



**SAVONIA**

■ OPINNÄYTETYÖ - AMMATTIKORKEAKOULUTUTKINTO  
TEKNIIKAN JA LIIKENTEEN ALA

# WWW-SIVUSTOJEN HALLINTAPANEELI

Bonsait CMS

TEKIJÄ: Mika Ovaskainen

Koulutusala Tekniikan ja liikenteen ala	
Koulutusohjelma/Tutkinto-ohjelma Tietotekniikan tutkinto-ohjelma	
Työn tekijä(t) Mika Ovaskainen	
Työn nimi Sivustojen hallintapaneeli – Bonsait CMS	
Päiväys 17.12.2018	Sivumäärä/Liitteet 24/0
Ohjaaja(t) Lehtori Jussi Koistinen, lehtori Keijo Kuosmanen	
Toimeksiantaja/Yhteistyökumppani(t) Entecore Ky	
Tiivistelmä <p>Opinnäytetyön aiheena oli uudistaa Bonsait-julkaisujärjestelmän sivustojen hallintapaneeli. Tavoitteena oli muuttaa julkaisujärjestelmä itsepalvelutuotteeksi, erottaa REST-rajapinta omaksi mikropalvelukseen, vaihtaa front-end tekniikka sekä lisätä palveluun uusia ominaisuuksia.</p> <p>Uusina ominaisuuksina tulivat muun muassa kokeilujakso, omatoiminen sivuston julkaisu, oikeuksien jakaminen muiden käyttäjien kesken, ilmoitukset, englanninkielinen versio hallintapaneelistä sekä mahdollisuus kirjautua palvelun hallintaan Google- tai Facebook-tilillä. Tunnistautumisen toteuttamiseen käytettiin Google Firebase -pilvipalvelua. Työn toimeksiantajana toimi Entecore Ky.</p> <p>Front-end toteutettiin JavaScriptin React-kirjastoa käyttäen. Rajapinta sen sijaan toteutettiin Java-ohjelmointikielellä, Play-viitekehystä hyödyntäen. Työn lopputuotteena syntyi itsepalvelutuote, joka julkaistaan osaksi Bonsait-julkaisujärjestelmää.</p>	
Avainsanat ReactJS, Java, Bonsait, Rajapinta	

Field of Study Technology, Communication and Transport			
Degree Programme Degree Programme in Information Technology			
Author(s) Mika Ovaskainen			
Title of Thesis Website Dashboard – Bonsait CMS			
Date	December 17, 2018	Pages/Appendices	24/0
Supervisor(s) Mr. Jussi Koistinen, Senior Lecturer and Mr. Keijo Kuosmanen, Senior Lecturer			
Client Organisation /Partners Entecore Ky			
<p>Abstract</p> <p>The subject of the thesis was to reform Bonsait CMS website dashboard. The aim was to alter the content management system into a self-service product, separate REST API as its own microservice, change front-end technique and add new features. In addition, new sign-in options were implemented. The thesis was commissioned by Entecore Ky.</p> <p>The Google Firebase cloud service was used for implementing the sign-in options. The front end was implemented using JavaScript's React library. REST API was implemented using the Java programming language and its Play-framework.</p> <p>As a result of this thesis, a self-service product was implemented. REST API was separated from the CMS, techniques were changed, and the new features were added. New features were for example a trial period, independent website publishing, sharing rights for friends, notifications, an English version of the dashboard and an option to sign in with the Google or Facebook account. Application will be published as part of the Bonsait CMS.</p>			
Keywords ReactJS, Java, API, CMS			

## ESIPUHE

Haluan kiittää erityisesti Mikko Antikaista (Entecore Ky) mielenkiintoisesta opinnäytetyön aiheesta, ohjauksesta, luottamuksesta sekä ulkoasun suunnittelusta. Lisäkiitokset Jussi Koistiselle ohjauksesta, joustavuudesta sekä ripeästä toiminnasta.

Lopuksi haluan kiittää vielä puolisoani Emmiä tuesta ja kannustuksesta.

Mika Ovaskainen

## TERMIT JA LYHENTEET

**Java** on Sun Microsystemsin kehittämä olio-ohjelmointikieli

**Play** on Scala-ohjelmointikieleen pohjautuva viitekehys, jota käytetään ohjelmistojen kehittämiseen

**Kirjasto** on kokoelma aliohjelmiä, joita käytetään ohjelmistojen kehittämiseen

**JavaScript** on komentosarjakieli, jolla voidaan lisätä web-sivustolle dynaamista toiminnallisuutta

**React** on Facebookin kehittämä ja ylläpitämä JavaScript-kirjasto, jota käytetään pääasiassa käyttöliittymien rakentamiseen

**Front- ja back-end** kuvaa sovelluksen eri tasoja: front-end eli ohjelman esitys taso ja back-end eli taso, joka sisältää sovelluksen logiikan

**CMS** (Content Management System) eli sisällönhallintajärjestelmä on järjestelmä, joka palvelee yrityksen tarpeita esimerkiksi verkkosivujen osalta

**REST** (Representational State Transfer) on yleinen arkkitehtuurimalli ohjelmointirajapintojen toteuttamiseen.

## SISÄLLYSLUETTELO

1	JOHDANTO .....	7
2	HALLINTAPANEELI .....	8
2.1	Käyttäjän tunnistautuminen .....	8
2.2	Kokeilujakso .....	9
2.3	Oikeuksien antaminen sivustoihin.....	10
2.4	Sivuston julkaisu .....	12
2.5	Ilmoitukset .....	13
2.6	Haku .....	14
2.7	Kieliversiot.....	14
2.8	Stripe-verkkomaksu .....	15
2.9	REST-Rajapinta.....	15
3	MENETELMÄT JA OHJELMISTOT .....	17
3.1	Java .....	17
3.2	Play-viitekehys.....	17
3.3	ReactJS.....	18
3.4	Bootstrap .....	18
3.5	MongoDB .....	18
3.6	Firebase .....	19
3.7	Eclipse IDE .....	19
3.8	Visual Studio Code .....	20
3.9	Versionhallinta .....	20
4	TOTEUTUS.....	21
5	YHTEENVETO.....	23
	LÄHTEET .....	24

## 1 JOHDANTO

Bonsait on suomalainen sisällönhallinnan ja digitaalisen markkinoinnin verkkopalvelu. Se on joustava ja monipuolinen työkalu, jonka avulla asiakkaan on helppo ylläpitää www-sivustoja sekä toteuttaa markkinointiviestintää. Bonsait on Entecore Ky:n rekisteröity tavaramerkki. (Bonsait, 01.07.2018)

Bonsaitin avulla on helppo muun muassa lähettää uutiskirjeitä, hallita asiakasrekisteriä, luoda kysely- tai tilauslomakkeita ja optimoida verkkosivuston näkyvyyttä hakukoneille. Bonsait XL-paketti sisältää myös asiakkaalle räätälöidyn verkkokaupparatkaisun.

