

Vesa-Matti Salo

RESPONSIIVISEN KÄYTTÖOHJEMODUULIN SUUNNITTELU JA
TOTEUTUS

Tietojenkäsittelyn koulutusohjelma

2014

RESPONSIIVISEN KÄYTTÖOHJEMODUULIN SUUNNITTELU JA TOTEUTUS

Salo, Vesa-Matti
Satakunnan ammattikorkeakoulu
Tietojenkäsittelyn koulutusohjelma
Toukokuu 2014
Ohjaaja: Nuutinen, Petri
Sivumäärä: 41
Liitteitä: 0

Asiasanat: responsiivisuus, sisällönhallinta, php , mysql, tietojärjestelmät

Tämän opinnäytetyön tarkoituksena oli suunnitella, toteuttaa sekä integroida responsiivinen käyttöohjelmoduuli Enersense International Oy:n tuotteen E-Sense-resurssienhallintajärjestelmään.

Opinnäytetyön teoriaosuudessa tutkitaan responsiivisuutta, sisällönhallintaa, käyttöliittymäsuunnittelua, tietokantoja ja sitä miten erilaisia sovelluksen kehitysmalleja voi käyttää tietojärjestelmien kehityksessä. Lisäksi käyn läpi kaikki opinnäytetyössäni tarvittavat ja käytettävät tekniikat ja ohjelmistot. Esittelen responsiivisen sovelluksen perustoimintaperiaatteet ja komponentit, joista selviää miten responsiivisuus toteutetaan.

Toteutusosuudessa esitellään käyttöohjelmoduulin perustoimintaperiaatteita, toimeksiannon kautta rakentamalla sisällönhallintajärjestelmä, jossa on moderni ja responsiivinen käyttöliittymä. Käyttöohjelmoduuli sisältää kaikki tarvittavat ominaisuudet, jotka tarvitaan käyttöohjeiden julkaisemiseen ja hallintaan. Järjestelmä tarvitsee myös hyvin toteutetun tietokannan toimiakseen tehokkaasti ja nopeasti. Esittelen myös toteutettua tietokannan rakennetta ja käytettäviä kenttiä. Projektin päätavoitteena oli suunnitella ja toteuttaa laadukas sisällönhallinnanjärjestelmä, joka tulisi käyttöön, sekä syventää opinnäytetyössä käytettyjen ohjelmointikielien ja tekniikoiden osaamista.

THE DESIGN AND IMPLEMENTATION OF RESPONSIVE MANUAL MODULE

Salo, Vesa-Matti

Satakunnan ammattikorkeakoulu, Satakunta University of Applied Sciences

Degree Programme in Information and Communication Technology

May 2014

Supervisor: Nuutinen, Petri

Number of pages: 41

Appendices: 0

Keywords: responsiveness, cms, php, mysql, information systems

The purpose of this thesis was to design, implement and integrate a responsive manual module to Enersense International LTD's product resource management system E-Sense.

In the theoretic part of this thesis will be studied responsiveness, content management, designing the user interface, databases and how to use different software development models in developing information systems. In addition I will explain all the techniques and software that I will be using in my thesis. I will showcase the basic principles and components of a responsive software, in order to explain it.

In the concrete part of the thesis is showcased the basic functional principles of the manual module by developing the content management system, which has a modern and responsive graphical user interface. The responsive manual module consists all the needed qualities that will be needed to publish and maintain manuals. I will also showcase the implemented structure and the fields that are in use in the database. The main goal of the project was to design and implement good-quality content management system, which would later on be on use. Another goal was to improve the skill of the programming languages and techniques used on this thesis.

SISÄLLYS

1	JOHDANTO.....	6
2	YRITYS JA TOIMEKSIANNON KUVAUS.....	7
2.1	Enersense International Oy	7
2.2	E-Sense-palveluympäristö	7
2.2.1	Asiakkaiden käyttöön tarkoitetut moduulit	7
2.2.2	Yrityksen omaan käyttöön tarkoitetut moduulit.....	8
2.3	Projektin tausta	8
2.4	Lähtötilanne	9
2.5	Projektin määrittely.....	9
3	TIETOJÄRJESTELMÄ	11
3.1	Yleistä	11
3.2	Kehittäminen ja suunnittelu	11
3.3	Ohjelmistokehitysmallit.....	12
3.4	Kehitysmallin valinta.....	17
4	WEB-SISÄLLÖNHALLINTAJÄRJESTELMÄ	18
4.1	Yleistä	18
4.2	Ominaisuudet.....	19
4.3	Käyttöohjelmoduulin ominaisuuksien suunnittelu	20
5	KÄYTETTÄVÄT TEKNIIKAT JA OHJELMISTOT	21
5.1	Yleistä	21
5.2	JavaScript.....	21
5.2.1	JQuery-kirjasto	21
5.3	PHP	21
5.4	HTML5 ja CSS3	22
5.4.1	Bootstrap-kirjasto	22
5.5	Ajax.....	23
5.6	XAMPP.....	23
5.7	MySQL ja PhpMyAdmin.....	24
6	TIETOKANTA.....	25
6.1	Tietokannat yleisesti	25
6.2	Relaatiotietokannat	25
6.3	Käyttöohjelmoduulin tietokannan rakenne	25
6.3.1	Manuals-taulu	26
6.3.2	Manuals_translations-taulu	27
7	RESPONSIIVINEN WEB-SUUNNITTELU	28

