

## Bougoudia Ry:n verkkosivut

Jami Lu

Opinnäytetyö

Tietojenkäsittelyn koulutusohjelma

2014



<b>Tekijä tai tekijät</b> Jami Lu	<b>Ryhmätunnus tai aloitusvuosi</b> 2009
<b>Raportin nimi</b> Bougoudia Ry:n verkkosivut	<b>Sivu- ja liitesivumäärä</b> 20 + 26
<b>Opettajat tai ohjaajat</b> Arvo Lipitsäinen	
<p>Bougoudia Ry on hyväntekeväisyys yhdistys, joka tekee hyväntekeväisyystyötä Afrikassa. Tämän yhdistyksen kohteena on auttaa naisten ja lasten asemaa opiskelussa. Yhdistyksen toiminta keskittyy Bakari Bougoun ja Nasson kylissä.</p> <p>Yhdistys tarvitsee verkkosivuston tietokeskukseksi, josta löytyy yhdistyksen tiedotteita ja muuta tietoa yhdistyksestä. Verkkosivuston muutamat tärkeimmät ominaisuudet ovat henkilö voi rekisteröityä kummiksi tai jäseneksi ja henkilö voi selata kummilapsia.</p> <p>Tavoite on toteuttaa ensimmäisen asteen verkkosivut, josta voidaan selata kummilapsia, tiedotteita, muita tietoja liittyen yhdistykseen, lukea yhdistyksen säännöt, katsoa kuvia, lukea kummitoiminnasta ja liittyä jäseneksi tai kummiksi. Projektin ulkopuolelle jää suoramaksulomakkeen toteuttaminen ja ne ominaisuudet joita ei saada määrättyssä ajassa täysin valmiiksi.</p> <p>Verkkosivusto toteutetaan Java Spring Frameworkillä ja HTML 5, CSS 3, JavaScript, jQuery ja AJAX tekniikoita käyttäen. Ohjelmointi tapahtuu eclipse ohjelmalla. Projektin aikana pyritään käyttämään Scrum menetelmiä.</p> <p>Bougoudia Ry yhdistyksen verkkosivusto toteutettiin opinnäytetyönä 12.12.2013 - 30.4.2014 välisenä aikana.</p> <p>HTML 5, CSS 3, JavaScript ja jQuery tekniikoiden ansiosta verkkosivustosta saatiin toteutettua moderni ja käyttäjäystävällinen. AJAX mahdollisti sujuvan kummilasten hallinnan. JSP ja JSTL tekniikoilla toteutettiin palvelimen vastausten esittäminen verkkosivulla. Spring Security toteutti sovelluksen autentikoinnin ja Spring jdbc toteutti sovelluksen tietokanta yhteyden. MySQL valittiin sovelluksen tietokannaksi.</p>	
<b>Asiasanat</b> Java, ohjelmointi, sovelluskehikset, tietokannat, SQL, kuvankäsittely	

Degree programme in Information Technology

<b>Authors</b> Jami Lu	<b>Group or year of entry</b> 2009
<b>The title of thesis</b> Bougoudia ry's website	<b>Number of report pages and attachment pages</b> 20 + 26
<b>Advisor(s)</b> Arvo Lipitsäinen	
<p>Bougoudia ry is a charity organization that does charity work in Africa. The target of the organization is to help women's and children's position in studies. The operations of the organization are focused on Bakari Bougou and Nasso villages.</p> <p>The organization needs a website to serve as an information center where it is possible to find announcements and further information concerning the organization. The website's most important features are that a person is able to register to the website as a member or a child sponsor and the person is able to browse children that can be sponsored.</p> <p>The purpose of this thesis was to implement the first level website, where the visitors are able to browse the children that can be sponsored, and they can find announcements and further information concerning the organization including the rules of the organization. It is possible look at photos, read about sponsoring the children and register to the website as a member or as a sponsor. Implementing the direct payment form and some other features that could not be finished within the deadline of this study were excluded from the scope of this thesis project.</p> <p>The website was implemented on Java Spring Framework and using HTML 5, CSS 3, JavaScript, jQuery and AJAX techniques. Programming was done using the Eclipse program. During the project, the Scrum technique was intended to be used.</p> <p>Because of the HTML 5, CSS 3, JavaScript, jQuery techniques, the website could be done to be modern and user-friendly. AJAX made it possible to administer the browsing of the children in a sensible manner. The JSP and the JSTL techniques were implemented for showing a server's responses on the webpage. Spring Security was used for the authentication of the application and Spring jdbc was used as the database connection to the application. MySQL was chosen as the database of the application.</p>	
<b>Key words</b> Java, programming, frameworks, databases, SQL, photo manipulation	

# Sisällys

1 Johdanto .....	1
1.1 Tavoitteet.....	1
1.2 Aiheen rajausta.....	1
1.3 Tekniikat ja menetelmät.....	2
2 Teoriatausta.....	3
2.1 Selainpuolen tausta.....	3
2.2 Palvelinpuolen tausta .....	4
3 Suunnittelu ja Toteutus.....	6
3.1 Kohde .....	6
3.2 Kehityksen haasteet.....	6
3.3 Suunnitelmakuvaus.....	7
3.4 Ohjelmiston suunnittelu .....	8
3.5 Toteutus.....	8
3.5.1 Tietokannan toteutus .....	9
3.5.2 Ulkoasun toteutus.....	9
3.5.3 Palvelinpuolen toteutus .....	10
3.5.4 Sovelluksen dokumentointi.....	11
3.6 Produkti .....	12
3.6.1 Julkiset sivut .....	12
3.6.2 Järjestelmänvalvojan sivut .....	12
3.6.3 Tavallisen käyttäjän sivut .....	13
3.7 Yhteenveto .....	13
4 Pohdinta .....	15
4.1 Tulokset ja tulkinnat.....	15
4.2 Eettiset näkökohdat .....	17
4.3 Jatkokehittämishdotukset .....	17
4.4 Opinnäytetyöprosessin ja oman oppimisen arviointi .....	18
Lähteet .....	19
Liitteet.....	21
Liite 1. Karkea malli Bougoudia Ry:n ulkosivun rakenteesta .....	21



Liite 2. Vaatimus- ja suunnittelumäärittely dokumentti .....	22
Liite 3. Tietokanta kaavio .....	26
Liite 4. Tietokannan attribuutit.....	27
Liite 5. Tietokannan indeksit .....	30
Liite 6. Tietokannan luontilauseet .....	31
Liite 7. Bougoudia etusivu.....	35
Liite 8. Bougoudia kummilasten hallintasivu.....	36
Liite 9. Sovelluksen viestintäkaavio.....	37
Liite 10. Kommentointi tavat .....	38
Liite 11. Julkiset sivut.....	39
Liite 13. Käyttäjän sivu .....	46

## Lyhenteet ja termit

Termi	Selite
Annotaatio	On Java koodissa määritettävää meta-dataa, joka määrittää kuinka tämä tietty alue koodista toimii
CSS	Cascading Style Sheet, HTML:n ulkoasun muokkaus kieli
Debug	Vian korjaus
Footer	HTML sivun alin elementti, sisältää yleistä tietoa sivustosta
Framework	Sovelluskehys, jonka avulla kehitetään standardin mukaisia sovelluksia.
HTML	Merkintä kieli, jolla rakennetaan HTML-verkkosivu
Javadoc	Java dokumentti kommentti.
MVC	Model View Controller, Sovelluskehysten ohjelmointi malli
Script (JavaScript, jQuery, AJAX)	Selaimessa toimiva HTML:n manipulointi kieli
SQL	Structured Query Language, kaikki tietokannat käyttäät tätä kieltä (esim. MySQL)
Scrum	Ketterä ohjelmistokehitys menetelmä
Tag	Tagi, notaatio, asiayhteys on yleensä HTML tagi kuten <p></p>

# 1 Johdanto

Bougoudia Ry on hyväntekeväisyys yhdistys, joka tekee ruohonjuuritason hyväntekeväisyyttä Afrikassa Bakaribougoun ja Nasson kylissä. Verkkosivut ovat tarkoitettu yhdistyksen toiminnan keskittämiseksi. Tällä tarkoitetaan keskeisten tietojen ja tiedotteiden löytymistä yhdestä osoitteesta.

Ennen tätä projektia yhdistyksellä oli vain Facebook -tili, Youtube -kanava ja blogi. Materiaalin ja tiedotteiden erillään oleminen on monimutkaistanut tiedon ylläpidettävyyttä, jonka takia verkkosivut päätettiin rakentaa.

## 1.1 Tavoitteet

Verkkosivuston toiminnallisuuden päämääränä oli ajatus, verkkosivuston tulisi olla niin kattava kuin mahdollista. Sivustosta pitäisi pystyä katsomaan kummilapsia, kuvia, tiedotteita, kummikirjeitä ja muuta tietoa yhdistyksestä, lisäksi muutamia tärkeitä asioita, joita sivustolta pitäisi löytyä ovat yhdistyksen säännöt, rekisteröitymislomake ja tavat tukea sivu.

Jokaisen ulkoisensivun sisältöä ohjelmassa pitäisi pystyä muuttamaan, jotta verkkosivusto palvelisi dynaamisuutta. Verkkosivusto tarvitsi ulkoisten sivujen lisäksi hallintapaneelin, josta voidaan muuttaa ulkoistensivujen sisältöä.

## 1.2 Aiheen rajaus

Aihetta ei ollut rajattu selkeällä rajaviivalla. Perus ajatuksena projektiin käytettävissä oleva aika toimi käytännössä rajaviivana. Mitä kerkesi ajanpuitteissa tehdä, oli osa projektia ja mitä ei ollut aikaa tehdä, ei ollut osa projektia.

Yhdistys toivoi verkkosivuilleen suoramaksulomaketta, jonka kautta tukimaksuja ja muita yhdistyksen vuosimaksuja olisi helppo käytännössä maksaa. Tämä rajattiin ulos projektista, koska työhön käytettävä aika oli kuitenkin rajallinen.

### 1.3 Tekniikat ja menetelmät

Grafiikan luomiseen käytettyjä menetelmiä oli Gimp2 kuvankäsittely ohjelma. Sivuston värejä ja sävyjä on myös katsottu sivulta w3schools:n verkkosivujen värivalitsin toiminnon avulla. Ulkoasun luomiseen on käytetty Bougoudia Ry:n omia valokuvia, joita on kuvankäsittely ohjelmalla editoitu sopiviksi verkkosivustolle. (W3schools color picker, 2014).

Toiminnallisuuden parantamiseksi ja hienontamiseksi on käytetty jQueryä, JavaScriptia ja AJAX:ia. JQuery oli pääroolissa sovelluksen ulkoasun manipuloimisessa. JavaScript oli toisessa roolissa ja AJAX vasta kolmannessa roolissa.

Palvelinpuolen ohjelmoimiseen on käytetty Java Spring Frameworkkiä ja Spring Frameworkin muutamia lisäkirjastoja, Spring Security, Spring Context Support, Spring Jdbc, Javax Validation ja Javax Mail.

Sivuston käyttämäksi tietokannaksi valittiin MySQL tietokanta. Pääsyyinä on, että MySQL on koulusta tuttu tietokanta, jonka takia se oli yksinkertaisin valinta tähän projektiin.

Muita käytettyjä työvälineitä olivat Eclipse, jota käytettiin koko ohjelmiston ohjelmoimiseen. Microsoft Office työkalut, joita käytettiin tämän projektin dokumenttien kirjoittamiseen ja ylläpitämiseen. Selaimet kuten Firefox, Chrome ja niiden vian korjaus (debug) työkalut, joilla varmistettiin sovelluksen ulkoasun laadullisuus ja toimivuus.

## 2 Teoriatausta

Toimivan ja modernin verkkosivuston rakentamiseksi tulisi olla tietoa seuraavista kielistä joita tässä raportissa käsitellään. HTML, JavaScript, jQuery, AJAX, Java EE, JSP, JSTL ja SQL. Nämä listatut ohjelmointikielet jakautuvat aika hyvin puoliksi, toiset ovat selainpohjaisia ja toiset ovat palvelinpohjaisia. Selainpohjaiset kielet toimivat selaimessa eivätkä tiedä palvelimesta mitään, kun taas palvelinpohjaiset kielet toimivat palvelimella ja kommunikoivat selainten pyyntöjen kanssa.

### 2.1 Selainpuolen tausta

Verkkosovelluksen ohjelmoimiseen tarvitaan tietoa HTML kielestä. HTML:n lisäksi on erittäin hyvä tietää nykypäivän HTML standardeista, joita ovat muun muassa XHTML ja HTML 5. HTML on merkintä kieli, jonka pohjalle kaikki verkkosivut rakentuvat.

