapin modaali-ikkunoihin ja aikakentässä otettiin käyttöön aikavalitsin, joka teki komponentin käytöstä loogisempaa ja helpompaa.

Toimintojen muuntamisessa törmättiin muutamiin ongelmiin. Esimerkiksi tapahtumat moduulissa käyttöön valittu Bootstrap Datetimestpicker -lisäosa aiheutti konfliktin ennen käytetyn jQuery UI:n päivämäärävalitsimen kanssa, ja sama ongelma toistui myös moduuleissa, joissa oli käytössä jQuery UI:n autocomplete-täydennys. Päivämääräformaattien kanssa oli myös ongelmia, mutta lopulta onnistuttiin määrittämään yksilölliset päivämääräformaatit eri kielille. Päivämääräkomponenttien yhtenäistämisessä onnistuttiin

lopulta hyvin. C7-versiossa kaikki päivämääräkomponentit ovat samanlaisia eli niiden käytettävyys ja käyttöliittymä ovat samanlaisia riippumatta moduulista.

Modaali-ikkunoiden käyttöönotto oli yksi näkyvimpiä muutoksia uudessa versiossa. Kaikki vanhat, uuteen ikkunaan auenneet toiminnot muutettiin aukeamaan modaali-ikkunoihin. Modaali-ikkunat toimivat kätevämmiin kuin uudet selainikkunat, etenkin tablet-laitteilla kuten iPadilla. Eri toimintojen aukeaminen modaaleihin ja tietojen tallentaminen tuotti hieman vaikeuksia.

Basen käyttöliittymä koostuu useasta eri iframe-elementistä ja funktioiden kutsuminen ja parametrien välitys iframesta toiseen oli joskus vaikeaa, varsinkin kun funktiot kutsuivat toisia funktioita eri tiedostoissa. Sen takia funktion seuraaminen alusta loppuun oli työlästä, jotta siihen pystyttiin tekemään muutoksia. Modaali-ikkunoiden käyttöönotossa onnistuttiin, mutta jatkokehityksessä modaali-ikkunoiden tyylejä tullaan hiomaan ja parantelemaan. Esimerkiksi uutiskirje- ja lomake-moduulien modaalien tyylejä tullaan parantelemaan jatkokehityksessä.

Facebook- ja Twitter-moduuleissa törmättiin myös ongelmaan. Facebook tai Twitter-tilin lisäystä ei voitu avata modaali-ikkunaan. Modaalin sisältö avattiin viestit-toimintoa lukuun ottamatta iframeen. Ongelmana oli siis se, ettei Facebook tai Twitter salli sivujensa avaamista iframeen. Nämä toiminnot oli pakko siis jättää vielä aukeamaan uuteen selainikkunaan.

Teemoitus otettiin tavoitteeksi uuteen versioon varsinkin asiakkailta saadun palautteen perusteella. Asiakkaalla saattoi olla useampi järjestelmä, joka käyttää Base C6 -julkaisujärjestelmää ja tällöin ulkoasun ollessa kaikissa sama, saattoi sattua sekaannuksia, minne järjestelmään muutoksia tehtiin. Teemoituksen tavoitteena oli se, että asiakas voisi itse valita ja vaihtaa muutamasta valmiiksi luodusta väriteemasta järjestelmään mieleisensä vaihtoehdon.

Teemoituksen kanssa ei päästy tavoitteisiin opinnäytetyössä. Asiakas ei ainakaan vielä itse voi vaihtaa julkaisujärjestelmän teemoitusta. Julkaisujärjestelmä on mahdollista nyt-

kin teemoittaa muuttamalla variables.scss-tiedostossa olevia muuttujia, mutta itse asiakas ei pysty muuttamaan teemaa. Teemoituksen kanssa ei edetty niin pitkälle kuin opinäytenäytetyössä oli tavoite, lähinnä muiden korjauksia vaativien asioiden takia.

Käytännössä valmiudet teemoitukselle ovat olemassa. Jatkokehityksessä teemoitus tullaan viemään loppuun, ja sen muokkaamisesta tehdään asiakkaalle mahdollista. Lisäksi teemoitus tullaan dokumentoimaan myöhemmin jatkokehityksessä, kun sen osalta myöskin jäätin tavoitteesta.