7.1	Yleistä	28
7.2	Toiminnallisuus ja komponentit.....	29
7.2.1	Flexible layout	30
7.2.2	Media queries	30
7.2.3	Flexible media	32
8	KÄYTTÖLIITTYMÄ JA TOIMINNALLISUUS	33
8.1	Sivuston ulkoasu	33
8.2	Sivuston toiminnallisuus.....	35
8.2.1	Manuaalien lisääminen.....	35
8.2.2	Manuaalin tarkastelusivu.....	37
8.2.3	Muokkaus ja poistaminen.....	37
8.2.4	Manuaalien hakeminen.....	38
8.2.5	Kieltenhallinta	39
8.2.6	Kirjautuminen	40
9	YHTEENVETO	41
	LÄHTEET.....	42

1 JOHDANTO

Mobiililaitteiden yleistymisen on lisännyt responsiivisen suunnittelun tarvetta erilaisissa web-sovelluksissa. Web-sivujen vaivaton käyttäminen mobiililaitteen kautta saattaa ratkaista markkinoilla menestymisen. Jos käyttöliittymän käyttö koetaan vaivalloiseksi, on todennäköistä etteivät käyttäjät viihdy sivulla kovin pitkää aikaa. Asiointit tehdään sovelluksessa yhä useammin jatkuvassa liikkeessä.

Tämän takia kaikki web-sovellukset pyritään nykyisin tekemään eri näyttöjen välillä skaalautuviksi, eli responsiivisiksi. Tässä työssä toteutin käyttöohjelmamoduulin, jossa ohjeistetaan käyttäjiä E-Sense-resurssienhallintajärjestelmän käytössä sekä, jossa voidaan hallita sisältöä sisällönhallintajärjestelmän tavoin.

Käsittelen opinnäytetyössäni web-sisällönhallintajärjestelmiä, sellaisen suunnittelua ja toteuttamista. Pohdin myös mitä hyvä sisällönhallintajärjestelmä pitää sisällään, ja esittelen miten sellainen rakennetaan. Lisäksi käyn läpi erilaisia kehitysmalleja tietojärjestelmien toteuttamiseksi.

Käsittelen työssäni myös sovelluksen toteuttamista responsiiviseksi, eli sellaiseksi että se on dynaaminen ja skaalautuu vaihdettaessa eri näyttölaitteesta toiseen, sisällön pysyessä muuttumattomana. Sovelluksen käyttömukavuuden ja käytännöllisyyden tulee myös pysyä hyvänä vaihdettaessa näkymää suuremmasta näyttölaitteesta pienempään. Työssä käy ilmi miten responsiivisuus toteutetaan käyttämällä mediahakuja.

Opinnäytetyöhöni kuuluu myös olennaisena osana tietokannat, ja niiden suunnitteleminen tehokkaiksi ja suorituskykyisiksi. Ohjeiden hakemisen tietokannasta tulee olla nopeaa ja vaivatonta.

2 YRITYS JA TOIMEKSIANNON KUVAUS

2.1 Enersense International Oy

Enersense International Oy on resurssienhallintayritys, joka on keskittynyt etsimään ja arvioimaan toimitusketjuyrityksiä. Suurin yhteistyö tapahtuu tällä hetkellä Olkiluoto 3-ydinvoimalan rakennustyömaiden kanssa. Enersense International Oy on perustettu vuonna 2006. Olen ollut työharjoittelussa yrityksen IT-tiimissä tätä kirjoittaessani 5 kk.

2.2 E-Sense-palveluympäristö

Resurssienhallintajärjestelmä E-Sense koostuu kuudesta päämoduulista jotka ovat: HR- (Human Resources), Alihankkija- (Subcontractor), Tilojenhallinta- (Accommodation), Kuljetus- (Transportation), Laite- (Devices) sekä Rekrytointimoduuli (Recruitment for clients).

Lisäksi yrityksen omaan käyttöön on toteutettu kolme moduulia, jotka ovat: Rekrytointi- (Recruitment), CRM- (Asiakkuudenhallinta) sekä Tehtävienhallintamoduuli (Labs)

2.2.1 Asiakkaiden käyttöön tarkoitetut moduulit

HR-moduulissa käyttäjä voi hallita työntekijöitä sekä työaikoja. Kaikki dokumentit, työluvut ja sertifikaatit pidetään ajan tasalla, ja niiden mennessä umpeen järjestelmä ilmoittaa vanhentuneista dokumenteista. Työlupakurssit voidaan järjestää työntekijöille, Kellokorttiosiossa (Timesheet) työtunnit voidaan kirjata ylös sekä tuoda pois järjestelmästä muihin ohjelmistoihin (esim. Excelliin).

Alihankkijamoduulissa hallitaan yrityksiä ja urakoitsijaketjuja, kaikki tarpeelliset dokumentit pidetään ajantasalla Dokumenttien pyytämisen ominaisuuden (Document Chasing) avulla. Tilojenhallintamoduulissa hallitaan työntekijöiden asuntoja, vapaita tiloja, varataan uusia sekä käsitellään asumiseen liittyviä dokumentteja.