Opinnäytetyön tarkoituksena oli uudistaa Bonsait-julkaisujärjestelmän sivustojen hallintapaneeli. Tavoitteena oli muuttaa julkaisujärjestelmä itsepalvelutuotteeksi, erottaa REST-rajapinta omaksi mikropalvelukseen, vaihtaa front-end tekniikka, sekä lisätä palveluun uusia ominaisuuksia. Uusina ominaisuuksina tulivat kokeilujakso, omatoiminen sivuston julkaisu, oikeuksien jakaminen muiden käyttäjien kesken, ilmoitukset, hallintapaneelin englanninkielinen versio sekä mahdollisuus kirjautua palvelun hallintaan Google -tai Facebook tilillä. Kirjautumisvaihtoehtojen toteuttamiseen käytettiin Google Firebase -pilvipalvelua. Lopputuotteena syntyi palvelu, joka julkaistaan osaksi Bonsait-julkaisujärjestelmää.

Suunnitteluvaiheessa pidimme toimeksiantajani kanssa palaverin, jossa kävimme läpi tarvittavat ominaisuudet ja keskustelimme siitä, kuinka vaativimmat ominaisuudet kannattaisi toteuttaa. Sain kuitenkin itse suunnitella, kuinka pääosa uusista ominaisuuksista toteutettaisiin tiettyjä reunaehtoja noudattaen. Toteutusvaiheessa työskentelin itsenäisesti ja kävimme viikoittain läpi, kuinka muutokset edistyvät. Tässä raportissa käydään läpi hallintapaneelin ominaisuuksia, käytettyjä menetelmiä sekä työn toteutuksen vaiheita.

## 2 HALLINTAPANEELI

Sivustojen hallintapaneelissa käyttäjän on mahdollista selata omia sivustoja, luoda uusia tai muokata olemassa olevan sivuston perusasetuksia. Perusasetuksia ovat esimerkiksi sivuston nimi, verkkotunnus, valittu Bonsait-paketti sekä laskutustiedot. Seuraavaksi kappaleen aliluvuissa esittelen hallintapaneeliin työssäni tekemät muutokset, joita ovat käyttäjän tunnistautuminen, kokeilujakso, oikeuksien jakaminen sivustolle, käyttäjän liittäminen jälleenmyyjään, sivuston julkaisu, ilmoitukset, haku, kieliversiot ja Stripe-verkkomaksupohjat.

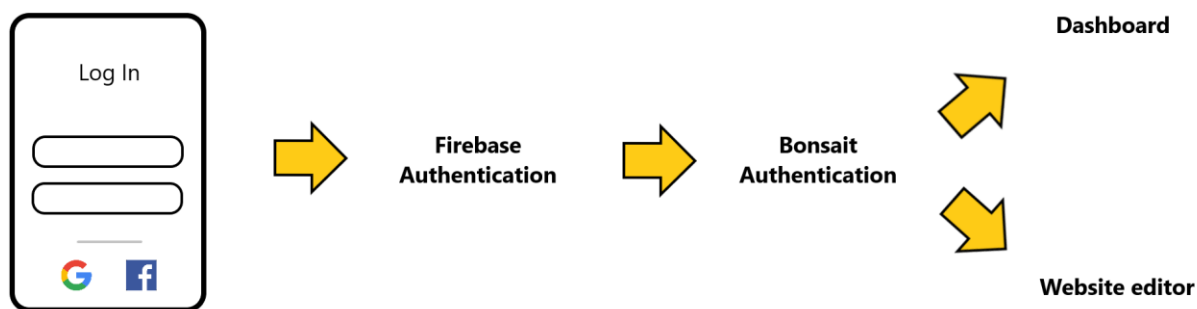
Ulkoasun on suunnitellut opinnäytetyön toimeksiantaja Mikko Antikainen Entecore Ky:stä.

### 2.1 Käyttäjän tunnistautuminen

Suurimpana muutoksena sivustojen hallintapaneeliin oli uudistaa käyttäjän tunnistautuminen palveluun. Aiemmin käyttäjä ei voinut itse rekisteröityä tai luoda sivuja vaan tämä täytyi hoitaa jälleenmyyjän tai suoraan Entecore Ky:n kautta.

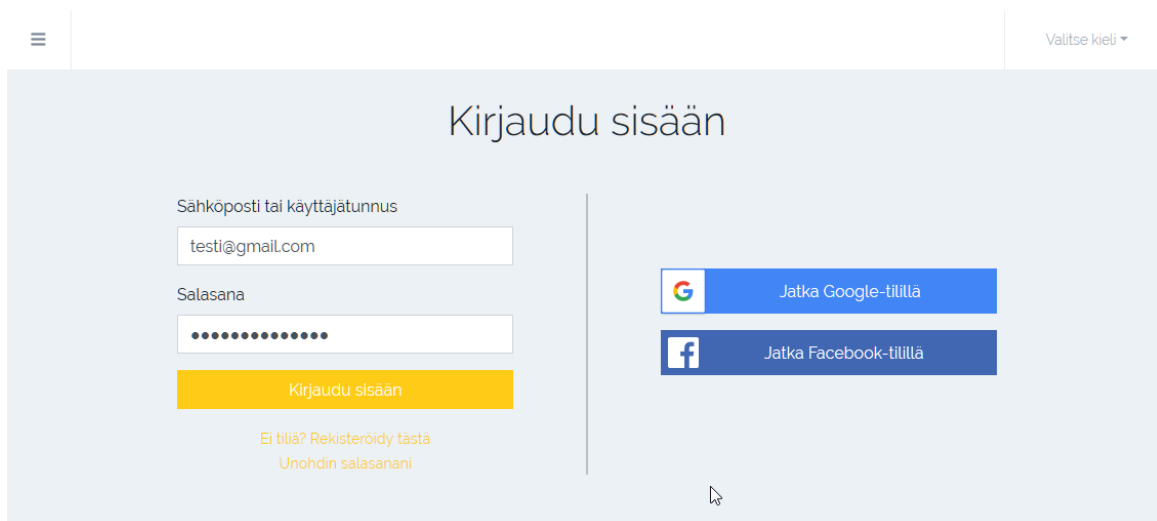
Uudistuksen jälkeen käyttäjä voi itse tunnistautua palveluun Google- tai Facebook-tilin avulla tai halutessaan pelkällä sähköpostiosoitteella ja salasanalla. Muutos käyttäjän tunnistautumiseen vaati myös muutoksen palvelun käyttöehtoihin.