LÄHTEET

- Bootstrap 3 Datetimepicker v4 Docs. N.d. Bootstrap3 Datetimepicker-verkkosivusto. Viitattu 1.9.2015. <http://eonasdan.github.io/bootstrap-datetimepicker/>
- Collins-Sussman, B. Fitzpatrick, B. & Pilato, C.M. What is SVN?. Viitattu 5.9.2015. <http://svnbook.red-bean.com/en/1.7/svn.intro.whatis.html>
- Download Bootstrap. N.d. Bootstrapin verkkosivut. Viitattu 2.4.2015. <http://getbootstrap.com/>
- Getting started. 2015. Vagrantin verkkosivut. Viitattu 10.4.2015. <http://docs.vagrantup.com/v2/why-vagrant/index.html>
- Giselle. 2014. What is Sass? Syntactically Awesome Stylesheets Explained. Viitattu 2.4.2015. <https://www.siliconbeachtraining.co.uk/blog/what-is-sass>
- Harvey, A. 2013. User experience: What Is It And Why Should I Care? Viitattu 3.9.2015. <http://usabilitygeek.com/user-experience/>
- Hyysalo, S. Oulasvirta, A. Väänänen-Vainio-Mattila, K. 2011. Ihmisen ja tietokoneen vuorovaikutus. Helsinki: Gaudeamus.
- Korpela, J. 2012. HTML5-kirja: Twitter Bootstrap. Viitattu 30.04.2015. <http://html5kirja.fi/2012/08/22/twitter-bootstrap/>
- Kurkkia, T. 2014. Uiah.fi-verkkosivusto. Vuorovaikutteisen tuotteen käytettävyys. Viitattu 12.9.2015. <http://www2.uiah.fi/projekti/metodi/058.htm>
- Käytettävyys. N.d. Wikipedia-verkkosivusto. Viitattu 15.5.2015. <http://fi.wikipedia.org/wiki/K%C3%A4ytett%C3%A4vyys>
- Leiniö, T. 2013. WP-opas-verkkosivusto. Modaali-ikkuna. Viitattu 4.6.2015. <https://wpopas.fi/avainsana/modaali-ikkuna/>
- Loons, C. 2012. Urban Insight-verkkosivusto. Ten reasons you should be using a css-preprocessor. Viitattu 11.10.2015. <https://www.urbaninsight.com/2012/04/12/ten-reasons-you-should-be-using-css-preprocessor>
- Miksi Base C6? N.d. Base C6:n verkkosivut. Viitattu 14.5.2015. <http://www.basec6.fi/miksi-base-c6>
- Nielsen, J. N,d. Norman Nielsen Group –verkkosivut. The Definition of User Experience. Viitattu 27.11.2015. <http://www.nngroup.com/articles/definition-user-experience/>

Ominaisuudet. N.d. Base C6:n verkkosivut. Viitattu 14.5.2015.

<http://www.basec6.fi/ominaisuudet>

Roto, V., Law, E., Vermeeren, A., Hoonhout, J. 2011. User experience white paper: Bringing clarity to the concept of user experience. All about UX Pdf-julkaisu. Viitattu 20.11.2015.

<http://www.allaboutux.org/files/UX-WhitePaper.pdf>

Singerman, D. 2010. Detecting a browser's language in Javascript. Viitattu 30.4.2015.

<http://blog.dansingerman.com/post/909213798/detecting-a-browsers-language-in-javascript>

Teknologiat. N.d. Media Cabinetin verkkosivut. Viitattu 2.4.2015.

<http://www.mediacabinet.fi/teknologiat>

Tolvanen, P. 2010. Esittelyssä: Base C6-julkaisujärjestelmä. Viitattu 30.04.2015.

<http://vierityspalkki.fi/2010/09/30/base-c6-julkaisujarjestelma/>

User Interface Design Basics. N.d. Usability.gov-verkkoasivusto. Viitattu 24.7.2015.

<http://www.usability.gov/what-and-why/user-interface-design.html>

Virta, P. 2014. Lumous Digital-yrityksen verkkosivut. Käyttäjäkokemus – mitä se on? Viitattu 10.9.2015.

<http://lumousdigital.fi/kayttajakokemus-mita-se-on/?lang=en>

What is usability?. N.d. Usabilitynet-verkkosivusto. Viitattu 1.9.2015.

http://www.usabilitynet.org/management/b_what.htm

Yritys. N.d. Media Cabinetin verkkosivut. Viitattu 2.4.2015.

<http://www.mediacabinet.fi/yritys>