Kuljetusmoduulissa hallitaan yrityksen työntekijöiden käyttämiä autoja sekä listataan notifikaatiot huolloista, tarkastuksista ja rengasvaihtoista. Laitemoduulissa hallitaan laitteita, (esim. tietokoneet jne.) huoltoja sekä laitteisiin liittyviä dokumentteja jotka liittyvät laitteisiin. Laitteiden käyttäjät, vanhentuvat dokumentit ja tulevat huollot voidaan jäljittää nopeasti. Rekrytointimoduulissa asiakkaat voivat hallita yrityksensä sisäisiä rekrytointeja.

2.2.2 Yrityksen omaan käyttöön tarkoitetut moduulit

Työntekijöiden rekrytoimisesta ja hankkimisesta vastaa rekrytointimoduuli, jonne rekrytoitavat työntekijät tallennetaan. Asiakasyritysten hallinnointia säädellään CRM-moduulista käsin. Tehtävien (tasks) lisääminen, muokkaaminen sekä käyttäjähallinta tapahtuvat tehtävienhallintamoduulissa. Lisäksi tehtävienhallintamoduulissa listataan projektien valmistumisasteet sekä käyttäjälle itselleen määrätyt tehtävät.

2.3 Projektin tausta

Sain projektiksi Enersense International Oy:ltä rakentaa käyttöohjesivuston neuvomaan käyttäjiä E-Sense-resurssienhallintajärjestelmän käytössä ja siinä vastaan tulevissa mahdollisissa ongelmissa. Sivusto tulee olemaan dynaaminen sekä mobiiliin skaalautuva, eli responsiivinen. Aiempaa ohjesivustoa ei varsinaisesti ole olemassa, joten työ tulee todelliseen tarpeeseen

E-Sensen uusi beta-vaiheessa oleva versio tulee olemaan responsiivinen ja skaalautumaan eri kokoisille näyttölaitteille hyvin luettavaksi ja käytettäväksi. Näiden samojen ominaisuuksien tulee myös löytyä rakennettavasta käyttöohjelmoduulista.

2.4 Lähtötilanne

Yrityksessä ei ollut olemassa käyttäjille suunnattua ohjesivustoa, jossa hallintaoikeudet omaava (admin) voisi ohjedokumenttien kirjoitustilassa luoda ja hallita kirjoitettuja ohjeita.

Käyttöohjelmoduulissa tullaan käyttämään jo valmiiksi E-Sense ohjelmistoon sisäänrakennettua käyttäjähallintaa, joten pääpainona ei ollut käyttäjähallintaan keskittyminen. Lisäksi E-Sense ohjelmistossa valmiiksi sisäänrakennettu kieltenhallinta hoitaa myös käyttöohjelmoduulin kieltenhallinnan.

2.5 Projektin määrittely

Projektin tavoite oli suunnitella ja toteuttaa responsiivinen käyttöohjelmoduuli, joka toimisi sisällönhallintajärjestelmän tavoin, eli alusta, jonne artikkelien ja ohjeiden lisäys onnistuisi vaivattomasti. E-Sense ohjelmiston käyttäjien ohjaus käyttöohjelmoduulin oikeaan ohjeeseen tulisi olla vaivatonta ja nopeaa. Projekti tulitaisiin suunnittelemaan ja toteuttamaan käyttäen vesiputousmallia. Järjestys meni näin: Vaatimusten määrittely, että mitä ominaisuuksia tehtävään järjestelmään halutaan. Sitten järjestelmä suunniteltaisiin ja tämän jälkeen seuraisi itse toteutus. Seuraavassa vaiheessa järjestelmä testataan mahdollisten virheiden varalta, ja jos järjestelmä toimii moitteettomasti, niin se otetaan käyttöön. Viimeinen vaihe koostuu järjestelmän ylläpidosta ja siihen tulevista päivityksistä. Käyn seuraavaksi suunnitelmani vaiheet läpi.

Ensimmäisessä vaiheessa tulisi tutkia tietojärjestelmien (engl. Information Systems) kehitystapoja, tutkimalla erilaisia sovelluksien kehitysmalleja. Lisäksi tulisi tutkia eri sisällönhallintajärjestelmiä (engl. content management system) ja katsoa jo olemassa olevia avoimeen lähdekoodiin perustuvia sisällönhallintasovelluksia, joista saattaisi löytyä ominaisuuksia käyttöohjelmomoduliani varten. Eli perehtyminen sisällönhallintajärjestelmien teoriaan ja siihen, mitä ominaisuuksia hyvä sisällönhallintajärjestelmä pitää sisällään, ja myös siihen, mitä asioita tulisi välttää.

Toisessa vaiheessa tulisi suunnitella ja toteuttaa käyttöohjelmomodulin tietokanta ja ottaa huomioon jo olemassaolevan tietokannan taulut E-Sense ohjelmistossa.

Ohjeiden hakeminen tietokannasta tulisi olla nopeaa. Kolmannessa vaiheessa suunniteltaisiin käyttöliittymä, jonka tulisi olla responsiivinen, jotta sivustoa voisi käyttää myös mobiililaitteilla. Käyttöliittymän tulisi olla myös yhteensopiva E-Sense:n 2.0 version värimaailman ja typografian kanssa. Ohjedokumenttien kirjoitustilassa käyttäjällä tulee olla mahdollisuus käyttää WYSIWYG -editoria (engl. What You See Is What You Get), eli tekstinmuokkausohjelmaa jonka sisältö näyttää samalta kuin lopputulos.