Kuvassa 1 esitetään prosessikaaviolla käyttäjän tunnistautuminen. Kuvassa 2 kirjautumislomake



KUVA 1. Tunnistautumisen periaate

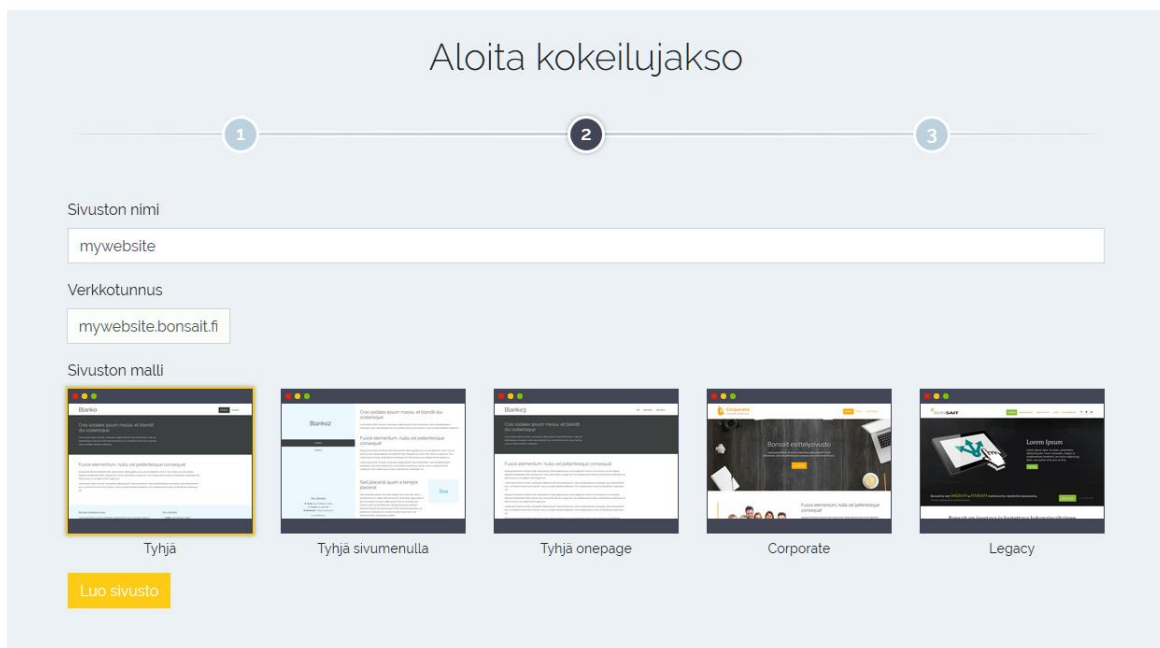
Vanhan käyttäjän ensimmäisen kerran kirjautuessa sivustonsa muokkaustilaan tai hallintapaneeliin ohjataan käyttäjä hyväksymään uudet käyttöehdot, jonka jälkeen luodaan uusi Firebase-tili. Tämän jälkeen käyttäjä voi jatkaa palvelun käyttöä normaalisti.



KUVA 2. Käyttäjän tunnistautuminen

## 2.2 Kokeilujakso

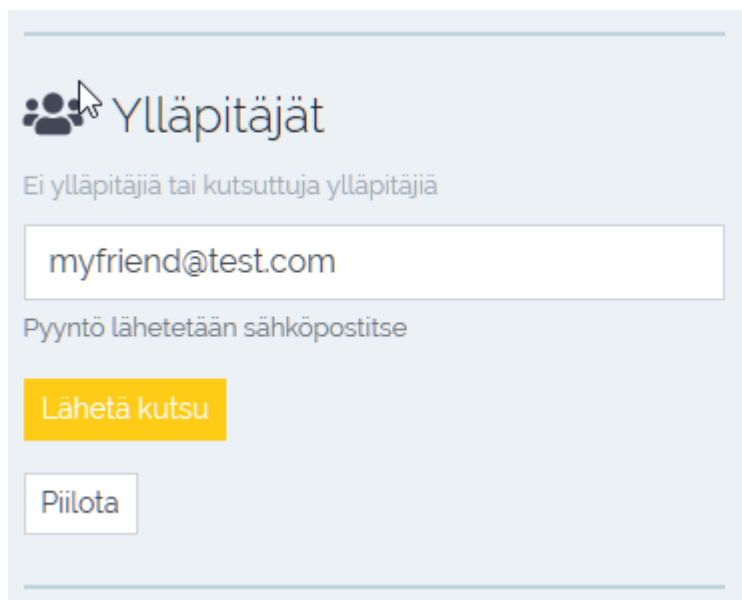
Käyttäjän on mahdollista luoda oma www-sivusto nopean ja helpon kokeilujakso-ohjelman kautta. Käyttäjä tunnistautuu palveluun, jonka jälkeen hän syöttää sivustonsa nimen, verkkotunnuksen, sekä valitsee haluamansa sivupohjan (Kuva 3.). Tämän jälkeen asiakas pääsee muokkaamaan sivustoa. Käyttäjän kokeilujakso kestää kaksi viikkoa.



KUVA 3. Kokeilujakson toinen vaihe

## 2.3 Oikeuksien antaminen sivustoihin

Käyttäjä voi myös halutessaan antaa ystävilleen oikeuden muokata sivustoaan. Kutsu lähetetään sähköpostitse, joka sisältää aktivointilinkin. Mikäli järjestelmässä ei ole käyttäjää kyseisellä sähköpostilla, edellytetään häntä rekisteröitymään palveluun.



Ylläpitäjät

Ei ylläpitäjiä tai kutsuttuja ylläpitäjiä

myfriend@test.com

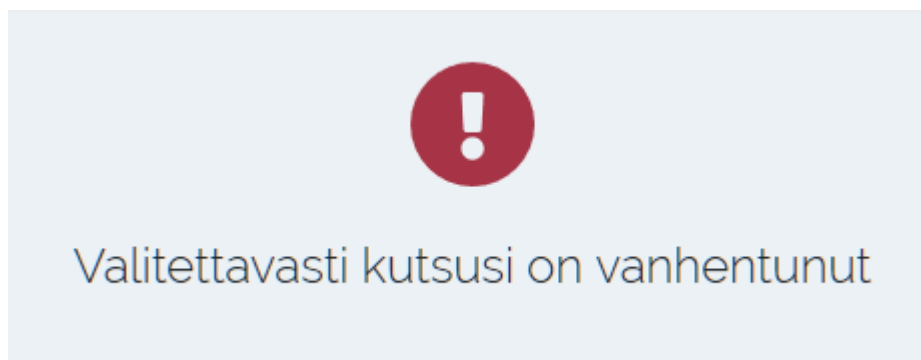
Pyyntö lähetetään sähköpostitse

Lähetä kutsu

Piilota

KUVA 4. Kutsun lähetys

Mikäli sähköpostilla on jo käyttäjä ja hän on kirjautunut palveluun, linkkiä klikkaamalla ilmoitetaan, että hänellä on nyt oikeus muokata sivustoa. Mikäli kutsuttu käyttäjä ei hyväksy oikeutta määräaikaan mennessä, käyttäjälle ilmoitetaan virhetilanteesta.



KUVA 5. Kutsu vanhentunut

Jälleenmyyjien on mahdollista antaa työntekijöilleen oikeus kaikkiin sivustoihinsa. Tällöin jälleenmyyjä lähettää työntekijöiden sähköpostiosoitteisiin linkin, jonka kautta henkilö saa oikeuden kaikkiin jälleenmyyjän sivustoihin. Vahvistettuaan kutsun työntekijä näkee kaikki jälleenmyyjän sivustot hallintapaneelissa, sekä hänellä on myös pääsy sivuston muokkaustilaan.