Uskottavan, näyttävän ja käyttäjäystävällisen käyttöliittymän luomiseksi on hyvä tietää JavaScript:sta, jQuery:stä ja tarvittaessa AJAX:sta. JavaScript on selaimessa pyörivä HTML sivun manipulointi kieli, jolla voidaan muokata HTML sivun elementtien ulkoasua ja näkyvyyttä, tietyillä ehdoilla ja komennoilla tai niin sanotuilla tapahtumilla (=englanniksi event). jQuery on JavaScriptin toteuttava kirjasto, joka on kehitetty helpottamaan ja yksinkertaistamaan JavaScriptin käyttämistä verkkosivustolla, esimerkiksi HTML elementtien manipuloiminen on helpompaa jQueryllä. AJAX on JavaScriptiä, joka mahdollistaa tiedon välittämisen palvelimen ja selaimen välillä niin sanottuna taustaprosessina. Tämä tarkoittaa, ettei verkkosivua tarvitse ladata uudestaan tiedon hakemisen tai tallentamisen yhteydessä. (W3schools JavaScript, 2014) (W3schools jQuery, 2014) (W3schools AJAX, 2014).

JSP tulee sanoista JavaServer Pages ja on Java verkkosovellusta varten kehitetty verkkosivu tyyppi, joka rakentuu HTML:stä, mutta erona on, tällä sivulla voidaan suorittaa Java koodia HTML:n seassa. Java koodin suorittaminen HTML koodin seassa teki HTML koodista sekavampaa ja tämän takia luotiin JSTL. (Oracle JavaServer Pages, 2014) (Oracle JavaServer Pages Standard Tag Library, 2014).

JSTL on JSP:n standardi tagi kirjasto. JSTL kirjasto sisältää paljon yleisiä tageja, moneen käyttöön. Tämä kirjasto poistaa tarpeen Java koodille HTML:n seassa, jolloin HTML koodista tulee luettavampaa, kun koodi rivien määrä pienenee. JSTL tagien käyttö on nykyaikaisessa verkkosovelluksessa erittäin suositeltavaa.

## 2.2 Palvelinpuolen tausta

Java EE on nykyään laajalti tunnettu termi jota käytetään palvelinpuolen Java ohjelmointiin viittaavana terminä. Java EE tulee sanasta Java Enterprise Edition. Käytännössä Java EE on palvelinpuolen ohjelmointiin tarkoitettu Javan versio. (Oracle Java EE, 2014).

Spring on Java EE ohjelmointiin kehitetty Framework, niin kutsuttu sovelluskehys malli. Tämä tarkoittaa käytännössä, että kaikki ohjelmoiminen tehdään saman kaavan mukaisesti, mikä helpottaa koodin tutkimista ja ohjelmoimista huomattavasti, kun käytössä on tällainen standardi tapa ohjelmoida. Spring Framework on MVC mallin mukainen, mikä tulee sanoista Model View Controller. Model View Controller toimii seuraavasti. Modelit ovat tieto luokkia joita näytetään verkkosivustolla tai joita tallennetaan verkkosivustolta tietokantaan. View on itse verkkosivu jonne tuodaan näitä Model luokkia. Controller on niin sanottu kommunikoija palvelimen ja selaimen välillä, eli vastaa selaimen kutsuihin ja lähettää palvelimen vastauksen selaimelle View:lle. Controller luokka itse on palvelimella, vaikka toimiikin niin sanottuna siltana selaimen ja palvelimen välillä. (Spring Framework, 2014).

Spring Framework:iin voidaan liittää muita Spring kirjastoja kuten Spring Security, Spring Context-Support, Spring-Jdbc, Javax.Validation ja Javax.Mail. Nämä kirjastot ovat jo valmiiksi tehtyjä ohjelman osia joita sovellukseen liittämällä saadaan sovelluksesta toimivampi ja turvallisempi. Kirjastojen lisääminen on kannattavaa, ettei samoja asioita jotka pitäisi toteuttaa samalla tavalla tarvitse tehdä aina uudestaan alusta, kun samat toiminnot voidaan yksinkertaisesti lisätä projektiin muutamalla komennolla. (Spring Security, 2014) (Spring Context Support, 2014) (Spring jdbc, 2014) (Validation, 2014) (Mail, 2014).

Lyhyesti kerrottuna mitä mikäkin paketti tekee sovelluksessa:

- Spring Security hoitaa käyttäjien oikeustasoa ja kirjautumista sovelluksessa.
- Spring Context Support tarvitaan Javax Mail paketin kanssa hoitamaan Sähköpostin lähetystä sovelluksessa.
- Javax Validation hoitaa tietojen tarkistamista ja todistaa syötetyt tiedot oikeanlaisiksi.
- Spring Jdbc hoitaa tietokanta yhteyttä, tiedon tallentamisvaiheessa.

MySQL on tietokanta sovellus, jota tämä verkkosivusto käyttää tiedon tallentamisessa ja näyttämisessä verkkosivulla. MySQL käyttää SQL kieltä. SQL kielessä ensin luodaan tietokanta. Toiseksi luodaan taulut ja taulujen väliset suhteet toisiinsa. Kolmanneksi luodaan valmis muuttumaton sisältö, jota sovellus tarvitsee toimiakseen. (MySQL, 2014).

### 3 Suunnittelu ja Toteutus

Vaihtoehtoiset tekniikat olivat PHP tai Java. Koulussa pääaineena on opiskeltu Java -ohjelmointia, joten päätin toteuttaa opinnäytetyönä tehtävät verkkosivut Bougoudia Ry:lle käyttäen Javan tekniikoita. Java -ohjelmointi oli mielekkäämpi vaihtoehto, sillä siinä halusin kehittyä ja pääaineeni ansiosta, tiesin Javasta enemmän kuin PHP:sta.

Koulussa saamani kokemuksen perusteella PHP:lla olisi pitänyt ohjelmoida huomattavasti enemmän, kuin käyttäen Javaa saadakseen aikaan yhtä turvallinen ohjelma samoilla ominaisuuksilla. Mielestäni syy tähän on, ettei koulussa ole ollut yhtä paljon PHP -ohjelmointia kuin Javassa, jolloin myös tietämys PHP:sta on suppeampi kuin Javasta. Javassa on jo tutuksi tullut hyvä ominaisuus, että siihen voidaan lisätä jo valmiiksi ohjelmoituja kirjastoja, jotka kattavat standardit toimenpiteet esimerkiksi seuraavissa toimenpiteissä kirjautumisessa, oikeuksien hallinnassa, monimutkaisissa tietokantayhteyksissä ja sähköpostiviestien lähettämisessä.

#### 3.1 Kohde

Opinnäytetyönä toteutin Bougoudia Ry:lle verkkosivut, joilla voidaan hallita Bougoudia Ry:n tiedotteita, kummilapsia, kummikirjeitä, kuvia, yhteystietoja, yhdistyksen jäseniä, rekisteröityneitä jäseniä, maksu tapoja, tuki tapoja ja muita tietoja koskien kummilapsia ja yhdistyksen toimintaa Afrikassa.

#### 3.2 Kehityksen haasteet

Tämän sovelluksen myötä olen oppinut paljon uusia asioita ohjelmoinnista. Ennen sovelluksen ohjelmoimisen aloittamista, suunnitteluvaiheessa, minulla oli jo paljon tietoa ja ajatuksia mitä tulen tekemään ja millä tavalla. Sovelluskehitys ei kuitenkaan sujunut suunnitelmien mukaisesti, vastaan tuli yllättäviäkin ongelmia. Ongelmia ilmeni sovellusalueen kaikilla puolilla selainpuolella, palvelinpuolella ja tietokannassa. Kerran tai kaksi tein tietokantaan ison muutaman taulun muutoksen, johtuen ettei suunnittelu ollut tarpeeksi pitävä. Toisaalta kehityksen edetessä huomasin ettei ollut edes osannut ottaa tiettyjä asioita huomioon. Ongelmat eivät ehkä johtuneet suunnittelemattomuudesta vaan



kokemattomuudesta. Tämä verkkosivusto tarjosi haasteita, mutta niiden kautta kehitystä tapahtui huomattavasti sovelluskehittäjänä.

Selainpuolella ongelmat eivät olleet niin yllättäviä kuin ehkä tietokannassa. Selainpuolen ongelma kohdat olivat melkein pä aina jo edeltäpäin tiedettävissä, mutta ratkaisu puuttui. Näiden ongelmien ratkaiseminen yleensä vaati miettimistä, kokeilemista ja tausta selvittämistä Internet oppaista.

Palvelinpuolen ongelmakohdat olivat harvinaisia. Nämäkin ongelmat olivat jo edeltä tiedettäviä, joihin piti etsiä ratkaisut Internetistä. Kaksi suurinta palvelinpuolen ongelmaa olivat kuvien tallentaminen palvelimelle verkkosivulta ja AJAX kutsun vastaus.

Sovelluskehityksessä kaikki ilmenneet ongelmat tulivat ratkaistuksi, joidenkin ratkaisu kesti useamman päivän, kun taas joidenkin useamman tunnin. Kaikki kehitykselle asetetut tavoitteet saavutettiin, mutta jatko- ja lisäkehittävää jäi jäljelle.

### **3.3 Suunnitelmakuvaus**

Opinnäytetyö on edennyt alkuperäisen suunnitelman mukaisesti muissa osa-alueissa, mutta neljänkuukauden aikaraja muuttunut viideksi kuukaudeksi jo aloituskokouksessa. Aloituksen ja viivästyminen viivästytti aikataulua viiteen kuukauteen.

Projekti alkoi tapaamisella asiakkaan kanssa, jossa teimme yhdessä muistiinpanot vaatimuksista ja toivotuista ominaisuuksista. Seuraavana oli projektin aloituskokous jossa kävimme jo projektin luonteesta tiedettävät asiat läpi ja hyväksyimme projektin aloitettavaksi.

Projektin käynnistyksen jälkeen aloituskokouksessa ensimmäinen suunniteltu vaihe on Ohjelmiston suunnittelu. Ohjelmiston suunnittelulle määritetty aika on yksi kuukausi. Ohjelmiston suunnitteluun kuuluu tietokannan suunnittelu ja ulkoasun suunnittelu.

Seuraava vaihe on ohjelmiston toteutus. Ohjelmiston toteutukseen määrätty aika on kolme ja puoli kuukautta. Ohjelmiston toteutukseen kuuluu tietokannan toteutus, Ulkoasun toteutus, Ohjelmiston palvelinpuolen toteutus ja dokumentointi.

Viimeisenä vaiheena on projektin viimeistely. Viimeiselle vaiheelle asetettu aika on kuukausi. Projektin viimeistelyyn kuuluu sovelluksen ohjelmoinnin viimeistely, dokumenttien viimeistely ja opinnäytetyön päättökokous.

### **3.4 Ohjelmiston suunnittelu**

Sovelluskehityksen ensimmäisessä vaiheessa suunnittelin ensimmäisen asteen tietokannan. Tietokanta päivittyi ja muuttui koko sovelluskehityksen ajan, koska sovellukseen tai verkkosivulle tehtävät muutokset piti useimmiten tehdä tietokasta lähtien alusta. Liitteenä on suunnitellun ja toteutetun tietokannan viimeisin versio. (Liite 3. Tietokanta kaavio)

Seuraavaksi suunnittelin Bougoudia Ry:n verkkosivuston karkean mallin julkisivun rakenteesta. (Liite 1. Karkea malli Bougoudia Ry:n ulkosivun rakenteesta) Verkkosovelluksen kaikki julkiset sivut toteutettiin liitetyn mallin mukaisesti.

Ennen sovelluskehityksen aloitusta, kirjoitin Bougoudia Ry:n vaatimuksista, toivotuista ominaisuuksista ja ulkoasu toiveista dokumentin, joka pitää sisällään kaikki ulkoasun perus ajatukset joita kehityksen aikana lähdin viemään eteenpäin. (Liite 2. Vaatimus- ja suunnittelumäärittely dokumentti)

### **3.5 Toteutus**

Tässä kappaleessa syvennyttään verkkosovelluksen eri kehitysvaiheisiin ja siihen kuinka kehitys on missäkin vaiheessa edennyt. Sovelluskehitystä on pyritty tekemään koulusta opittujen ja tuttujen metodien avulla. Scrum -menetelmää on pyritty käyttämään sovelluksen kehityksessä.