Neljännessä vaiheessa toteutettava kirjautuminen ja todentaminen toteutettaisiin E-Sensen käyttäjätunnistusta käyttäen. Peruskäyttäjät voivat käyttää sivustoa ohjeiden katselemiseen ja hakemiseen. Ohjedokumenttien muokkaus, poisto sekä luominen edellyttäisi admin- eli hallintaoikeutta. Viidennessä vaiheessa toteutetaan kaikki käyttöohjelmomodulin tarvitsemat PHP-luokat ja ominaisuudet, joita järjestelmä tarvitsee toimiakseen. Kuudennessa vaiheessa järjestelmä testattaisiin ohjelmistovirheiden eli bugien varalta, jotta järjestelmä toimisi suunnitellulla tavalla, ilman odottamattomia tapahtumia.

Seitsemännessä ja viimeisessä vaiheessa järjestelmä integroidaan osaksi E-Sense ohjelmistoa hyväksymisen ja lopputestauksen jälkeen.

3 TIETOJÄRJESTELMÄ

3.1 Yleistä

Tietojärjestelmä (engl. Information systems) on järjestelmä, joka koostuu ohjelmistoista, tiedonsiirtolaitteista, tietojenkäsittelylaitteista sekä ihmisistä. Tietojärjestelmän tarkoituksena on helpottaa jotakin toimintaa, tai mahdollistaa se. (Wikipedia A 2014.) Suunnittelen ja toteutan työssäni tietojärjestelmän, joka koostuu sisällönhallinnasta, tietokannoista, ohjeiden kirjoittamisesta ja hakemisesta.

3.2 Kehittäminen ja suunnittelu

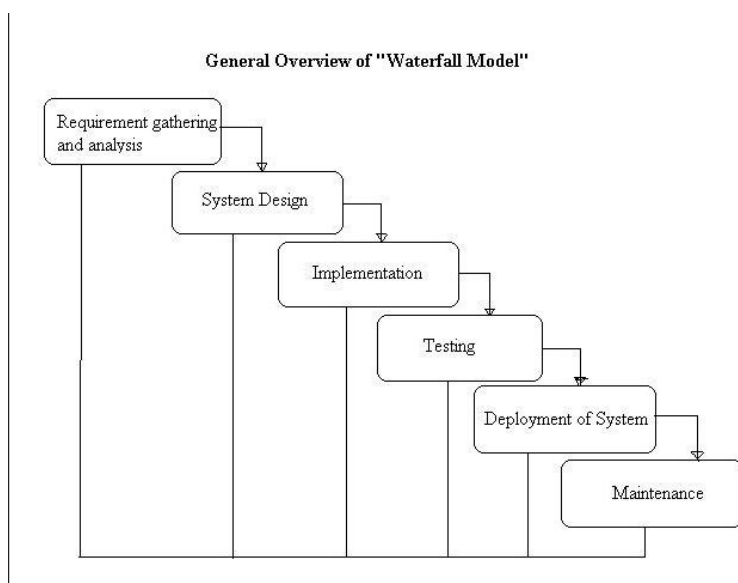
Tietojärjestelmien suunnittelumenetelmiä on kehitetty 1950-luvulta lähtien. Suunnittelijat jakaantuivat, joko keskittymään suunnittelussaan tekniseen puoleen, toiset tietojärjestelmän strategiaan vaikutuksiin organisaatioissa, toiset työn muutoksiin ja toiset kustannusten pienentämiseen. Näitä edelläänmainittuja asioita ei kuitenkaan tulisi eritellä, vaan kaikki pitäisi ottaa huomioon. Eri näkökulmien ottaminen huomioon helpottaa suunnitteluprosessia. Suunnitteluprosessin ymmärtämistä voidaan helpottaa tutkimalla järjestelmän elinkaarta, joka sisältää kaikki suunnitteluprosessin vaiheet ongelman havaitsemisesta valmiiseen ja toimivaan järjestelmään. (Cs.utu 1999.)

Kaikissa elinkaarimalleissa ilmenee viisi päävaihetta: toiminnan tai ongelma-alueen analysointi sekä järjestelmän suunnittelu, toteutus, käyttöönotto ja ylläpito. Käytettävän elinkaarimallin valinta riippuu tietojärjestelmän koosta, rakenteesta, ratkaistavasta ongelmasta, ympäristöstä, käyttäjistä, ja myös siitä minne tietojärjestelmä suunnitellaan. Koska rakentaminen vie paljon aikaa ja järjestelmät ovat laajoja sekä monimutkaisia, täytyy kehittämisprojektit jakaa osiin. Osat eli vaihejaot tunnetaan myös elinkaarimalli-nimellä (engl. life cycle models). (Cs.utu 1999.)

3.3 Ohjelmistokehitysmallit

Ohjelmistokehitysmallit ovat joukko prosesseja tai metodeja, joista valitaan sopiva projektin tavoitteiden ja halutun tuloksen mukaan. On olemassa monia elinkaarikehitysmalleja, jotka on kehitetty vastaamaan erilaiset projektit huomioon ottaen. Käyn läpi seuraavaksi erilaisia ohjelmistokehitysmalleja. (Istqbexamcertification 2012.)