The screenshot displays the Bonsait account management interface. On the left is a dark sidebar with the Bonsait logo and 'Account' text. Below the logo are menu items: 'SIVUSTOT' (with sub-items 'New website' and 'All websites'), 'TILI', 'JM. TILI', 'KIRJAUDU ULOS', and 'TUKI'. The main content area has a top navigation bar with a search icon and 'Etsi', a user profile 'Mika the Manager', and a language dropdown 'Valitse kieli'. The main content is divided into sections: 'Muokkaa yrityksen tietoja' (Edit company information) with fields for 'Yrityksen nimi' (Mikan mainostoimisto Ky) and 'Oletus verkkotunnus' (.mikantoimisto.fi), and 'Lisää käyttäjiä' (Add users) with a search field containing 'testi@gmail.com' and a 'Lisää' button. Below this is a 'Kutsutut käyttäjät' (Invited users) section showing a card for 'testi@gmail.com' with an envelope icon and a close button.

KUVA 6. Jälleenmyyjän sivu

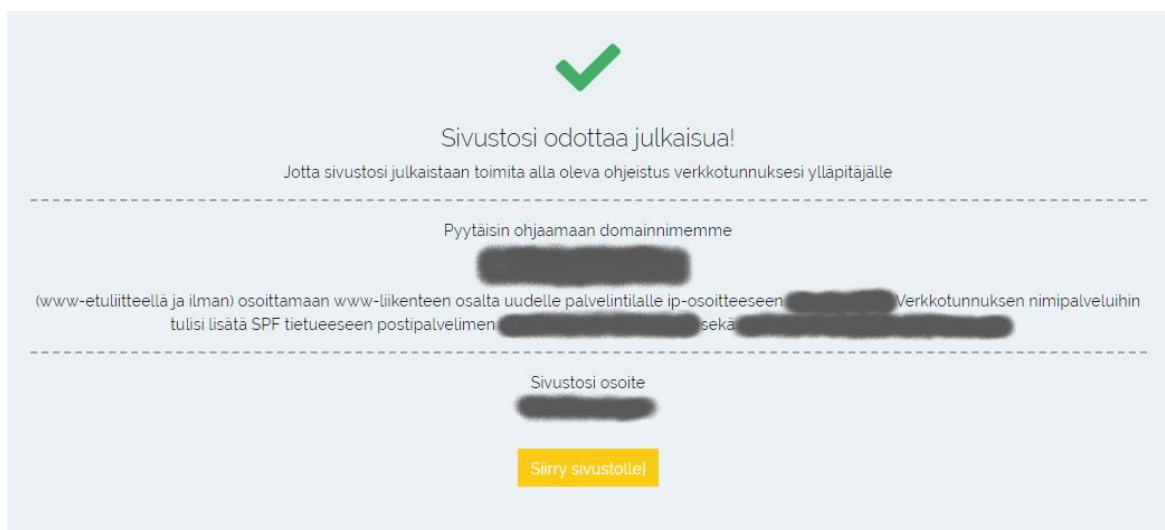
## 2.4 Sivuston julkaisu

Käyttäjän on mahdollista julkaista sivusto kolmella eri tavalla. Käyttäjä voi valita julkaistaanko sivusto bonsait.fi -päätteisellä demo-verkkotunnuksella vai jo olemassa olevalla verkkotunnuksella. Lisäksi käyttäjä voi pyytää Entecore Ky:tä varaamaan halutun verkkotunnuksen.

The screenshot shows a web form for publishing a website. At the top, there is a navigation bar with a search icon and the text 'Etsi', a user profile 'Mika Facebook', and a language selection dropdown 'Valitse kieli'. The main form area is titled 'Julkaistava sivusto: Mikan Facebook sivu'. It contains three radio buttons for selection: 'Julkaise osoitteella: mikanfacebookisivu.bonsait.fi', 'Tilaa uusi verkkotunnus', and 'Julkaise jo olemassa olevalla osoitteella'. Below these are two browser tabs for 'mikanfacebookisivu.fi' and 'www.mikanfacebookisivu.fi', with a 'Lisää verkkotunnuksia' link. A note says 'Kirjoita kenttään verkkotunnukset. Tämän jälkeen verkkotunnukset täytyy...'. The 'Laskutus' section includes dropdown menus for 'Laskutustapa' (set to 'Lasku'), 'Laskutuskausi' (set to '12 kk'), and 'Lähetystapa' (set to 'Posti'). There are input fields for 'Laskutusosoite', 'Laskutusosoite 2', 'Postitoimipaikka' (set to 'Kuopio'), 'Postinumero' (set to '70500'), 'Verkkolaskutusosoite', and 'Verkkolaskutusoperaattori'. At the bottom, a note reads 'Voit lisätä verkko-osoitteita myöhemmin, mutta ET VOI peruuttaa julkaisua'. There are 'Peruuta' and 'Julkaise' buttons.

KUVA 7. Julkaisulomake

Täytettyään lomakkeen onnistuneesti, käyttäjälle ilmoitetaan julkaisun onnistumisesta, sekä tarvittavista toimenpiteistä. Esimerkiksi jos käyttäjä valitsee julkaisun jo olemassa olevalla osoitteella, käyttäjälle annetaan ohjeet verkkotunnuksen siirtoprosessia varten (Kuva 8.).



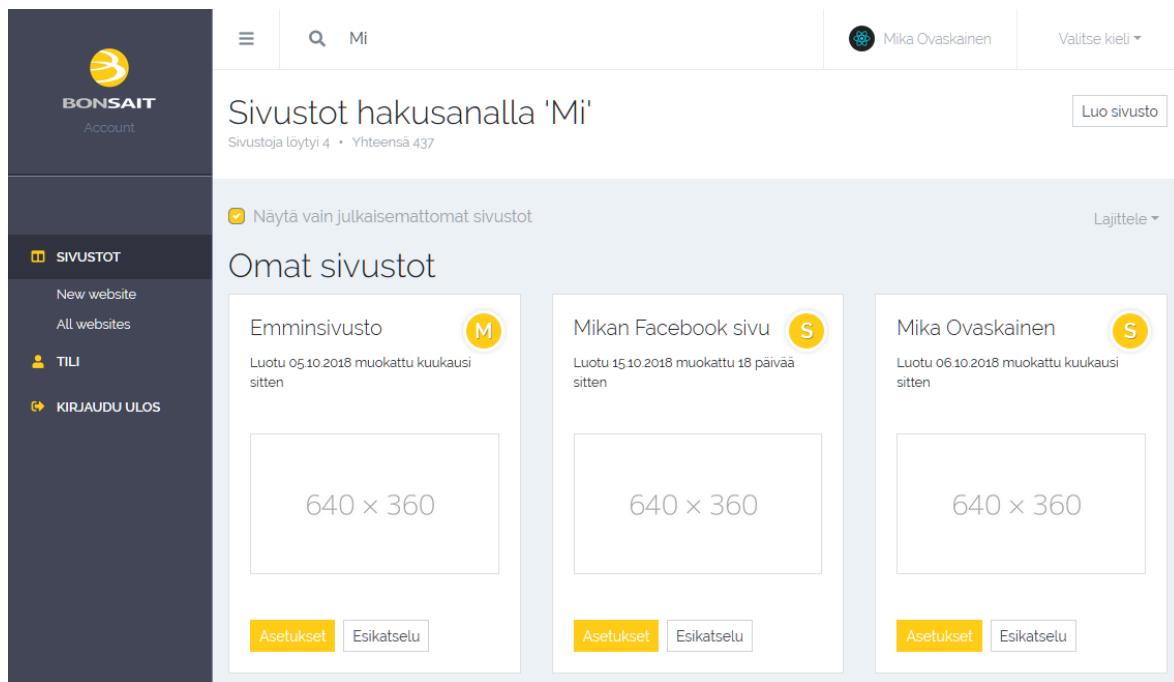
KUVA 8. Ohjeet verkkotunnuksen siirtoprosessiin