### 3.5.1 Tietokannan toteutus

Ensimmäisenä verkkosovellukselle piti luoda tietokanta, jota sovellus käyttää. Tietokanta ei tullut koskaan yhdellä kerralla valmiiksi, vaan tietokantaa hiottiin monta kertaa jotta siitä tuli sovellukseen sopiva. Tietokannan viimeisin versio luotiin liitetyillä tietokanta lauseilla. (Liite 6. Tietokannan luontilauseet)

Tietokanta on indeksoitu, jotta tiedon haku olisi nopeampaa ja tehokkaampaa. Tietokanta indeksit tehtiin aina muiden tietokanta muutoksien mukana. Tietokannan indeksien luonti lauseet löytyvät tietokannan luontilauseista. (Liite 6. Tietokannan luontilauseet) Indeksit on myös merkitty erilliseen tietokannan indeksit liitteeseen. (Liite 5. Tietokannan indeksit)

Tietokannan attribuuteille on luotu eheyssäännöt. Eheyssääntöjen luominen on tietokannan kannalta tärkeää, jotta sovellus toimisi eheästi. Tämä tarkoittaa, ettei vahingossa voida jostain sovelluksesta tallentaa tietokannan johonkin kenttään sellaista tietoa mitä sinne ei saisi tallentaa. Tämän tilanteen sattuessa joko tietokanta hajoaisi tai sovellus hajoaisi. Syy tähän on, ettei sovellus tai tietokanta osaisi toimia oikein, mikäli väärän tyyppistä tietoa yritetään käsitellä sovelluksessa tai tietokannassa. Tietokannan tauluille määritetyt eheyssäännöt on nähtävillä tietokannan attribuutit liitteessä. (Liite 4. Tietokannan attribuutit)

### 3.5.2 Ulkoasun toteutus

Ulkoasun toteutus oli suunnitelmasta huolimatta ehkä toiseksi eniten aikaa vaativin osa-alue sovelluskehityksestä. Ulkoasun toteutus sisälsi paljon scriptien kirjoittamista, kuvien käsittelyä, css tyylimäärittelyjen kirjoittamista, html sivun rakentamista. Julkisten verkkosivujen rakenteen ja perusulkoasun toteuttaminen valmistui suunnitelman mukaisesti ja suunnitellussa aikajaksossa. Seuraavaksi aloitin järjestelmänvalvojan hallintasivuston suunnittelun ja toteuttamisen.

Järjestelmänvalvojan hallintasivuston ulkoasua ei ollut projektin suunnittelu vaiheessa vielä suunniteltu, joten hallintasivuston ulkoasun kehitys meni vierekkäin hallintasivuston palvelinpuolen ohjelmoinnin kanssa. Hallintasivuston kehityksen aikana oli elintär-

keää pitää mielessä mitä toimintoja ja ominaisuuksia sivustolla pitäisi olla. Koska hallintasivuston kehitys ja suunnittelu tapahtuivat samanaikaisesti, jouduttiin välillä tekemään isojaakin muutoksia sovellukseen, tietokantaan ja niiden toimintaan. Hallintasivujen ulkoasusta oli käytännössä kolme isoa versiota ja muut päivitykset olivat pieniä satunnaisia päivityksiä. Hallintasivuston viimeisin ulkoasupäivitys on nähtävillä liitteestä 8. (Liite 8. Bougoudia kummilasten hallintasivu)

Hallintasivuston valmistuttua, siirryin takaisin julkisten verkkosivujen ulkoasun suunnitteluun ja kehitykseen. Ulkoasu suunnitelmassa ei mietitty tarkasti miten sivuston sisältö näytetään sivuilla, joten tässä vaiheessa ensin suunnittelin kuinka sisältö näytetään ja sitten toteutin suunnitelman mukaisen ratkaisun. Sisällön oikein näyttämisen toteuttamiseksi jQuery, JavaScript ja JSTL olivat olennaiset osat julkisten verkkosivujen ulkoasun kehityksen viimeistelyssä. Viimeisin versio Bougoudia Ry:n julkisten verkkosivujen ulkoasusta löytyy liitteestä 7. (Liite 7. Bougoudia etusivu)

### **3.5.3 Palvelinpuolen toteutus**

Palvelintoteutus alkoi heti julkisten sivujen perus ulkoasun ja rakenteen valmistuttua. Palvelinpuolen toteuttaminen vaati ehdottomasti eniten aikaa koko projektista. Muutamissa luokissa koodirivejä tuli kirjoitettua yli 1000. Palvelinpuolella on toteutettu kolme kontrolleria AdminController, MainController ja UserController. Kontrollerit vastaanottavat selaimelta tulevia kutsuja ja lähettävät niihin vastauksen. Vastaus tässä sovelluksessa yleensä on JSP -sivu jonne tulostetaan palvelimen lähettämiä tietoja. Näillä kolmella kontrollerilla on niin sanotusti tietyt hierarkiatasot jotka nimikin kertoo. UserController käsittelee tavallisen käyttäjän toimintoja sovelluksessa. MainController käsittelee kaikkia julkisia toimintoja sovelluksessa. AdminController käsittelee kaikkia järjestelmänvalvojan toimintoja sovelluksessa. UserController voi kutsua vain UserService luokkaa ja UserService voi kutsua vain UserDao luokkaa. User tason luokat eivät voi kutsua ylemmän tason luokkia. MainController voi kutsua vain MainService luokkaa ja MainService voi kutsua MainDao luokkaa ja MailService luokkaa. AdminController voi kutsua AdminService luokkaa ja MainService luokkaa. AdminService voi kutsua AdminDao luokkaa ja MainDao luokkaa. Service luokat ovat toiminnallisuus luokkia, joissa tehdään kaikki tarvittava logiikka sovelluksessa. MailService luokassa käsitel-

lään sovelluksen sähköpostien lähettäminen. Dao luokat käsittelevät tietokanta kyselyitä, hakuja ja tiedon päivittämistä ja lisäämistä. Lisäksi kaikki kontrolleri luokat kutsuvat CustomValidator luokkaa jossa validoidaan kaikki tieto jota sovelluksessa käsitellään. Palvelinpuolen toiminnan ja koko sovelluksen toiminnan voimme nähdä graafisesti sovelluksen viestintäkaaviossa. (Liite 9. Sovelluksen viestintäkaavio)

Kontrollereissa on kahdenlaisia metodeita, jotka vastaavat selainten kutsuihin. Toiset metodit palauttavat JSP -sivun ja toiset palauttavat tekstiä. Metodin yläpuolella oleva @ResponseBody annotaatio kertoo metodin palauttavan JSP -sivulle body elementin sisälle tekstiä (Kuva 1) jos tätä annotaatioita ei ole määritetty metodin yläpuolella metodi palauttaa selaimelle JSP -sivun.

```
@ResponseBody
@RequestMapping("/getchild/{cid}")
public String getChildDetails(Model model, @PathVariable("cid") int cid, HttpServletResponse response) {
```

Kuva 1. Tekstin palautus verkkosivulle

Sovelluksen kokoonpano asetuksissa on sovellukselle määritetty sovelluksen ulkopuolinen resurssikansio kuville, jotka sovelluksen kautta lisätään sovellukseen (Kuva 2). Tämä on tärkeä asetus, jos halutaan sovelluksen pystyvän käyttämään ulkopuolista polkua kuvien tallentamiseen ja hakemiseen.

```
<mvc:resources location="file:/usr/share/tomcat/webapps/uploads/" mapping="/images/*"/>
```

Kuva 2. Sovelluksen ulkopuolisten resurssien määrittäminen

### 3.5.4 Sovelluksen dokumentointi

Sovellus on dokumentoitu kommenteilla. Ensimmäinen kommentointi tapa on kommentoida Javadoc kommenteilla, jotka näyttävät koodi editorissa sinisiltä. Toinen tapa on kommentoida rivikommenteilla tai tavallisilla kappalekommenteilla, jotka näyttävät koodi editorissa vihreältä. (Liite 10. Kommentointi tavat)

Sovelluksen kehitystä on dokumentoitu ulkopuolisilla dokumenteilla, jotka ovat tämän raportin liitteinä. (Liite 1. Karkeamalli Bougoudia Ry:n ulkosivun rakenteesta) (Liite 2. Vaatimus- ja suunnittelumäärittely dokumentti) (Liite 3. Tietokanta kaavio) (Liite 4.

Tietokannan attribuutit) (Liite 5. Tietokannan indeksit) (Liite 6. Tietokannan luontilauseet)

### **3.6 Produkti**

Tämä verkkosovellus koostuu kolmesta eri osa-alueesta, julkiset verkkosivut, järjestelmänvalvojan verkkosivut ja tavallisten käyttäjien verkkosivut. Julkisten verkkosivujen ulkoasu on aina sama, vaikka sisältö vaihtuu. Sisällön vaihtuessa myös sisällön näyttäminen muuttuu. Järjestelmänvalvojan verkkosivuilla hallitaan kaikkea julkisilla sivuilla näytettävää sisältöä ja yksittäisten käyttäjien sivuilla, käyttäjät voivat hallita vain omia tietojaan.

#### **3.6.1 Julkiset sivut**

Julkisten sivujen ylä laidassa on aina logo, vasemmassa laidassa on aina tärkeät linkit kuvagalleria, tarinagalleria, kummikirjeet ja kummikansio ja yhteistyökumppanit listattuna. Yhteistyökumppaneiden logo näytetään, jos on logo tallennettu, muutoin yhteistyökumppaneiden nimi näytetään. Sivuston alalaidassa on aina niin kutsuttu footer, josta löytyy kaikki julkisten sivujen linkit jäseneltynä allekkain. Footer osiossa on lisäksi linkki viittaukset Bougoudia Ry:n ulkopuolisiin palveluihin kuten Facebook, Youtubet ja blogi. (Liite 7. Bougoudia etusivu)

Julkisilta sivuilla voi katsoa kummilapsia, katsoa kuvia, lukea kummikirjeitä, lukea tarinoita, lukea yhteistyökumppanikouluista, lukea tapahtuvista projekteista liittyen Bougoudia Ry:n toimintaan Afrikassa, lukea ympäristö kasvatuksesta, lukea ajankohtaisia ilmoituksia, rekisteröityä jäseneksi ja kummiksi, katsoa tapoja tukea, katsoa tietoa eettisistä lahjakorteista, katsoa Bougoudia Ry:n yhdistyksen yhteystiedot, katsoa Bougoudia Ry:n hallituksen jäsenet, lukea miten kaikki alkoi ja lukea yhdistyksen säännöt. Muutamia mallikuvia julkisista sivuista löytyy liitteenä. (Liite 11. Julkiset sivut)

#### **3.6.2 Järjestelmänvalvojan sivut**

Kirjaututtua sovellukseen tunnuksilla, joilla on järjestelmänvalvojan oikeudet, sovellus vie käyttäjän automaattisesti järjestelmänvalvonta sivuille. Järjestelmänvalvonta sivuilla

on aina selaimen ylälaudassa valikko, josta valitaan hallittava osa-alue ja voidaan kirjautua ulos. Järjestelmänvalvontasivu on keskitetty keskelle ruutua. Taustalla on kuva Afrikan maisemasta ja keskellä on läpinäkyvä harmaa alue josta tapahtuu kaikki hallinta.

Järjestelmänvalvonta sivuilla järjestelmänvalvoja voi päivittää omia tietojaan, vaihtaa oman salasanan, päivittää yhdistyksentietoja, luoda uusia yhdistyksenjäseniä, selata yhdistyksenjäseniä, päivittää yhdistyksenjäsenien tietoja, lisätä yhdistyksenjäsenelle kummilapsia, poistaa yhdistyksenjäseniltä kummilapsia, poistaa yhdistyksenjäsenen, luoda yhteistyökumppanin, poistaa yhteistyökumppanin, päivittää yhteistyökumppanin tietoja, luoda kummilapsen, poistaa kummilapsen, päivittää kummilapsen tietoja, hakea kummilapsia, luoda kylän, poistaa kylän, muokata kylän nimeä, lisätä kuvia julkisille sivuille, poistaa kuvia julkisilta sivuilta, julkaista tekstiä julkisilla sivuilla, muokata julkaistua tekstiä julkisilla sivuilla, poistaa julkaisu, selata rekisteröityneitä käyttäjiä, päivittää rekisteröityneiden käyttäjien tietoja, lisätä kummilapsia rekisteröityneille käyttäjille, poistaa kummilapsia rekisteröityneiltä käyttäjiltä, poistaa rekisteröityneen käyttäjän. (Liite 12. Järjestelmänvalvojan sivut)

### **3.6.3 Tavallisen käyttäjän sivut**

Tavallisen käyttäjän sivut ovat ulkoasultaan samanlaiset kuin järjestelmänvalvojan sivut, mutta tavallisen käyttäjän sivulla ei ole kaikkia järjestelmänvalvojan toimintoja käytössä. Tavallisen käyttäjän sivuilta voi käyttäjä vain päivittää omia tietojaan ja vaihtaa oman salasanan ja kirjautua ulos omilta sivuilta. (Liite 13. Käyttäjän sivu)

## **3.7 Yhteenveto**

Tämä opinnäytetyö on Java ohjelmointikielellä toteutettu verkkosovellus Bougoudia Ry hyväntekeväisyys yhdistykselle. Valitsin tämän opinnäytetyön aiheen, koska halusin kehittyä Java ohjelmoinnissa, jota HAAGA-HELIA ammattikorkeakoulussa on opetettu.