Vesiputousmalli (engl. Waterfall Model), joka tunnetaan myös lineaarisena elinkaarimallina (engl. linear-sequential life cycle model). Mallin perusajatus on helppo ymmärtää, jokainen vaihe on suoritettava loppuun, ennen kuin voidaan edetä seuraavaan vaiheeseen. Jokaisen vaiheen lopussa arvioidaan, että onko projekti menossa oikeaan suuntaan, vai täytyykö arvioitava vaihe tehdä uudelleen. Kuvassa 1 kuvataan vesiputousmallin vaiheet: määrittely, järjestelmän suunnittelu, toteutus, testaus, järjestelmän käyttöönotto sekä ylläpito. Vesiputousmallia käytetään kun projektilta haluttavat asiat tunnetaan ja projekti on lyhyt. (Istqbexamcertification 2012.)

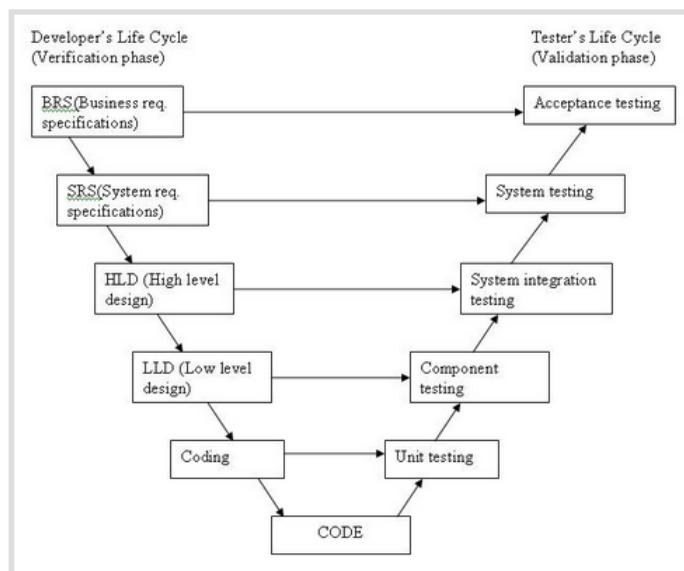


Kuva 1. Vesiputousmallin vaihedigrammi.

(<http://istqbexamcertification.com/wp-content/uploads/2012/01/Waterfall-model.jpg>)

V-malli (engl. V-model) tarkoittaa varmistus ja tarkistusmallia (engl. Verification and Validation). Aivan kuten vesiputousmallissa, jokainen vaihe on suoritettava loppuun, ennen kuin seuraava voidaan aloittaa. Kuvassa 2 on kuvattuna V-mallin eri suoritusvaiheet. Määrittely aloittaa suorituksen kuten vesiputousmallissakin. Ensin tehdään BRS (engl. Business requirements), eli vaatimusmäärittely yritykselle. Seuraavaksi tehdään SRS (engl. System requirements), eli järjestelmämäärittely. Näistä kahdesta tehdään järjestelmän testaussuunnitelma, ennen kuin edetään seuraavaan vaiheeseen. (Istqbexamcertification 2012.)

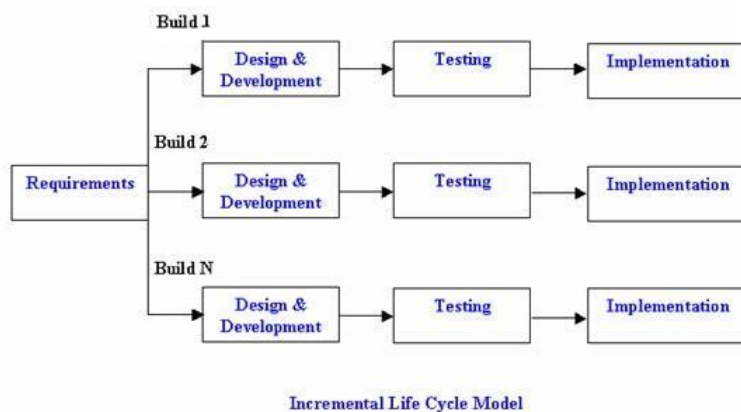
Korkeatasoisen suunnittelun-vaiheessa (engl. HLD, High Level Design), keskitytään arkkitehtuuriin ja kokoinaiskuvaan. Tätä seuraa alemmantason suunnittelu (engl. LLD, Low Level Design), jossa keskitytään sovelluksen pääkomponenttien suunnitteluun. Tämän jälkeen seuraa itse sovelluksen koodaaminen, joka löytyy V-mallin pohjalta. Kun koodaus on saatu päätökseen, seuraa V-mallin toisen puolen noudattaminen alhaalta ylös. Aiemmin tehdyt suunnitelmat toteutetaan. V-malli sopii keskisuuriin projekteihin, joissa määrittelyt ovat selkeät. (Istqbexamcertification 2012.)



Kuva 2. V-mallin vaihediagrammi.