## 2.5 Ilmoitukset

Sovellukseen rakennettiin tulevaisuutta varten myös mahdollisuus vastaanottaa ilmoituksia. Tätä kautta käyttäjä saa halutessaan selaimen ja mobiilisovellukseen ilmoituksen esimerkiksi julkaisujärjestelmän uusista ominaisuuksista tai muutoksista.

## 2.6 Haku

Käyttäjän on mahdollista hakea omia sivustoja nopean hakutoiminnon avulla. Halutessaan käyttäjä voi myös suodattaa pois julkaistut sivustot. Tämä toiminto on erittäin hyödyllinen esimerkiksi jälleenmyyjille, joilla saattaa olla kymmeniä sivustoja. Sivustot voidaan myös lajitella nimen, julkaisupäivän tai luontipäivämäärän perusteella.



KUVA 9. Hakutoiminto

## 2.7 Kieliversiot

Sivustojen hallintapaneeliin lisättiin myös mahdollisuus muuttaa paneelin kieli englanniksi. Kieliversioiden toteutukseen käytettiin react-i18next -lisäosaa. Lisäosa on erittäin helppokäyttöinen eikä kielen vaihtaminen vaadi sivun uudelleen lataamista. Käännökset lisätään JSON-tiedostoon ja tulostetaan funktion kautta haluttuun kohtaan komponentissa.

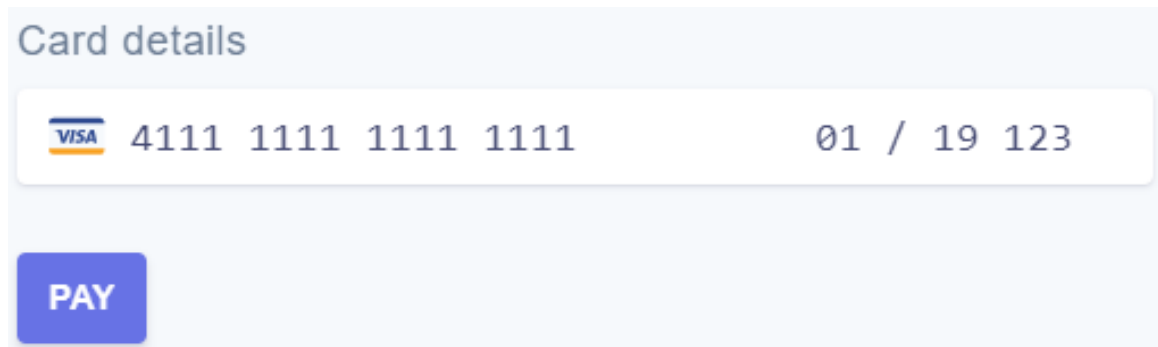
```
<ul className="list-group list-group-flush" style={{fontSize: "0.85rem"}}>
  <li className="list-group-item p-2">{{t("plan.cms")}}</li>
  <li className="list-group-item p-2">{{t("plan.webshop")}}</li>
  <li className="list-group-item p-2">{{t("plan.newsletters")}}</li>
  <li className="list-group-item p-2">{{t("plan.register")}}</li>
  <li className="list-group-item p-2">{{t("plan.intra")}}</li>
  <li className="list-group-item p-2">{{t("plan.blog")}}</li>
  <li className="list-group-item p-2">{{t("plan.langunlimit")}}</li>
  <li className="list-group-item p-2">{{t("plan.supportprio")}}</li>
</ul>
```

```
"plan": {
  "title": "Paketti",
  "cms": "Julkaisualusta",
  "lang": "kieliversio",
  "langmulti": "kieliversiota",
  "support": "Asiakastuki sähköpostilla",
  "supportprio": "Asiakastuki priority",
  "intra": "Intra- ja extranet",
  "blog": "Blogi",
  "newsletters": "Uutiskirjeet",
```

KUVA 10. Esimerkki react-i18next lisäosan käytöstä

## 2.8 Stripe-verkkomaksu

Palveluun toteutettiin tulevaisuutta varten pohja Stripe-verkkomaksuille. Käyttäjä voisi siis halutesaan maksaa sivustonsa kuukausimaksun luottokortilla perinteisen paperi- tai verkkolaskun sijaan. Kuvassa 11 käyttäjää pyydetään syöttämään luotto- tai pankkikortin tiedot lomakkeelle.



The image shows a 'Card details' form. At the top, it says 'Card details'. Below that, there is a white rounded rectangle containing a Visa logo on the left, followed by the card number '4111 1111 1111 1111' in the center, and the expiration date '01 / 19 123' on the right. Below this rectangle is a blue button with the word 'PAY' in white capital letters.

KUVA 11. Luottokortin tiedot

Lomakkeen validoinnin jälkeen tiedot lähetetään Stripe-rajapintaan, joka palauttaa vastauksen maksupyynnön tilasta. Mikäli maksupyyntö onnistui, näytetään käyttäjälle yhteenveto tilauksesta. Maksutavan pohjan toteutukseen käytettiin react-stripe-elements -lisäosaa.

## 2.9 REST-Rajapinta

REST-rajapinnalla tarkoitetaan lyhykäisyydessään arkkitehtuurityyliä, jota käytetään sovellusten väliin kommunikointiin. Esimerkiksi kaikki mobiilisovellukset tai JavaScriptillä toteutetut selainsovellukset, joissa käsitellään palvelimella sijaitsevaa dataa, käyttävät poikkeuksetta jonkinlaista REST-rajapintaa. REST rakentuu http-protokollan päälle (KIVISAARI, T. 2016. 18.10.2018).

Rajapinta ottaa vastaan kaikki hallintapaneelin puolelta tulevat pyynnöt ja suorittaa tarvittavat tehtävät. Tarvittaessa rajapinta palauttaa pyydettyä dataa, kun taas virhetilanteessa palautetaan tieto, että jotain meni pieleen.