Projekti alkoi tapaamisella asiakkaan kanssa, jossa kävimme läpi asiakkaan näkemyksiä toiveita ja vaatimuksia sovelluksen suhteen. Seuraavaksi järjestettiin koulussa projektin aloituskokous jossa virallistettiin opinnäytetyön tekeminen voi alkaa.

Ennen suunnittelu vaihetta kirjoitin tästä asiakkaan kanssa laaditusta suunnitelmasta, virallisen suunnittelu- ja vaatimusmäärittämis dokumentin.

Suunnittelu vaiheessa ensimmäisenä suunnittelin sovelluksen tietokannan ensimmäisen version ja kirjoitin suunnitelmasta tietokanta dokumentin. Tämän jälkeen suunnittelin sovelluksen julkisten sivujen ulkoasun.

Suunnittelu vaiheen jälkeen aloitin tietokannan luomisen, jota seurasi sovelluksen julkisten verkkosivujen ulkoasun toteutus. Julkisen ulkoasun toteutuksessa ilmeni haasteita joihin en ollut ennen törmännyt, mutta niistä selvittiin Internetistä tutkimien ohjeiden avulla. Julkisten verkkosivujen ulkoasun valmistuttua aloitin järjestelmänvalvojan hallintasivujen toteuttamisen. Hallintasivujen ulkoasua ei ollut edeltäpäin suunniteltu, vaan sivut kehittyivät toiminnallisuuden kehittyessä.

Hallintasivuston valmistuttua viimeistelin julkistensivujen ulkoasun. Sovelluksen lähes valmistuttua kommentoin koodin mielestäni tarvittavalla tasolla. Viimeiseksi päivitin jo tehdyt dokumentit vastaamaan tämänhetkistä sovelluksen tilannetta ja loin tarvittavia lisä dokumentteja.



## 4 Pohdinta

Bougoudia Ry on hyväntekeväisyys yhdistys, joka tekee hyväntekeväisyystyötä Afrikassa. Yhdistyksen pääajatuksena on etenkin auttaa naisten asemaa opiskelussa. Yhdistys tekee kummilapsi toimintaa Afrikan lasten kanssa Bakari Bougoun ja Nasson kylissä. Oppinnäytetyön tekeminen aloitettiin yhdistyksen verkkosivusto tarpeen täyttämiseksi.

### 4.1 Tulokset ja tulkinnat

HTML 5 ja CSS 3:n ansiosta verkkosivustosta tuli näyttävän näköinen. Mielestäni verkkosivuston ulkoasu, julkiset sivut ja hallintasivut näyttävät oikein hyviltä. Verkkosivujen teema onnistui hyvin, johtuen hyvästä suunnitelmasta ennen toteuttamista. Ennen projektin virallista aloittamista oli erittäin hyvä tavata asiakkaan kanssa jolloin selkisi asiakkaan mielipiteet ja toiveet verkkosivustosta ja ulkoasusta. Tapaamisessa päätettiin sivuston ulkoasun päävärit, jotka auttoivat huomattavasti verkkosivuston teeman tekemistä.

JavaScript, jQuery ja AJAX olivat tärkeässä roolissa verkkosivun käyttäjäystävällisyyden toteuttamisessa. Näiden selaimessa toimivien scriptien ansiosta verkkosivustosta saatiin modernin tyylikäs ja nykyajan käyttäjäystävällinen sivusto. Ilman JavaScriptia ja jQueryä ei verkkosivustosta olisi voinut saada näinkään käyttäjäystävällistä kuin se nyt on.

(W3schools jQuery) AJAX oli avain asia kummilasten lisäämisessä jäsenille. Sillä AJAX mahdollistaa tiedon hakemisen tietokannasta ilman että sivua ladataan uudestaan.

(W3schools AJAX) Kummilapsen valinnassa tulos oli juuri mitä halusinkin, aina kummilapsen valittua AJAX hakee tietokannasta automaattisesti tämän kummilapsen tiedot ja tulostaa verkkosivulle ilmansivun uudelleen lataamista. Verkkosivuston kaikki osat alueet onnistuivat haluamallani tavalla JavaScriptin ja jQueryn ansiosta.

JSP ja JSTL tekniikan ansiosta verkkosivulle saatiin tulostettua palvelimenvastaukset halutulla tavalla. JSTL on JSP sivujen standardi tagi kirjasto jonka avulla voidaan tulostaa palvelimen vastauksia JSP sivuille turvallisesti. (Oracle JavaServer Pages) (Oracle JavaServer Pages Standard Tag Library) Nykyään tämä on tullut uusia ja luultavasti parempia tapoja, joilla voidaan korvata JSP ja JSTL. Näitä uusia tapoja ovat muun muassa JSON tekniikka, jonka avulla Java oliot voidaan kääntää tekstiksi ja viedä selaimelle ja

selaimella parsia teksti järkevän näköiseksi ja tulostaa se käyttäen jQueryä tai muuta JavaScript kirjastoa.

Java palvelinpuolen koodi toimii niin kuin odotettiin. Spring Frameworkin avulla toteutettu sovellus toimii juuri kuin odotettiin. Sovellus on toteutettu koulussa opitun Spring Framework MVC mallin mukaisesti, jossa sovelluksessa on tietokantaluokka taso, palveluluokka taso, kontrolleriluokka taso, erilaisia tietoluokkia joita palautetaan verkkosivulle.

Spring Security huolehtii tässä sovelluksessa kirjautumisesta ja oikeustasoista. Tämä on Spring Frameworkin valmis kirjasto ja toimii odotetusti. Ilman tätä osiota olisi mahdollisesti pitänyt ottaa paljon asioita huomioon ja todennäköisesti ohjelmoida uudestaan jo valmiiksi ohjelmoituja asioita, liittyen kirjautumiseen ja oikeustasojen hallintaan. Mikäli haluttaisiin sovelluksen olevan perus turvallinen käyttää.

Sovelluksessa tietokanta logiikka ja yhteys on toteutettu Spring jdbc tekniikalla. Tämä tekniikka on jo kohtuullisen vanhaa ja sitä ilmeisesti harvoin käytetään, johtuen sen monimutkaisuudesta verrattuna uudempiin tekniikoihin. Todennäköisesti uudempi tekniikka Spring hibernate ja jpa tekniikka olisi ollut paljon parempi vaihtoehto, koodirivien vähentämiseksi ja tietokannan paremman käytön varmistamiseksi. Tätä jpa tekniikkaa ei kuitenkaan ole koulussa niin paljon käytetty, joten koin turvallisemmaksi vaihtoehdoksi käyttää tekniikkaa josta on enemmän kokemusta.

Tietokannaksi valittiin MySQL tietokanta. MySQL on ilmainen tietokanta, joka vaikutti valintaan. MySQL tietokanta toimi kohtuullisen hyvin tämän sovelluksen kanssa. Luultavasti olisi parempiakin tietokantoja, joissa olisi paremmat toiminnallisuudet isoille sovelluksille. Tämä verkkosovellus kuitenkin on sen verran pieni, että MySQL tietokanta välttää hyvinkin. Muutamia pieniä yllättäviä juttuja tuli vastaan tämän tietokannan käytöstä, joihin en ollut aikaisemmin törmännyt. Nämä yllättävät asiat eivät kuitenkaan haitanneet tämän sovelluksen toimintaa. MySQL on koulussa paljon käytetty, joka varmisti valintani MySQL:n käytöstä.

## 4.2 Eettiset näkökohdat

Ulkoasultaan sovellus on mielestäni oikein luotettava pieni verkkosivusto. Enempää tai parempaa tämän opinnäytteen ajanpuitteissa ei olisi pystynyt tekemään. Mikäli kokemusta olisi ollut enemmän, ehkä olisi osannut paremmin suunnitella alusta lähtien sovelluksen toiminnan. Suunnittelusta huolimatta sovelluksesta tuli ihan toimiva. Sovelluksen koko toteutus puoli oli mielestäni erittäin vahva, ehkä johtuen vähemmästä suunnittelusta. Suunnitelmassa oli, ettei sovelluksen sisälle tulisi kiinteitä arvoja, mutta ajan puutteen takia joitain arvoja joutui asettamaan kiinteiksi arvoiksi. Tämän takia sovellus ei ehkä ole täydellisin ratkaisu, mutta toimiva kuitenkin.

## 4.3 Jatkokehittämisehdotukset

Heti ensimetreillä tämän opinnäytteen ulkopuolelle rajatut asiat ovat automaattisesti erittäin vartenotettavia jatkokehitysehdotuksia. Asiakkaan pyytämä suoramaksulomake rajattiin pois, koska tiesin, ettei aika tule riittämään erittäin huolellisesti tehtäviin maksuoperaatioihin. Olisi pitänyt tutkia onko mahdollista liittää pankin toimintaa jollain tavalla ilmaiseksi sovellukseen, vai joutuuko ostamaan jonkun yrityksen palvelun tällaiseen suoramaksun suorittamiseen tämän sovelluksen verkkosivulta. Tämä maksu operaation suorittaminen sovelluksen verkkosivulta vaikuttaa isolta kokonaisuudelta, jonka vuoksi sitä ei pystytty ottamaan mukaan tähän muutenkin jo isoon verkkosivustoon.

Verkkosivusto lähettää rekisteröityneille asiakkaille sähköpostia, joka on jo etukäteen määritelty. Ajanpuutteen takia sovellukseen ei ollut mahdollista tehdä toimintoa jonka avulla voitaisiin muokata lähetettäviä sähköposteja. Tämä toiminnallisuus olisi mukava lisä sovellukseen, ettei sähköpostin muuttamiseen tarvitsisi mennä sovelluksen tekstitiedostoja muuttamaan.

Samoin ajanpuutteen tähden muutamia kiinteitä arvoja joutui jättämään sovellukseen. Mukava lisätoiminnallisuus olisi voida sovelluksen hallintasivujen kautta tai jollain muulla tavalla muuttaa näitä vakioarvoja joita käyttää muun muassa kuvien tallentaminen verkkosivulta. Tämän toteuttaminen poistaisi tarpeen mennä sovelluksen tiedostoja muuttamaan aina jos halutaan vaihtaa vakio arvoja. Tähän liittyy kysymys, että kuinka tällainen on mahdollista toteuttaa. Tätä kysymystä pitäisi tutkia jotta vastaus löytyisi.

#### 4.4 Opinnäytetyöprosessin ja oman oppimisen arviointi

Opinnäytetyöprosessi on hieno prosessi oppiakseen lisää ja vahvistaakseen osaamistaan siinä mitä osaa ja on oppinut aikaisemmin. Aloitusvaiheessa koko opinnäytetyöprosessi vaikutti hankalalta ja pitkältä. Välillä ihmetytti, että mitä varten pitää kirjoittaa tällaisia ihmeellisiä dokumentteja kuten projektisuunnitelma. Opinnäytetyön loppuvaiheessa ymmärsin kuinka helpottavaa oli katsoa projektisuunnitelmasta viitteitä opinnäytetyön raportin kirjoittamiseen ja huomasin kuinka tärkeä tämäkin turhalta tuntunut dokumentti sittenkin oli.

Opinnäytetyöni on produktiivinen, jos olisin tehnyt tutkimustyyppisen opinnäytetyön, en millään olisi kehittynyt siinä mitä olin koulussa opiskellut, enkä olisi oppinut uusia asioita. Tämän takia valitsin produktiivisen opinnäytetyön, jotta voisin kehittyä osaamisessani ja oppia jotain lisää.