(<http://istqbexamcertification.com/wp-content/uploads/2012/01/V-model.jpg>)

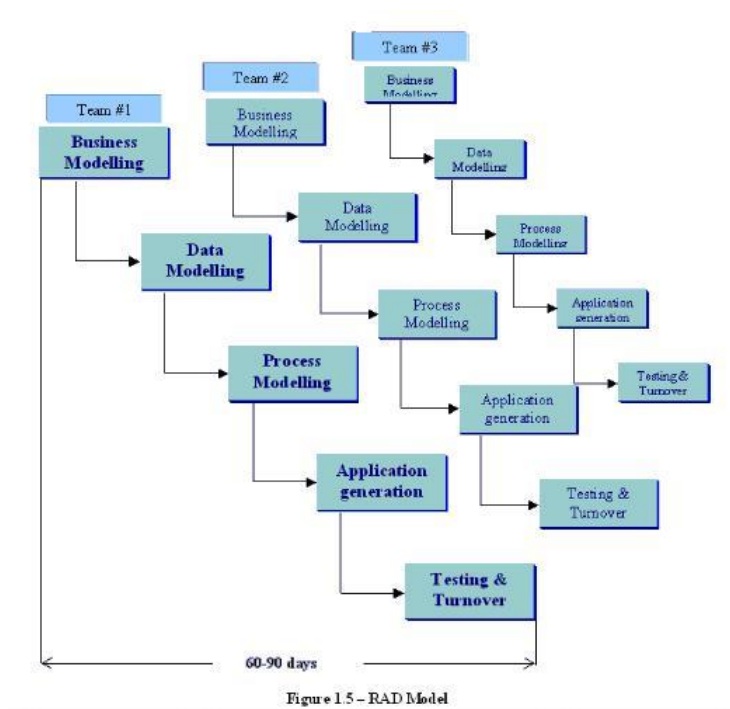
Kasvavassa-mallissa (incremental model) vaatimukset on jaettu erilaisiin versioihin (engl. build). Elinkaaresta tulee kuin vesiputousmalli monistettuna. Kasvavassa-mallissa syklit jaetaan pienempiin, helpommin hallittaviin moduuleihin. Jokainen moduuli käy läpi vaatimukset, suunnittelun, toteutuksen ja testauksen. Toimiva versio sovelluksesta syntyy jo ensimmäisen version aikana, jota aina parannellaan uusilla ominaisuuksilla seuraavissa moduulisykleissä. Tällainen lähestymistapa soveltuu hyvin, jos tuotteen nopealle julkaisulle on tarvetta. (Istqbexamcertification 2012.)



Kuva 3. Kasvava elinkaarimalli.

(http://istqbexamcertification.com/wp-content/uploads/2012/01/Incremental_model.jpg)

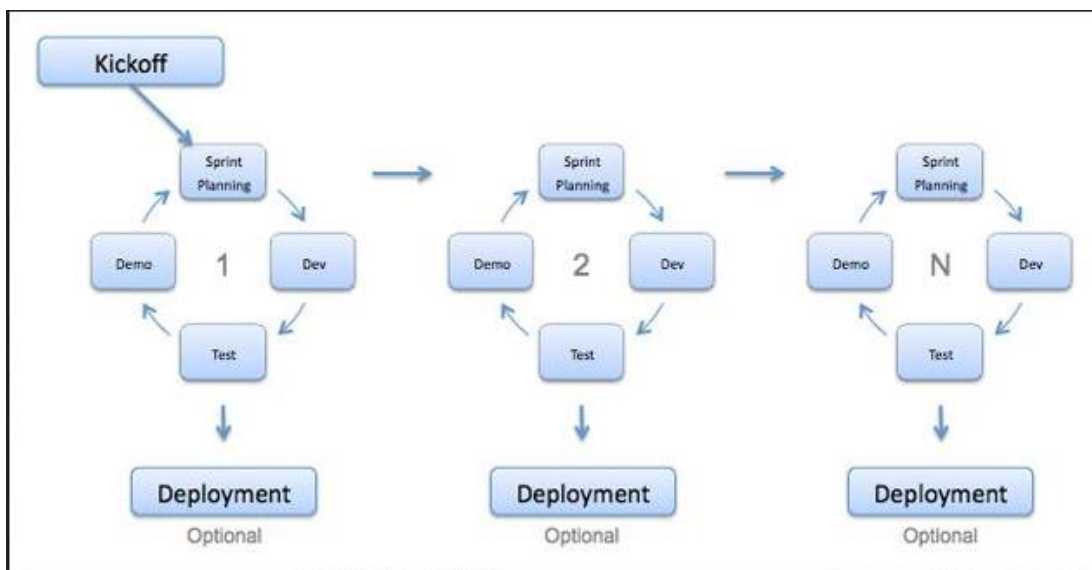
RAD-malli (engl. Rapid Application Development model) on yksi tyyppi kasvavasta-mallista. Ideana on komponenttien ja funktioiden kehittäminen rinnakkain, aivan kuten ne olisivat pieniä projekteja (kuva 4). Kehittäjäryhmiä on useita, joista kukin tekee oman toimivan prototyypinsä. Tämä antaa asiakkaalle mahdollisuuden nähdä nopeasti jotakin valmista ja konkreettista. RAD-malli soveltuu nopeisiin projekteihin, joihin voidaan panostaa palkkaamalla suurempi joukko sovellussuunnittelijoita. (Istqbexamcertification 2012.)