Opinnäytetyössäni osa rajapinnan toiminnoista oli jo olemassa. Tehtävänäni oli erottaa rajapinta julkaisujärjestelmästä niin sanotuksi mikro-palveluksi, sekä lisätä rajapintaan logiikka uusille ominaisuuksille. Käytännössä tämä tarkoitti sitä, että loin kokonaan uuden projektin, jonka päälle aloin rakentamaan rajapintaa. Rajapintaan toteutettiin muun muassa sivustojen luonti-, listaus-, päivitys- ja poisto toiminnot, sekä tarvittavat metodit käyttäjän tunnistautumiselle, sivustojen julkaisulle ja oikeuksien jakamiselle. Rajapinnasta kaikki tarvittava data palautetaan JSON-muodossa, koska sitä on helppo käsitellä JavaScriptillä.

```
int start = 0;
if (request().getQueryString("start") != null) {
    start = Integer.parseInt(request().getQueryString("start"));
}

int limit = 21;
if (request().getQueryString("limit") != null) {
    limit = Integer.parseInt(request().getQueryString("limit"));
}

boolean showall = true;
if (request().getQueryString("showonlyunpublished") != null && request().getQueryString("showonlyunpublished").equals("true")) {
    showall = false;
}

String sortby = "created";
if (request().getQueryString("sortby") != null && !request().getQueryString("sortby").equals("")) {
    sortby = request().getQueryString("sortby");
}

int order = -1;
if (request().getQueryString("reverse") != null && request().getQueryString("reverse").equals("true")) {
    order = 1;
}
```

KUVA 12. Rajapinnassa suoritettava suodatustoiminto

Rajapinta ottaa vastaan asiakasohjelmasta lähetetyn pyynnön, josta suoritetaan parametrien perusteella tietokanta haku. Tämän jälkeen rajapinta palauttaa datan asiakasohjelmalle, jossa se parsitaan käyttäjälle näytettävään muotoon.

### 3 MENETELMÄT JA OHJELMISTOT

Tässä osiossa käydään läpi opinnäytetyössä käytettyjä menetelmiä ja ohjelmistoja. Menetelmät olivat itselleni työn puolesta entuudestaan tuttuja, pois lukien ReactJS ja Firebase -pilvipalvelut. Lyhyen perehtymisen jälkeen, pääsin kuitenkin näihinkin sisään melko nopeasti.

#### 3.1 Java

Java on Sun Microsystemsin kehittämä olio-ohjelmointikieli. Java on yksi maailman suosituimpia ohjelmointikieliä. Olorakenteensa ansiosta Java-ohjelmia on helppo laajentaa sekä kielenä Java on nopea oppia. Hyvinä ominaisuuksina voidaan mainita esimerkiksi multi-threading, jolla mahdollistetaan kehittäjän usean tehtävän samanaikainen suorittaminen (Java Overview, Tutorialspoint).

Opinnäytetyössä rajapinta toteutettiin Javaa tukevan Play-viitekehityksen avulla. Java valittiin ohjelmointikieleksi, sillä se on yleisesti käytössä Entecore Ky:llä.

#### 3.2 Play-viitekehitys

Play on avoimen lähdekoodin viitekehitys, joka on ohjelmoitu Scala-ohjelmointikielellä. Sitä voidaan käyttää myös muilla ohjelmointikielillä, jotka käännetään bittikoodiksi. Tässä tapauksessa Javalla. Viitekehityksellä on mahdollista toteuttaa kokonaisia internetsovelluksia, mutta opinnäytetyössä sitä käytettiin ainoastaan REST-rajapinnan toteutukseen. Kuvassa 13 esimerkki Play-viitekehityksellä toteutetusta rajapinnan metodista ja sen vastauksesta.

```
public Result sayHello() {  
    ObjectNode result = Json.newObject();  
    result.put("exampleField1", "foobar");  
    result.put("exampleField2", "Hello world!");  
    return ok(result);  
}
```

HTTP/1.1 200 OK  
Content-Type: application/json; charset=utf-8  
{"exampleField1":"foobar","exampleField2":"Hello world!"}

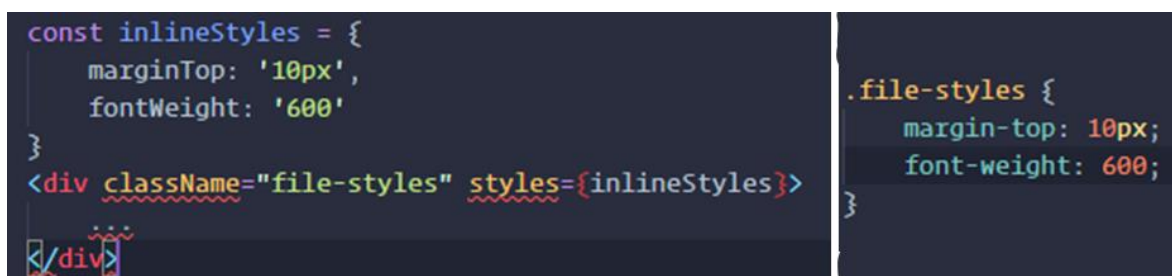
KUVA 13. Rajapinnan metodi ja vastaus

### 3.3 ReactJS

ReactJS on Facebookin luoma komponenttipohjainen JavaScript-kirjasto. React:n avulla voidaan kehittää tehokkaasti interaktiivisia käyttöliittymiä (React). Lisäksi React on erittäin nopea päivittämään käyttöliittymää. Tämä johtuu siitä, että kirjasto muuttaa vain tarvittavaa komponenttia tiedon näyttämiseen.

ReactJS:ssä suositetaan yleisesti inline-tyylejä. Inline-tyyleillä tarkoitetaan CSS:n lisäämistä suoraan HTML-elementtiin. Jos kuitenkin halutaan luoda tyyliä CSS-tiedostoon, sekin on mahdollista toteuttaa. Tällöin HTML-elementtiin lisätään 'className' -attribuutti.

Opinnäytetyössäni toteutin suurimman osan tyyleistä CSS-tiedostoon, sillä kokemusteni perusteella tyylien erottaminen muusta koodista selventää kokonaisuutta huomattavasti. Lisäksi samoja CSS-luokkia voidaan käyttää myös kätevästi muualla. CSS-luokat ovat myös parempi vaihtoehto sovelluksen suorituskyvyn kannalta (Styling and CSS).



```

const inlineStyles = {
  marginTop: '10px',
  fontWeight: '600'
}
<div className="file-styles" styles={inlineStyles}>
  </div>

```

```

.file-styles {
  margin-top: 10px;
  font-weight: 600;
}

```

KUVA 14. Esimerkki inline-tyyleistä sekä CSS-tiedostossa sijaitsevista tyyleistä

### 3.4 Bootstrap

Bootstrap on ilmainen, avoimen lähdekoodin kirjasto, jota käytetään responsiivisten selainsovellusten kehittämiseen HTML:n, CSS:n ja JavaScriptin osana. Bootstrap sisältää esirakennettuja komponentteja sekä lisäosia, jotka tekevät sovelluksen kehittämisestä huomattavasti nopeampaa (Bootstrap). Opinnäytetyössäni hyödynsin Bootstrap:n responsiivistä kolumni -rakennetta, muutamia esirakennettuja komponentteja sekä SASS-muuttujia.

### 3.5 MongoDB

MongoDB on ilmainen avoimen lähdekoodin asiakirjapohjainen tietokanta, jonne data tallennetaan JSON-formaatissa (MongoDB). Relaatiotietokantoihin verrattuna MongoDB:n etuja ovat kannan skaalautuvuus tiedon määrän kasvaessa sekä kyselyiden suorituskyky. Lisäksi etuna voidaan pitää helppoa ja yksinkertaista tietokannan käsittelyä.