Koko opinnäytetyöprosessi oli todella opettavainen. Tämä opinnäytetyö on minun isoin ja laajin työ jonka olen tehnyt. Suunnitteluvaiheessa kaikkea ei kokemattomuuden takia osannut ottaa huomioon ja tästä johtuen jouduin kehittämisen ohella tekemään isojakin muutoksia sovellukseen tietokannasta lähtien. Näistäkin virheistä oppii hyvin ja kokemuksen karttuessa vastaisuudessa osaa ottaa enemmän pieniä asioita huomioon, kun ei tarvitse keskittyä itsestäänselvyksiin.

Tämän opinnäytetyön aikana en ole vain oppinut ohjelmoinnista, mutta myös kokonaisen projektin erivaiheista ja kuinka kirjoitetaan pitkiä raportteja. Mielestäni minulla on vielä paljon opittavaa tieteellisessä kirjoittamisessa ja projektien eri vaiheista. Moniin asioihin olen oikeastaan kunnolla törmännyt vasta opinnäytetyön aikana. Lopuksi sanoisin, että vaikka opinnäytetyö on ehkä pelottava, mutta muuten tosi hyvä oppimiskokemus jos sen näin itselle suo.

## Lähteet

Mail. Luettavissa: <http://www.oracle.com/technetwork/java/javamail/index.html>.

Luettu: 31.03.2014.

MySQL. Luettavissa: <http://dev.mysql.com/doc/refman/4.1/en/what-is-mysql.html>.

Luettu: 31.03.2014.

Oracle Java EE. Luettavissa:

<http://www.oracle.com/technetwork/java/javaee/overview/index.html>. Luettu:

28.03.2014

Oracle JavaServer Pages. Luettavissa:

<http://www.oracle.com/technetwork/java/javaee/jsp/index.html>. Luettu: 28.03.2014.

Oracle JavaServer Pages Standard Tag Library. Luettavissa:

<http://www.oracle.com/technetwork/java/jstl-137486.html>. Luettu: 28.03.2014.

Spring Context Support. Luettavissa

[http://docs.spring.io/spring/docs/3.2.6.RELEASE/javadoc-](http://docs.spring.io/spring/docs/3.2.6.RELEASE/javadoc-api/org/springframework/context/support/package-summary.html)

[api/org/springframework/context/support/package-summary.html](http://docs.spring.io/spring/docs/3.2.6.RELEASE/javadoc-api/org/springframework/context/support/package-summary.html). Luettu:

31.03.2014.

Spring Framework. Luettavissa: <http://projects.spring.io/spring-framework/>. Luettu:

29.03.2014.

Spring jdbc. Luettavissa :<http://docs.spring.io/spring/docs/3.0.x/spring-framework-reference/html/jdbc.html>. Luettu: 31.03.2014.

Spring Security. Luettavissa: <http://projects.spring.io/spring-security/>. Luettu:

29.03.2014.

Validation. Luettavissa:

<http://docs.oracle.com/javase/7/api/javax/validation/package-summary.html>. Luettu: 31.03.2014.

W3schools color picker (W3schools:n värinvalitsin) Luettavissa:

[http://www.w3schools.com/tags/ref\\_colorpicker.asp](http://www.w3schools.com/tags/ref_colorpicker.asp). Luettu: 10.01.2014.

W3schools jQuery. Luettavissa: <http://www.w3schools.com/jquery/default.asp>. Luettu: 02.02.2014.

W3schools JavaScript. Luettavissa: <http://www.w3schools.com/js/default.asp>. Luettu: 02.02.2014.

W3schools AJAX. Luettavissa: <http://www.w3schools.com/ajax/default.asp>. Luettu: 02.02.2014.

## Liitteet

Liite 1. Karkea malli Bougoudia Ry:n ulkosivun rakenteesta

Bougoudia Ry						
Kuvagalleria	Etusivu	Toiminta	Tule tukijoukkoihin	Yhteystiedot		
Tarinagalleria	Alilinkki1	Alilinkki2	Alilinkki3	Alilinkki4		
Kummikirjeet						
Kummikansio						
Yhteistyö kumppani Logo1						
Yhteistyö kumppani Logo2						
Etusivu	Toiminta	Tule tukijoukkoihin	Yhteystiedot	Facebook	Youtube	Blogi
Alilinkki	Alilinkki	Alilinkki	Alilinkki			
Alilinkki	Alilinkki	Alilinkki	Alilinkki			
Alilinkki	Alilinkki	Copyright Bougoudia Ry				

## Liite 2. Vaatimus- ja suunnittelumäärittely dokumentti

### Bougoudia Ry:n verkkosivuston vaatimus- ja suunnittelumäärittely

Sivuston vaatimuksena on toimia tiedon välittämiseen tarkoitettuna alustana. Sivustolta rekisteröityneet jäsenet ja kummit saavat tietoa tämän hetkisistä tapahtumista.

#### Sivuston rakenne

Sivun yläosassa on ensimmäisenä logo alue, jonka alla on sivuston päälinkit vaakasuorassa linjassa. Päälinkkien alla sivusto on jaettu kahteen osioon, vasen laita ja sivuston pääasiallinen sisältö alue. Sivun viimeinen lohko on niin kutsuttu footer osio.

Logo alueeseen laitetaan logo.

Päälinkkivalikko sisältää kaikki päälinkit, joita klikkaamalla aukeaa niiden alle linkkien alilinkki laatikko.

Vasemman puoleinen kapea alue sisältää kaksi osiota. Ylemmänä sijaitsevat toisen prioriteetin linkit. Tämän alapuolella on laatikko joka sisältää yhteistyökumppanien logot.

Oikeanpuoleinen sivuston pääasiallisen sisällön alue sisältää kaiken vaihtuvan ja muuttuvan tiedon, jota järjestelmä käsittelee ja näyttää sivuston käyttäjille.

Footer osio sisältää sivusto kartan, josta löytyvät ryhmiteltyinä kaikki sivuston linkit, lisäksi linkit yhdistyksen muihin julkisiin palveluihin.



## Sivuston mitat ja värit

Sivuston leveys on 1000px leveä ja korkeus riippuu sisällön paljoudesta.

Väri teema on vihreä, oranssi, keltainen.

## Etusivu

Etusivu linkki vie käyttäjän etusivulle, jolla näytetään ajankohtaisia artikkeleita noin 3-5 kpl kerrallaan. Artikkelit vaihtuu aika ajoin. Artikkeleista voidaan painaa lue lisää, joka vie käyttäjän artikkelin omalle sivulle.

## Toiminta

Toiminta linkkiä klikkaamalla päälinkkien alle aukeaa toiminnan lisälinkit, joista päästään asianomaisille sivuille. Toiminta valikon alla linkit ovat: koulut, projektit, ympäristökasvatus, ajankohtaista.

Koulut sivulta löytyy yhteistyökoulut.

Projektit sivulta löytyy käynnissä olevat projektit.

Ympäristökasvatus sivulta löytyy Fason alueen ympäristökasvatukseen liittyvää materiaalia.

Ajankohtaista sivulta löytyvät kaikki dokumentit joista muutama uusin näytetään etusivun ajankohtaista laatikossa.

## Tule tukijoukkoihin

Tule tukijoukkoihin linkkiä päälinkkien alle aukeaa tule tukijoukkoihin lisälinkit, joista päästään asianomaisille sivuille. Tule tukijoukkoihin lisälinkit ovat: liity jäseneksi tai ryhdy kummiksi, kannatusjäsenmaksut, eettinen lahja-kortti, muut tavat tukea. Jäsenmaksusivulla ja eettinen lahja-korttisivulla toivotaan jonkun laista pankkiyhteys ratkaisua.

Liity jäseneksi tai ryhdy kummiksi sivulta löytyy lomake, jonka kautta uusikäyttäjä voi rekisteröityä käyttäjäksi tai ryhtyä kummiksi.

Kannatusjäsenmaksu sivulta ainakin löytyy pankin tilinumerot jonne voidaan maksaa jäsenmaksut.

Eettinen lahja-kortti on vielä ajatusvaiheessa, sivun toiminta täsmentyy myöhemmin.

Muut tavat tukea sivulta löytyy mahdollisia muita mahdollisuuksia tukea tämän yhdistyksen toimintaa.

## Yhteystiedot

Yhteystiedot linkkiä klikkaamalla päälinkkien alle aukeaa yhteystietojen lisälinkit, jotka vievät käyttäjän asianomaiselle sivulle. Yhteystietojen lisälinkit ovat: yhdistyksen yhteystiedot, hallituksen jäsenet, miten kaikki alkoi, yhdistyksen säännöt.

Yhdistyksen yhteystiedot sivulta löytyvät yhdistyksen yhteystiedot.

Hallituksen jäsenet sivulta löytyy hallituksen jäsenet.

Miten kaikki alkoi sivulta löytyy kerronta yhdistyksen aloittamisesta.

Yhdistyksen säännöt sivulta löytyvät yhdistyksen säännöt.

## Vasen palkki

Vasemmanlaidan linkkipalkista löytyvät toisen prioriteetin linkit, jotka ovat: kuvagalleria, tarinagalleria, kummikirjeet, kummikansio. Tämän linkkipalkin alapuolella ovat yhteistyökumppaneiden logot.

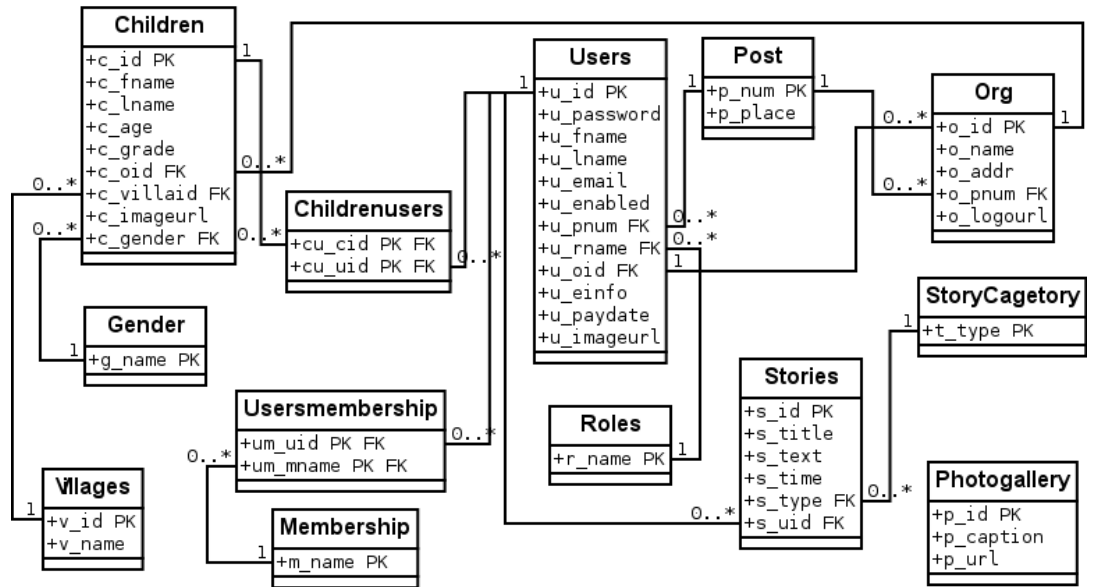
Kuvagalleriasta löytyy kuvia tapahtumista ja tapahtumasta Fason alueella Afrikassa.

Tarinagalleriasta löytyy tarinoita tapahtumista ja tapahtumasta Fason alueella Afrikassa.

Kummikirjeet sivulta löytyvät kummeille lähetetyt ja kummien lähettämiä kirjeitä.

Kummikansio sivulta löytävät kaikki kummit.