Kuva 4. Rad-malli.

(<http://istqbexamcertification.com/wp-content/uploads/2012/01/RAD-model.jpg>)

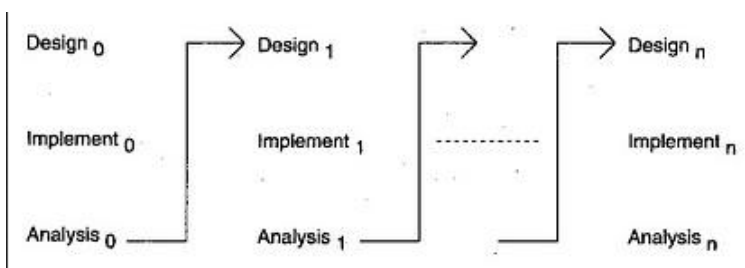
Ketterä-malli (engl. Agile-model) on myös yksi kasvavan-mallin tyypeistä. Sovellus kehitetään kasvavissa ja nopeissa sykleissä, ja tämän johdosta jokaisessa uudessa julkaistavassa versiossa on jotakin uutta, joka kasaantuu vanhan päälle (kuva 5). Jokainen julkaisu kuitenkin testataan perinpohjin, jotta voidaan olla varmoja ohjelmiston laadukkuudesta. Yksi tunnetuista ketterän kehityksen elinkaarimalleista on Extreme Programming (XP). Ketterä-malli ei tarvitse yhtä paljon suunnittelua kuin vesiputousmalli, ja uusia ominaisuuksia voidaan toteuttaa vähin kustannuksin, koska lisäykset tapahtuvat nopealla tahdilla. (Istqbexamcertification 2012.)



Kuva 5. Ketterä-malli.

(<http://istqbexamcertification.com/wp-content/uploads/2012/01/Agile-model.jpg>)

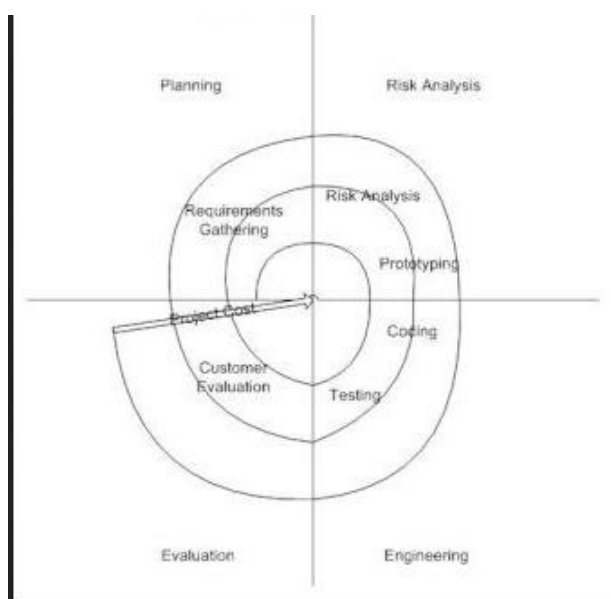
Iteratiivisessa-mallissa (engl. Iterative-model) ei aloiteta kokonaisella vaatimusmäärittelyllä, vaan sen sijaan kehitys aloitetaan tarkentamalla ja toteuttamalla vain yhden osan sovelluksesta, jotka tämän jälkeen arvioidaan ja näin ollen saadaan enemmän tietoa vaatimuksista. Tätä tehdään niin kauan kunnes haluttu lopputulos on saavutettu. Iteratiivinen-malli soveltuu isoihin projekteihin, joissa tuotetta parannellaan asteittain asiakasta jatkuvasti kuunnellen. (Istqbexamcertification 2012.)



Kuva 6. Iteratiivinen-malli, jossa jokaisen iteraation aikana tuotetta parannellaan, kunnes haluttu lopputulos on saavutettu.

(<http://istqbexamcertification.com/wp-content/uploads/2012/01/Iterative-model.jpg>)

Spiraali-mallissa (engl. Spiral model) on samoja ominaisuuksia kuin kasvavassa-mallissa, mutta sillä erotuksella, että spiraali-mallissa on enemmän painotusta riskianalyysissä. Vaiheet menevät näin: suunnittelu, riskianalyysi, koodaaminen/insinöörytyö ja evaluointi eli arvon määrittäminen. Projekti ohittaa toistuvasti nämä vaiheet iteraatioissa, joita kutsutaan tässä mallissa spiraaleiksi. Riskianalyysi-vaiheessa tunnistetaan riskit ja toteutetaan vaihtoehtoiset ratkaisut niiden välttämiseksi. (Istqbexamcertification 2012.)



Kuva 7. Spiraali-malli.

(<http://istqbexamcertification.com/wp-content/uploads/2012/01/Spiral-model.jpg>)

3.4 Kehitysmallin valinta

Pohdittuani eri kehitysmallivaihtoehtoja, päädyin käyttämään vesiputous-mallia, sen perinteisyyden ja virtaviivaisuuden takia. Myös hyvän vaatimusmäärittelyn tuomat selvät linjaukset vahvistivat tämänkaltaisen mallin käyttöä. Tein jokaisen vaiheen huolellisesti ja kiirehtimättä, koska halusin tehdä järjestelmäni kerralla kuntoon. Lisäksi harjoittelun aikana saamani varmuus käytettävistä tekniikoista lisäsi halukkuutta tämänkaltaisen prosessi-mallin käyttöönottoon.