SQL-kannoissa rivit tallennetaan tauluihin, kun taas Mongossa vastaavat dokumentit tallennetaan kokoelmiin. Dokumentti voi olla hierarkkinen, jolloin monen taulun asiat voidaan tallettaa yhteen

dokumenttiin. Tallennusmuotona JSON helpottaa ohjelmointia, koska oliorakennetta ei tarvitse muuttaa relaatiomuotoon. JSON on yleistynyt rajapintojen muotona oliorakenteensa ansiosta. Huonona puoleena MongoDB:ssä voidaan mainita yhdistetyt haut, jotka voivat olla monimutkaisia ja hitaita verrattuna SQL-tietokantoihin. (HOVI, A. 2015)

MongoDB on tällä hetkellä maailman viidenneksi käytetyin tietokanta (ARSENAULT, C. 2017). Se on ilmainen, helppo oppia ja nopea asentaa, eikä vaadi ylimääräisiä konfigurointeja. Tästä syystä se onkin suosittu erityisesti aloittelevien ohjelmoijien keskuudessa.

Opinnäytetyössäni kaikki data tallennettiin Mongo-tietokantaan, ettei julkaisujärjestelmään tarvitsisi tehdä muutoksia. Firebasen tietokantaa käytettiin ainoastaan käyttäjän tunnistautumiseen, siihen liittyvän datan tallentamiseen, sekä ilmoituksiin. Kyseinen toimintamalli mahdollistaa myös sen, että tekniikoita voidaan muuttaa tarvittaessa vaikuttamatta toiseen.

### 3.6 Firebase

Firebase on Googlen kehittämä mobiilialusta, jonka avulla kehittäjä voi nopeasti luoda korkealaatuisia sovelluksia (Firebase). Opinnäytetyössäni palveluista on käytössä Firestore -tietokanta, Authentication ja Cloud Messaging.

Firestore on NoSQL -tietokanta, johon tieto tallennetaan dokumenttimuodossa, aivan kuten MongoDB:ssä. Datatyyppinä toimii JSON. Firestore valittiin käyttöön sen vuoksi, että kyselyt tietokantaan ovat yksinkertaisia eikä sovellukseen tarvita reaaliaikaisuutta.

Cloud Storage on Google-palveluille rakennettu yksinkertainen objektien tallennuspalvelu. Cloud Storageen voidaan tallentaa esimerkiksi kuvia ja tiedostoja. Tallennuspalvelusta voidaan myös ladata tallennettuja tiedostoja Firebasea käyttäviin sovelluksiin (Cloud Storage).

Authentication-osioon tallennetaan käyttäjän tunnistautumistiedot. Tämä mahdollistaa sovellukseen kirjautumisen ja rekisteröitymisen joko sähköpostiosoitteella ja salasanalla tai esimerkiksi Googlen ja Facebookin avulla.

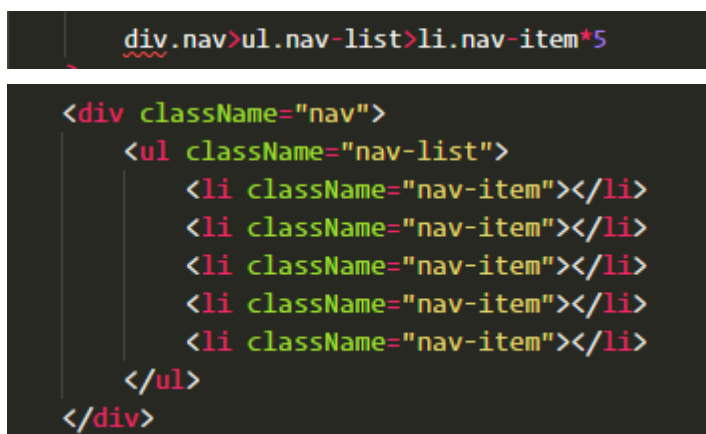
Cloud Messaging -toiminnon kautta käyttäjälle voidaan lähettää esimerkiksi ilmoituksia tai uutisia joko selaimeen tai mobiilisovellukseen (Firebase Cloud Messaging). Palvelun avulla on myös mahdollista myös luoda reaaliaikaisia viestintä -sovelluksia.

### 3.7 Eclipse IDE

Kehitysympäristönä opinnäytetyössä palveli back-end -puolella Eclipse Oxygen hyvän suorituskyvyn ansiosta. Lisäksi Eclipsessä on erittäin tehokkaat virheentarkastelu-työkalut.

### 3.8 Visual Studio Code

Front-end puolella kehitysympäristönä oli käytössä Microsoftin kehittämä moderni tekstieditori Visual Studio Code. VSCode on noussut nopeasti yhdeksi maailman käytetyimmäksi tekstieditoriksi. Editoriin on saatavilla lukuisia määriä lisäosia, teemoja, sekä fontteja. Lisäosia voidaan asentaa muun muassa tukemaan useita ohjelmointikieliä tai helpottamaan ohjelman kirjoittamista. Sisäänrakennettuna VSCode:ssa on esimerkiksi Emmett-lisäosa, joka mahdollistaa nopean HTML-koodin kirjoittamisen. Käytännössä tämä tarkoittaa sitä, että ohjelmoijan tarvitsee ainoastaan kirjoittaa tietynlaisia lyhenteitä, jonka perusteella lisäosa generoi HTML-koodia (Kuva 15). Lisäksi VSCode sisältää oman komentorivin.



The image shows a code editor with a dark background. At the top, the Emmet abbreviation `div.nav>ul.nav-list>li.nav-item*5` is shown. Below it, the expanded HTML code is displayed:

```
<div className="nav">
  <ul className="nav-list">
    <li className="nav-item"></li>
    <li className="nav-item"></li>
    <li className="nav-item"></li>
    <li className="nav-item"></li>
    <li className="nav-item"></li>
  </ul>
</div>
```

KUVA 15. Esimerkki lyhenteestä, jonka perusteella Emmet generoi HTML-koodia

### 3.9 Versionhallinta

Versionhallintaan projektissa käytettiin Git-versionhallintaohjelmistoa. Versionhallinnan tarkoituksena on pitää kirjaa koodin muutoksista ohjelmaa kehitettäessä. Ohjelma on erittäin hyödyllinen esimerkiksi tilanteissa, jolloin ohjelmoija tekee virheen. Hän voi palauttaa kehittämänsä sovelluksen vanhaan versioon, kun kaikki vielä toimii oikein. Git helpottaa myös huomattavasti työskentelyä projektiryhmässä. Ohjelmistokehittäjille jaetaan usein työtehtäviä ominaisuuskohtaisesti, jolloin yksi kehittäjä toteuttaa tarvittavan ominaisuuden alusta loppuun. Tällaisissa tapauksissa on kätevää luoda niin sanottuja haaroja (branch) Git-versionhallintaan, joka voidaan myöhemmin yhdistää päähaaraan (master branch), jossa uusin toimiva versio sijaitsee. Mikäli kehittäjät ovat muokanneet samoja tiedostoja ristiin, ohjelmisto ilmoittaa yhdistämisvaiheessa, että tiedostoissa on konflikti. Tällä vältetään ohjelman kaatumiselta, joka voi joissain tapauksissa käydä myös kalliiksi. Git on yksi tämän hetken suosituimmista versionhallintaohjelmistoista (Atlassian).