Liite 3. Tietokanta kaavio



#### Liite 4. Tietokannan attribuutit

User		
Attribuutin nimi	Attribuutin selite	Attribuutin arvo
U_id	Käyttäjän id numero	PRIMARY KEY, NOT NULL, INT, AUTO_INCREMENT
U_password	Käyttäjän salasana	VARCHAR(200), NULL,
U_fname	Käyttäjän etunimi	VARCHAR(50), NOT NULL
U_lname	Käyttäjän sukunimi	VARCHAR(50), NOT NULL
U_email	Käyttäjän sähköposti osoite	VARCHAR(75), NOT NULL
U_enabled	Käyttäjän päällä/pois arvo	INT, NOT NULL, DEFAULT 1
U_pnum	Käyttäjän postinumero	FOREIGN KEY, VARCHAR(10), NULL
U_rname	Käyttäjän rooli nimi	FOREIGN KEY, VARCHAR(20), NOT NULL
U_oid	Organisaation/yhteistyökumppanin id numero	INT, NULL
U_einfo	Käyttäjän lisätietokenttä rekisteröitymislomakkeessa	TEXT, NULL
U_paydate	Käyttäjän maksupäivämäärä	DATE, NULL
U_imageurl	Käyttäjän profiilikuva	VARCHAR(200), NULL

Roles		
Attribuutin nimi	Attribuutin selite	Attribuutin arvo
R_name	Roolin nimi, määrittää käyttäjän oikeudet	PRIMARY KEY, NOT NULL, VARCHAR(20)

Post		
Attribuutin nimi	Attribuutin selite	Attribuutin arvo
P_num	Postipaikan numero	PRIMARY KEY, NOT NULL, VARCHAR(10)
P_place	Postipaikan nimi	VARCHAR(50), NOT NULL, VARCHAR(30)

Org		
Attribuutin nimi	Attribuutin selite	Attribuutin arvo
O_id	Organisaation id numero	PRIMARY KEY, NOT NULL, INT, AUTO_INCREMENT
O_name	Organisaation nimi	VARCHAR(50), NOT NULL
O_addr	Organisaation osoite	VARCHAR(50), NOT NULL
O_pnum	Organisaation postinumero	VARCHAR(10), NULL
O_logourl	Organisaation logo	VARCHAR(200), NULL

Stories		
Attribuutin nimi	Attribuutin selite	Attribuutin arvo
S_id	Sisällön id numero	PRIMARY KEY, NOT NULL, INT, AUTO_INCREMENT
S_title	Sisällön otsikko	VARCHAR(200), NOT NULL
S_text	Sisällön teksti sisältö	TEXT, NOT NULL
S_time	Sisällön lisäys aika	TIME, NULL
S_type	Sisällön kategoria tyyppi	VARCHAR(20), NOT NULL
S_uid	Käyttäjän id numero, sisällön luoja	FOREIGN KEY, NOT NULL, INT

StoryCategory		
Attribuutin nimi	Attribuutin selite	Attribuutin arvo
T_type	Kategoria tyyppin nimi	PRIMARY KEY, NOT NULL, VARCHAR(20)

Children		
Attribuutin nimi	Attribuutin selite	Attribuutin arvo
C_id	Kummilapsen id numero	PRIMARY KEY, NOT NULL, INT, AUTO_INCREMENT
C_fname	Kummilapsen etunimi	VARCHAR(50), NOT NULL
C_lname	Kummilapsen sukunimi	VARCHAR(50), NOT NULL
C_age	Kummilapsen ikä	INT, NOT NULL
C_grande	Kummilapsen luokka	INT, NULL
C_oid	Kummilapsen koulu/organisaatio	FOREIGN KEY, INT, NULL
C_villaid	Kummilapsen kylä id	FOREIGN KEY, INT, NULL
C_imageurl	Kummilapsen kuva	VARCHAR(200), NULL
C_gender	Kummilapsen sukupuoli	FOREIGN KEY, VARCHAR(10), NULL

Villages		
Attribuutin nimi	Attribuutin selite	Attribuutin arvo
V_id	Kylän id numero	PRIMARY KEY, INT, NOT NULL, AUTO_INCREMENT
V_name	Kylän nimi	VARCHAR(30)

Gender		
Attribuutin nimi	Attribuutin selite	Attribuutin arvo
G_name	Sukupuoli	PRIMARY KEY, VARCHAR(10), NOT NULL

Childrenusers		
Attribuutin nimi	Attribuutin selite	Attribuutin arvo
Cu_cid	Lapsen id numero	PRIMARY KEY, INT, NOT NULL, FOREIGN KEY
Cu_uid	Käyttäjän id numero	PRIMARY KEY, INT, NOT NULL, FOREIGN KEY

Usersmembership		
Attribuutin nimi	Attribuutin selite	Attribuutin arvo
Um_uid	Käyttäjän id numero	PRIMARY KEY, INT, NOT NULL, FOREIGN KEY
Um_mname	Tilin tyyppi	PRIMARY KEY, VARCHAR(20), FOREIGN KEY

Memership		
Attribuutin nimi	Attribuutin selite	Attribuutin arvo
M_name	Tilin tyyppi nimi	PRIMARY KEY, VARCHAR(20), NOT NULL

Photogallery		
Attribuutin nimi	Attribuutin selite	Attribuutin arvo
P_id	Kuva id	PRIMARY KEY, INT, NOT NULL, AUTO_INCREMENT
P_caption	Kuvateksti	VARCHAR(200), NULL
P_url	Kuvan osoite	VARCHAR(200), NOT NULL

## Liite 5. Tietokannan indeksit

```
IndexUsersId_pnum (u_id, u_pnum);  
IndexUsersId_rname(u_id, u_rname);  
IndexUsersId_oid(u_id, u_oid);  
IndexOrgId_pnum(o_id, o_pnum);  
IndexChildrenId_oid(c_id, c_oid);  
IndexChildrenId_villageld(c_id, c_villaid);  
IndexChildrenId_gender(c_id, c_gender);  
IndexStoriesId_type(s_id,s_type);  
IndexStoriesId_uid(s_id, s_uid);
```



## Liite 6. Tietokannan luontilauseet

```
CREATE DATABASE bougoudia;
```

```
USE bougoudia;
```

```
CREATE TABLE post(  
  p_num VARCHAR(10) NOT NULL,  
  p_place VARCHAR(30) NOT NULL,  
  CONSTRAINT pk_postnumber PRIMARY KEY (p_num)  
)Engine=InnoDB;
```

```
CREATE TABLE roles(  
  r_name VARCHAR(20) NOT NULL,  
  CONSTRAINT pk_rolename PRIMARY KEY (r_name)  
)Engine=InnoDB;
```

```
CREATE TABLE villages(  
  v_id INT NOT NULL AUTO_INCREMENT,  
  v_name VARCHAR(30),  
  CONSTRAINT pk_villageid PRIMARY KEY (v_id)  
)Engine=InnoDB;
```

```
CREATE TABLE org(  
  o_id INT NOT NULL AUTO_INCREMENT,  
  o_name VARCHAR(50) NOT NULL,  
  o_addr VARCHAR(50) NOT NULL,  
  o_pnum VARCHAR(10) NULL,  
  o_logourl VARCHAR(200) NULL,  
  CONSTRAINT pk_organisationid PRIMARY KEY (o_id),  
  CONSTRAINT fk_o_pnum_post FOREIGN KEY (o_pnum) REFERENCES post(p_num)  
)Engine=InnoDB;
```

```
CREATE TABLE gender(  
  g_name VARCHAR(10) NOT NULL,  
  CONSTRAINT pk_gendername PRIMARY KEY (g_name)  
)Engine=InnoDB;
```

```
CREATE TABLE children(  
  c_id INT NOT NULL AUTO_INCREMENT,  
  c_fname VARCHAR(50) NOT NULL,  
  c_lname VARCHAR(50) NOT NULL,  
  c_age INT NOT NULL,  
  c_grade INT NULL,  
  c_oid INT NULL,  
  c_villaid INT NULL,  
  c_imageurl VARCHAR(200) NULL,  
  c_gender VARCHAR(10) NULL,  
  CONSTRAINT fk_orgid_corgid FOREIGN KEY (c_oid) REFERENCES org(o_id),  
  CONSTRAINT fk_villagesid_cvillaid FOREIGN KEY (c_villaid) REFERENCES villages(v_id) ON UPDATE  
    CASCADE,  
  CONSTRAINT fk_gendername_cgender FOREIGN KEY (c_gender) REFERENCES gender(g_name),  
  CONSTRAINT pk_childrenid PRIMARY KEY (c_id)
```

```
)Engine=InnoDB;
```

```
CREATE TABLE users(  
  u_id INT NOT NULL AUTO_INCREMENT,  
  u_password VARCHAR(200) NULL,  
  u_fname VARCHAR(50) NOT NULL,  
  u_lname VARCHAR(50) NOT NULL,  
  u_email VARCHAR(75) NOT NULL,  
  u_enabled INT NOT NULL DEFAULT 0,  
  u_addr VARCHAR(50) NULL,  
  u_pnum VARCHAR(10) NULL,  
  u_rname VARCHAR(20) NOT NULL,  
  u_oid INT NULL,  
  u_einfo TEXT NULL,  
  u_paypvm DATE NULL,  
  u_imageurl VARCHAR(200) NULL,  
  CONSTRAINT pk_userid PRIMARY KEY (u_id),  
  CONSTRAINT fk_pnum_post FOREIGN KEY (u_pnum) REFERENCES post(p_num),  
  CONSTRAINT fk_rname_roles FOREIGN KEY (u_rname) REFERENCES roles(r_name),  
  CONSTRAINT fk_uoid_org FOREIGN KEY (u_oid) REFERENCES org(o_id)  
)Engine=InnoDB;
```

```
CREATE TABLE membership (  
  m_name VARCHAR(20) NOT NULL,  
  CONSTRAINT pk_membership_name PRIMARY KEY (m_name)  
)Engine=InnoDB;
```

```
CREATE TABLE usersmembership (  
  um_uid INT NOT NULL,  
  um_mname VARCHAR(20),  
  CONSTRAINT pk_usersmembership_uid_mname PRIMARY KEY (um_uid, um_mname),  
  CONSTRAINT fk_usersmembership_umuid FOREIGN KEY (um_uid) REFERENCES users(u_id) ON  
      DELETE CASCADE ON UPDATE CASCADE,  
  CONSTRAINT fk_usersmembership_ummane FOREIGN KEY (um_mname) REFERENCES member-  
      ship(m_name)  
)Engine=InnoDB;
```

```
CREATE TABLE childrenusers (  
  cu_cid INT NOT NULL,  
  cu_uid INT NOT NULL,  
  CONSTRAINT pk_childrenusers_cid_uid PRIMARY KEY (cu_cid, cu_uid),  
  CONSTRAINT fk_childrenusers_cucid FOREIGN KEY (cu_cid) REFERENCES children(c_id),  
  CONSTRAINT fk_childrenusers_cuuid FOREIGN KEY (cu_uid) REFERENCES users(u_id) ON DELETE  
      CASCADE ON UPDATE CASCADE  
)Engine=InnoDB;
```

```
CREATE TABLE storycategory(  
  t_type VARCHAR(20) NOT NULL,  
  CONSTRAINT pk_categoryname PRIMARY KEY (t_type)  
)Engine=InnoDB;
```

```
CREATE TABLE stories(  
  s_id INT NOT NULL AUTO_INCREMENT,
```

```

s_title VARCHAR(200) NOT NULL,
s_text TEXT NOT NULL,
s_time DATE NULL,
s_type VARCHAR(20) NOT NULL,
s_uid INT NOT NULL,
CONSTRAINT pk_storiesid PRIMARY KEY (s_id),
CONSTRAINT fk_stype_category FOREIGN KEY (s_type) REFERENCES storycategory(t_type),
CONSTRAINT fk_s_uid_user FOREIGN KEY (s_uid) REFERENCES users(u_id)
)Engine=InnoDB;

```

```

CREATE TABLE photogallery (
p_id INT NOT NULL AUTO_INCREMENT,
p_caption VARCHAR(200) NULL,
p_url VARCHAR(200) NOT NULL,
CONSTRAINT pk_pid PRIMARY KEY (p_id)
)Engine=InnoDB;

```

```

INSERT INTO roles(r_name)
VALUES ("ROLE_ADMIN");
INSERT INTO roles(r_name)
VALUES ("ROLE_USER");

```

```

INSERT INTO storycategory(t_type)
VALUES ("storygallery");
INSERT INTO storycategory(t_type)
VALUES ("parentletters");
INSERT INTO storycategory(t_type)
VALUES ("presentmatters");
INSERT INTO storycategory(t_type)
VALUES ("thebegining");
INSERT INTO storycategory(t_type)
VALUES ("schools");
INSERT INTO storycategory(t_type)
VALUES ("projects");
INSERT INTO storycategory(t_type)
VALUES ("natureplanting");
INSERT INTO storycategory(t_type)
VALUES ("supportoptions");
INSERT INTO storycategory(t_type)
VALUE ("giftcard");

```

```

INSERT INTO org(o_name, o_addr)
VALUES ("Bougoudia Ry", "Osoitetie 2");

```

```

INSERT INTO users(u_password,u_fname,u_lname,u_email,u_enabled,u_rname, u_oid)
VALUES
("$2a$10$8oaHyvfM11HwaZRgjAOmQ.m3KpRpXxcnT8YGx/Zv9/HuqJS
jZtevG","Bougoudia","Ry","bougoudia@gmail.com",1,"ROLE_ADMIN",
1);