## 4 TOTEUTUS

Opinnäytetyön alkuvaiheessa määriteltiin projektissa kehitettävät ominaisuudet sekä käytettävät kehitysmenetelmät, joilla työ lähdetään toteuttamaan. Suunnitteluvaiheessa käytiin teoriassa läpi, mitä vaiheita kuhunkin kehitettävään ominaisuuteen kuuluu ja kuinka ne kannattaisi koodin puolella toteuttaa. Ominaisuuksia havainnollistettiin erilaisilla näyttökuvilla ja kaavioilla.

Toteutusvaiheessa ensimmäisenä työstettiin kieliversion vaihto, jotta kaikki sivustolla näkyvä teksti voidaan kääntää suoraan toiselle kielelle. Seuraavaksi toteutettiin uusi tunnistautumislogiikka, sillä suurin osa palvelusta on tunnistautumisen takana. Tunnistautuminen oli ehdottomasti sovelluksen monimutkaisin ominaisuus. Muutoksessa täytyi ottaa huomioon sekä vanhojen että uusien käyttäjien ja jälleenmyyjien tunnistautuminen. Ensimmäisenä toteutettiin rekisteröityminen sähköpostiosoitteella ja salasanalla, jonka jälkeen mukaan lisättiin Google- ja Facebook-tunnusten käyttömahdollisuus. Tämän jälkeen toteutettiin logiikka vanhojen käyttäjien ja jälleenmyyjien osalta.

Tunnistautumislogiikan testauksen jälkeen toteutettiin sovelluksen CRUD-toiminnot (create, read, update, delete) eli sivustojen luonti-, listaus-, päivitys- ja poistotoiminnot. Samalla tehtiin myös sivustojen hakutoiminto, sivumenu sekä ylätunniste (header).

Seuraavaksi lisättiin muita uusia ominaisuuksia pala kerrallaan seuraavan listauksen mukaisesti:

1. Mahdollisuus antaa ystävälle oikeus sivustoon
  - a. Olemassa olevan käyttäjän sähköpostikutsu
  - b. Uuden käyttäjän sähköpostikutsu
  - c. Oikeuden hyväksyminen
2. Sivuston julkaisuprosessi
  - a. Julkaisu nykyisellä verkkotunnuksella
  - b. Julkaisu muulla verkkotunnuksella
  - c. Julkaisu toivotulla verkkotunnuksella
3. Uudet maksutavat (Stripe luottokorttimaksu sekä verkkolasku)
  - a. Tarvittavat validoinnit lomakkeen yhteyteen
4. Konfigurointi mahdollisten ilmoitusten vastaanottamiseen
5. Kokeilujakso
  - a. Käyttäjän tunnistautuminen
  - b. Yksinkertaistettu versio sivuston luonnista
6. Jälleenmyyjän oma sivu
  - a. Olemassa olevan käyttäjän sähköpostikutsu
  - b. Uuden käyttäjän sähköpostikutsu
  - c. Oikeuden hyväksyminen

Jokaiselle yllä mainitulle toiminnolle tehtiin ensin front-end -puoli tyylejä myöten kuntoon, jonka jälkeen toteutettiin back-end -logiikka. Lopuksi sovellukselle tehtiin lukuisia testiajoja ja se jätettiin odottamaan tuotantoon laittoa.

## 5 YHTEENVETO

Opinnäytetyön tavoitteena oli muuttaa Bonsait-julkaisujärjestelmä itsepalvelutuotteeksi sekä lisätä palveluun uusia ominaisuuksia. Lopputulos vastasi suunnitelmaa ainoana erona, että Stripe-maksutavalle sekä ilmoituksille tehtiin toistaiseksi ainoastaan pohjakoodit. Palveluun tehdään mahdollisesti vielä lisäominaisuuksia tarpeiden mukaan.

Suunnitellut aikataulut venyivät monien tekijöiden summana, mutta aikataulu oli vain suuntaa antava. Tärkeintä toimeksiantajalle kuitenkin oli, että työ valmistui syksyn aikana.

Mielestäni työ onnistui erittäin hyvin. Toteutusta hankaloitti hieman, että en ollut aikaisemmin toteuttanut sovelluksia ReactJS-kirjastoa käyttäen. Lisäksi minulla oli töiden ja opiskelujen ohessa aikaa opinnäytetyöhön ainoastaan iltaisin ja viikonloppuisin. Vaikka näyttökuvien perusteella sovelluksen toteutus vaikuttaa yksinkertaiselta, oli back-end -logiikan tuottaminen erittäin monimutkainen ja huolellisesti tehtävä prosessi.

Projektin aikana opin monia tärkeitä asioita ohjelmistokehityksestä. Huomasin, kuinka tärkeässä asemassa tarkka suunnittelu ja vaiheistus todellisuudessa on. Lisäksi opin itselleni uuden JavaScript-kirjaston käytön, harjaannuin yleisesti Java-ohjelmoinnissa ja rajapintojen toteutuksessa.

## LÄHTEET

Bonsait, (Viitattu 01.07.2018) Saatavissa:

<https://www.bonsait.fi/>

Java Overview, (Viitattu 07.07.2018) Saatavissa:

[https://www.tutorialspoint.com/java/java\\_overview.htm](https://www.tutorialspoint.com/java/java_overview.htm)

ReactJS, (Viitattu 25.07.2018) Saatavissa:

<https://reactjs.org/>

Styling and CSS, (Viitattu 25.07.2018) Saatavissa:

<https://reactjs.org/docs/faq-styling.html>

What is MongoDB? (Viitattu 01.08.2018) Saatavissa:

<https://www.mongodb.com/what-is-mongodb>

HOVI, A. 2015. MongoDB haastaa relaatiokantoja. (Viitattu 10.10.2018) Saatavissa:

<https://www.arihovi.com/mongodb-haastaa-relaatiokantoja/>

Bootstrap, (Viitattu 10.10.2018) Saatavissa:

<https://getbootstrap.com/>

Firebase, (Viitattu 17.10.2018) Saatavissa:

<https://firebase.google.com/>

Cloud Storage (Viitattu 17.10.2018) Saatavissa:

<https://firebase.google.com/docs/storage/>

Firebase Cloud Messaging (Viitattu 17.10.2018) Saatavissa:

<https://firebase.google.com/docs/cloud-messaging/>

KIVISAARI, T. 2016. (Viitattu 18.10.2018) Saatavissa:

<https://blog.digia.com/rest-api>

ARSENAULT, C. 2017 (Viitattu 24.10.2018) Saatavissa:

<https://www.keycdn.com/blog/popular-databases>

Atlassian Bitbucket, (Viitattu 25.10.2018) Saatavissa:

<https://www.atlassian.com/git/tutorials/what-is-version-control>