```

```

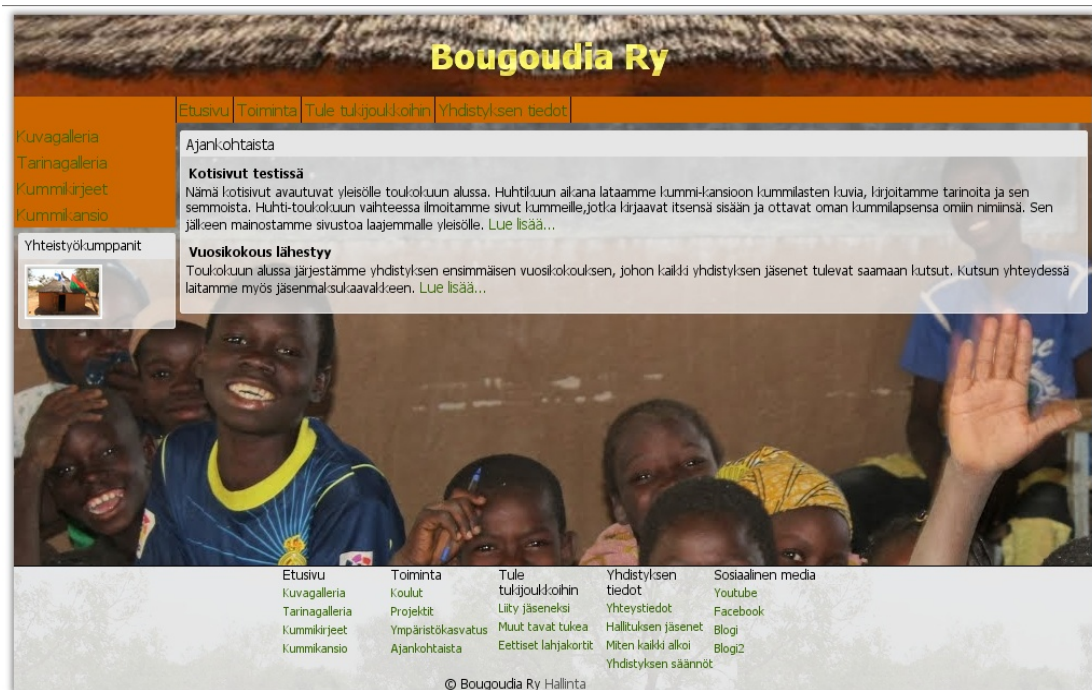
INSERT INTO gender(g_name) VALUES("male");
INSERT INTO gender(g_name) VALUES("female");

```

```
INSERT INTO membership (m_name) VALUES ("parent");
INSERT INTO membership (m_name) VALUES ("member");

CREATE INDEX i_users_id_pnum ON users(u_id, u_pnum);
CREATE INDEX i_users_id_rname ON users(u_id, u_rname);
CREATE INDEX i_users_id_oid ON users(u_id, u_oid);
CREATE INDEX i_org_id_pnum ON org(o_id, o_pnum);
CREATE INDEX i_children_id_oid ON children(c_id, c_oid);
CREATE INDEX i_children_id_villaid ON children(c_id, c_villaid);
CREATE INDEX i_children_id_gender ON children(c_id, c_gender);
CREATE INDEX i_stories_id_type ON stories(s_id, s_type);
CREATE INDEX i_stories_id_uid ON stories(s_id, s_uid);
```

## Liite 7. Bougoudia etusivu



## Liite 8. Bougoudia kummilasten hallintasivu

**Luo uusi lapsi**

Etu nimi

Sukunimi

Sisä

Luskiia

Sukupuoli Tyttö ☒ Poika ☐


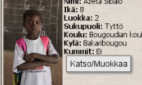


Koulu Valitse

Kylän nimi Valitse

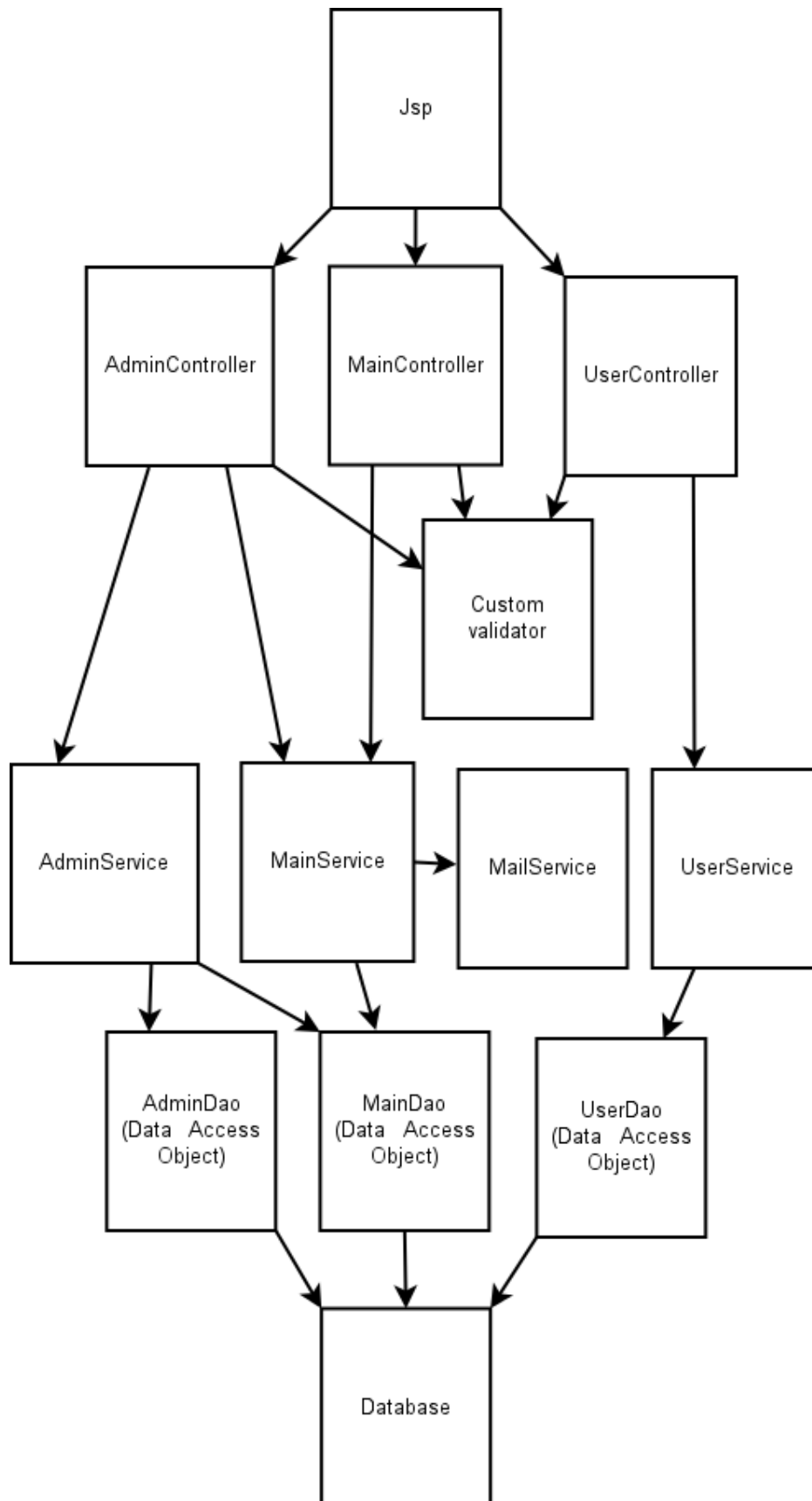
Kuva  No file selected

---

**Listaa lapsista**

Kalkiki	Kummittomat	Kummiliset	Etsi kummitella tai sukunimellä tai sisällä tai sukulla	Etsi
 <p>New: Aminata Cisse Sisä: S Luskiia: 2 Sukupuoli: Tyttö Koului: Bougoudin koulu Kylä: Salambougou Kummit: G <a href="#">Katso Muokkaa</a></p>	 <p>New: Awa Siba Sisä: S Luskiia: 2 Sukupuoli: Tyttö Koului: Bougoudin koulu Kylä: Salambougou Kummit: G <a href="#">Katso Muokkaa</a></p>	 <p>New: Fatumata Toro Sisä: S Luskiia: 2 Sukupuoli: Tyttö Koului: Bougoudin koulu Kylä: Salambougou Kummit: G <a href="#">Katso Muokkaa</a></p>		
 <p>New: Fanta Siba Sisä: S Luskiia: 2 Sukupuoli: Tyttö Koului: Bougoudin koulu Kylä: Salambougou Kummit: G <a href="#">Katso Muokkaa</a></p>				

Liite 9. Sovelluksen viestintäkaavio



Liite 10. Kommentointi tavat

```
/**  
 * Method is called from the admin controller.  
 * Method return user details.  
 */
```

Kuva 3. Javadoc kommentti

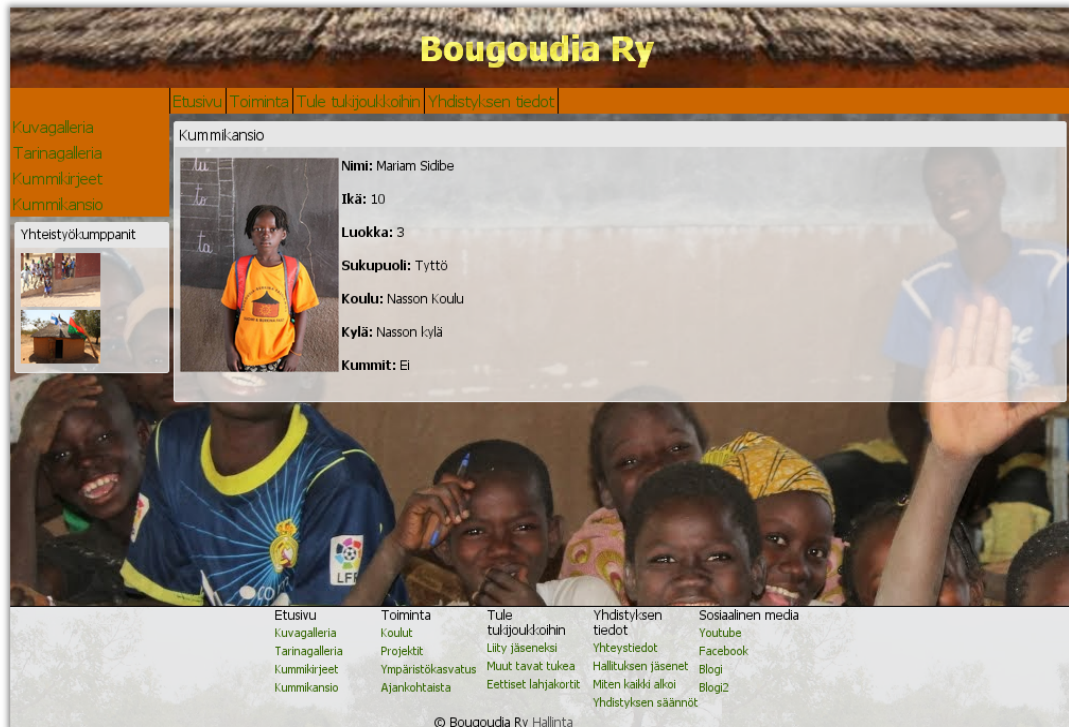
```
//If database does not have inserted post number
```

Kuva 4. Rivi kommentti

```
/*  
 * Owninfo page  
 */
```

Kuva 5. Kappale kommentti





# Bougoudia Ry


Kuvagalleria

Tarinagalleria

Kummikirjeet

Kummikansio

Yhteistyökumppanit



[Etusivu](#) | [Toiminta](#) | [Tule tulijoukkoihin](#) | [Yhdistyksen tiedot](#)

**Liity jäseneksi**

Kalikki kentät ovat pakollisia, paitsi lisätietoja kenttä. Tilin tyyppi voi olla joko kummi tai jäsen tai molemmat.

Etunimi  Sukunimi

Sähköposti

Osoite

Postinumero  Postitoimi paikka

Valitse tilin tyyppi

Jäsen ☐ Kummi ☐

Lisätietoja

Hyväksy yhdistyksen säännöt, lue säännöt täältä: [Yhdistyksen säännöt](#)

Hyväksyn ☐

Valmis

Etusivu

Kuvagalleria

Tarinagalleria

Kummikirjeet

Kummikansio

Toiminta

Koulut

Projektit

Ympäristökasvatus

Ajankohtaista

Tule tulijoukkoihin

Liity jäseneksi

Muut tavat tukea

Eettiset lahjakortit

Yhdistyksen tiedot

Yhteystiedot

Hallituksen jäsenet

Miten kaikki alkoi

Yhdistyksen säännöt

Sosiaalinen media

Youtube

Facebook

Blogi

Blogi2

© Bougoudia Ry Hallinta

# Bougoudia Ry

Kuvagalleria

Tarinagalleria

Kummikirjeet

Kummikansio

Yhteistyökumppanit



[Etusivu](#) | [Toiminta](#) | [Tule tulijoukkoihin](#) | [Yhdistyksen tiedot](#)

**Kummikirjeet**

**Ensimmäinen kummikirje**

Tulossa tekstiä.

**Kummikirje 2014**

Tässä tämä nyt tulee.



Etusivu

Kuvagalleria

Tarinagalleria

Kummikirjeet

Kummikansio

Toiminta

Koulut

Projektit

Ympäristökasvatus

Ajankohtaista

Tule tulijoukkoihin

Liity jäseneksi

Muut tavat tukea

Eettiset lahjakortit

Yhdistyksen tiedot

Yhteystiedot

Hallituksen jäsenet

Miten kaikki alkoi

Yhdistyksen säännöt

Sosiaalinen media

Youtube

Facebook

Blogi

Blogi2

© Bougoudia Ry Hallinta



## Liite 12. Järjestelmänvalvojan sivut

[Omat tiedot](#) [Tulot](#) [Terveystilinpapeli](#) [Lomakosten hallinta](#) [Uusi hakemus](#) [Uusi julkaisu](#) [Jäsenkirjeiden käyttö](#) [Täydä luo](#)

### Omat tiedot

Etunimi

Sukunimi

Sähköposti

Osoite

Postinumero ja postinimi palkit ovat ole palkissa täyttää, mutta jos haluat täyttää on molemmat täytettävä.

Postinumero

Postinimi palkit

### Vaihda salasana

Vanha salasana

Uusi salasana

Vahvista uusi salasana

### Luo uusi julkaisu

Kategoria

Otsikko

Teksti

### Lista julkaisuista

Hae sivulta

#### Vuosikokous lähestyy

Toukokuun alussa järjestämme yhdistyksen ensimmäisen vuosikokouksen, johon kaikki yhdistyksen jäsenet tulevat saamaan luvut. Klubin yhteydessä laitamme myös jäsenmaksu-asiakseen.

Julkaisu päivä: 2014-03-30

Kategoria: Ajankohtaisista

Julkaisija: Bougouda ry

Bougouda ry

#### Eettiset laajakortit

Tuossa tekstiä.

Julkaisu päivä: 2014-03-26

Kategoria: Eettiset laajakortit

Julkaisija: Bougouda ry

Bougouda ry

#### Ympäristökasvatus

Tuossa tekstiä.

Julkaisu päivä: 2014-03-26

Kategoria: Ympäristökasvatus

Julkaisija: Bougouda ry

Bougouda ry

### Luo uusi lapsi

Etunimi

Sukunimi

Ikä

Luskiä


Sukupuoli ☒ Tyttö ☐ Poika

hou

Kylän nimi

Kuva  No file selected

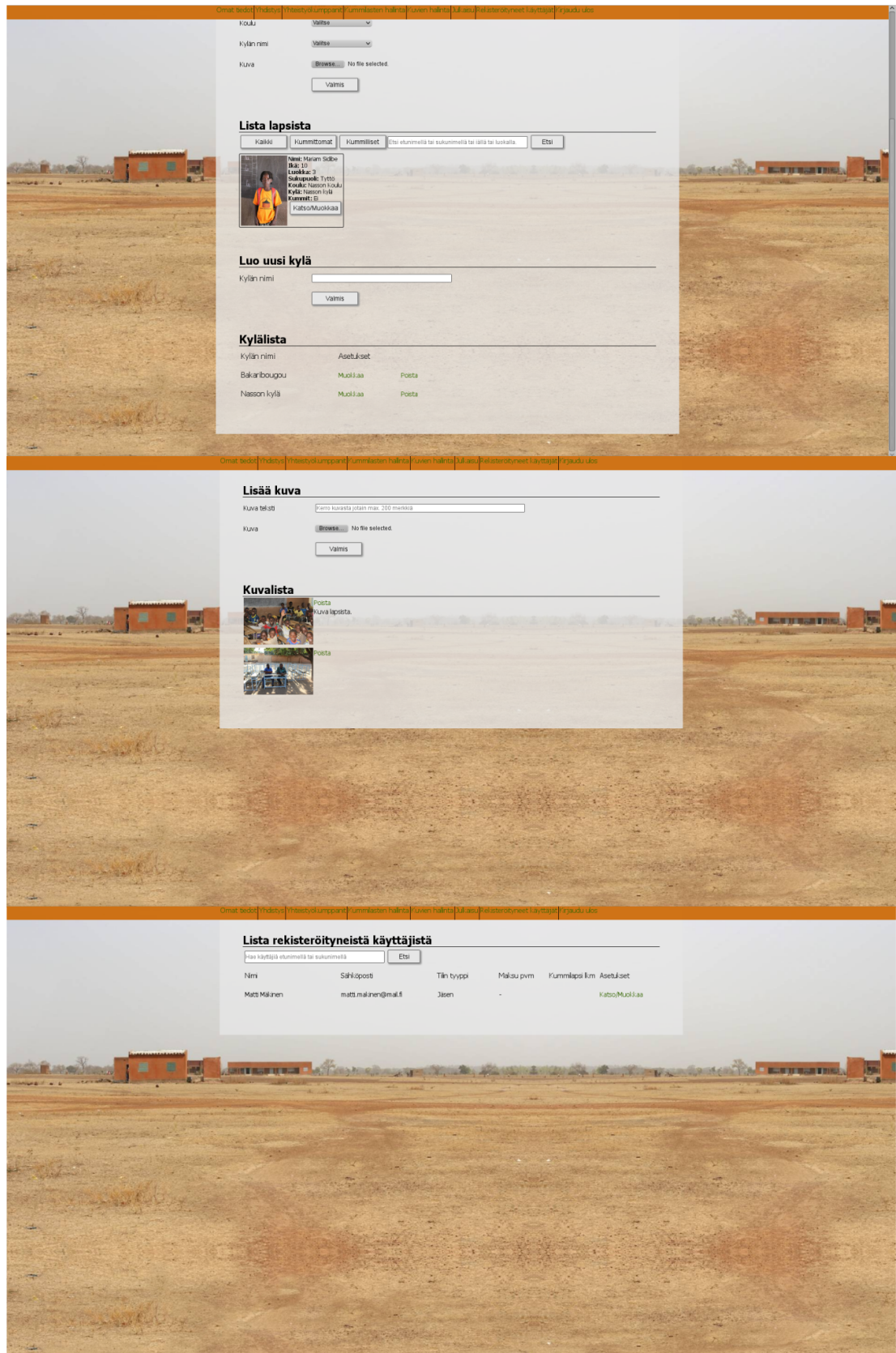
### Lista lapsista



Nimi: Marja Söder  
Ikä: 12  
Luskiä: 2  
Sukupuoli: Tyttö  
Kokous: Nainen hou  
Kylä: Nainen hou  
Kunnitt: 0  
Katso/Muokkaa

### Luo uusi kylä

Kylän nimi





[Omat tiedot](#)
[Yhteisö](#)
[Yhteisöluoppo](#)
[Kummilapset](#)
[Uusi hallinto](#)
[Uusi](#)
[Käyttäjien käyttö](#)
[Käyttö](#)

### Rekisteröityneen käyttäjän tiedot

Etunimi

Sukunimi

Sähköposti

Osoite

Postinumero

Postitoimi pöytä

Makau pvm

Tilin tyyppi

☒ Jäsen
☐ Kummi

Kummilapsi

Lisätietoja

### Kummilapset

Postinumero

Postitoimi pöytä

Makau pvm

Tilin tyyppi

☒ Jäsen
☐ Kummi

Kummilapsi

Lisätietoja

### Kummilapset

### Poista profiili

Profiili on pysyvä, eikä sitä voi perua.

[Omat tiedot](#)
[Yhteisö](#)
[Yhteisöluoppo](#)
[Kummilapset](#)
[Uusi hallinto](#)
[Uusi](#)
[Käyttäjien käyttö](#)
[Käyttö](#)

### Yhdistyksen tiedot

Nimi

Osoite

Postinumero ja postitoimi pöytä

Postinumero

Postitoimi pöytä

### Luo uusi yhdistyksen jäsen

Etunimi

Sukunimi

Sähköposti

Osoite

Postinumero ja postitoimi pöytä

Postinumero

Postitoimi pöytä

Salaus

Käyttäjätilin kirjautuminen

☐ Ei
☒ Kyllä

Käyttäjän oikeudet

Kummilapsi

Käyttäjän kirjautuminen on pakollista, mutta jos haluat käyttää sitä, sinun on oltava kirjautunut sisään. Oletusarvo on, että käyttäjän on oltava kirjautunut sisään, jotta se voi käyttää sitä.

Kummilapsen valinta on pakollista, mutta jos haluat käyttää sitä, sinun on oltava kirjautunut sisään. Oletusarvo on, että kummilapsen on oltava kirjautunut sisään, jotta se voi käyttää sitä.





### Lisää uusi yhteistyökumppani

Nimi

Osoite

Postinumero

Postitoimituspaikka

Valitse logo  No file selected.

### Yhteistyökumppanilista

Nimi	Osoite	Postinumero	Postitoimituspaikka	Asetukset
Bakaribougoun koulu	Bakaribougou			Katso/Muokkaa
Nasson Koulu	Nasso			Katso/Muokkaa

### Päivitä yhteistyökumppanin tiedot

Nimi

Osoite

Postinumero

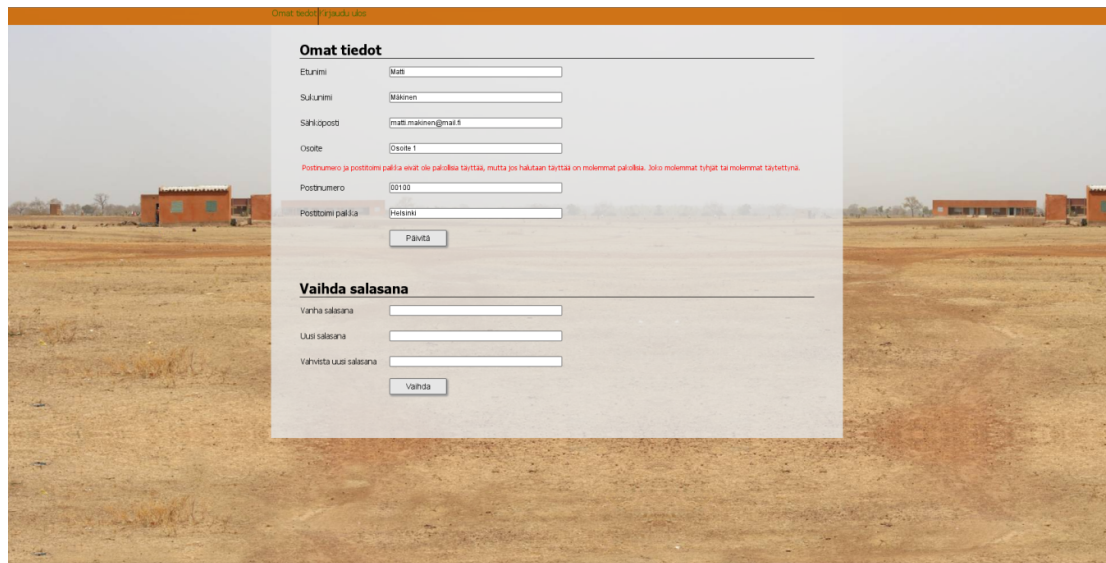
Postitoimituspaikka

Kumppanin logo  Poista logo

### Poista yhteistyökumppani

Poista yhteistyökumppani  Poisto on pysyvä, eikä sitä voi perua.

## Liite 13. Käyttäjän sivu



Omat tiedot | [Uusi salasana](#)

### Omat tiedot

Etunimi

Sukunimi

Sähköposti

Osoite

Postinumero ja postinimi pakollisia tietoja, mutta jos halutaan täyttää, niin molemmat täytyy täyttää.

Postinumero

Postinimi

### Vaihda salasana

Vanha salasana

Uusi salasana

Vahvista uusi salasana