4 WEB-SISÄLLÖNHALLINTAJÄRJESTELMÄ

4.1 Yleistä

Sisällönhallintajärjestelmä (engl. Content Management System, CMS) on ohjelmisto ja työkalu, joka sallii käyttäjän luoda, muokata ja julkaista sisältöä. Varhaisia CMS-ohjelmistoja käytettiin dokumenttien hallintaan paikallisella koneella (engl. localhost), mutta nykyään CMS-ohjelmistot tehdään poikkeuksetta sisällönhallintaan web-sovelluksiksi, eli puhutaan web-sisällönhallinnasta.

CMS-ohjelmiston tavoitteena on graafisen käyttöliittymän avulla antaa käyttäjilleen mahdollisuus web-sivujen rakenteen ja sisällön muokkaukseen. Käyttäjät voivat ilman ohjelmointitietämystä päivittää ja muokata sivujen sisältöä mieleisekseen. Yksi suosituimmista web-pohjaisista sisällönhallintatyökaluista on WordPress, joka pohjautuu avoimeen lähdekoodiin, ja on tarkoitettu blogien luomiseen ja sivustojen ylläpitämiseen. (Techterms 2013.)

Sisällönhallintaan kuuluu ainakin kolme erilaista näkökulmaa: web-sisällönhallinta, dokumenttien hallinta ja rakenteiset dokumentit. Web-sisällönhallinta (engl. Web content management) keskittyy web-sisältöjen julkaisemiseen ja tarkasteluun. Web-sisällönhallintaan kuuluvat pienet sisältöyksiköt, julkaisut jotka perustuvat sivupohjaan, metatietojen, eli tietoa tiedosta hallintaan. (Tolvanen, 2007, 8.)

Dokumenttienhallinta (engl. document management) on tärkeä osa-alue, koska yrityksen tärkeimmät tiedot ovat usein sähköisessä muodossa tallennettuna. (Tolvanen, 2007, 9.) Dokumentin elinkaari koostuu: talteenottamisesta/luomisesta, käytöstä, säilyttämisestä ja tuhoamisesta. (Salminen, 2011.) Sisällönhallinnasta suurin osa on dokumenttien julkaisua tai muokkausta. (Tolvanen, 2007, 9.) Omassa moduulissani käsitellään manuaaleja eli ohjeita, joiden avulla pääjärjestelmän E-sensen käyttäminen tehostuu.

Rakenteiset dokumentit (engl. structured documents) ovat jonkun ohjelman avulla tulkittavissa olevia dokumentteja. Rakenteisten dokumenttien hyödyt näkyvät käsittelyn automatisoinnissa, sekä metatiedossa joka saadaan dokumenteista. (Tolvanen, 2007, 9.) Rakenteisen dokumentin elinkaari koostuu seuraavista vaiheista: rakenteiden luominen, sisällöntuotanto, käyttö, säilyttäminen ja tuhoaminen(mahdollisesti vain osittain). (Salminen, 2011.)

4.2 Ominaisuudet

Web-sisällönhallintajärjestelmien yleisiin ominaisuuksiin kuuluu käyttäjäoikeuksien ja tunnuksien hallinta, kielivaihtoehtojen sekä järjestelmän suorituskyvyn ja vikasietoisuuden hallinta. (Tolvanen, 2007, 24.) Omassa toteutuksessa jätin suorituskyvyn ja vikasietoisuuden hallinnan pois ja keskityin sisällön tuottamiseen, muokkaamiseen, käyttäjähallintaan sekä kielivaihtoehtoihin.

Web-sisällönhallintajärjestelmän vakio ominaisuuksiin voidaan lukea kolme kategoriaa: sisällön koostaminen, sisältökokoelman hallinta ja julkaisun hallinta. Sisällön koostaminen koostuu kaikista toimenpiteistä, joiden avulla järjestelmään luodaan tai tuodaan uutta sisältöä. Sisältökokoelman hallinta tarkoittaa järjestelmään tuotua sisältöä voidaan hallita, organisoida tai ylläpitää. Viimeisenä oleva julkaisun hallinta tarkoittaa sisältöyksiköiden julkaisemisen muokkaamista, julkaisemisen ajastamista, tapahtumien valvomista sekä mahdollisissa ongelmatilanteissa niiden hallintaa. (Tolvanen, 2007, 25-26.)

Monissa avoimeen lähdekoodiin perustuvissa web-sisällönhallintajärjestelmissä on usein liikaa ominaisuuksia, joita käyttäjä ei tarvitse, joten on helpompaa, halvempaa ja tehokkaampaa rakentaa itse oma järjestelmä vain tarpeellisilla komponenteilla. Näin järjestelmästä tulee kevyt ja tehokas.

4.3 Käyttöohjelmomodulin ominaisuuksien suunnittelu

Suunnittelin, että toteutan kaikki manuaalimodulin tarvitsemat toiminnallisuudet php-luokkaan, jonka nimesin manuaali-luokaksi. Luokkaan pitäisi sisällyttää PDO (engl. PHP Data Objects) tietokantaluokka sisälle, joka varmistaisi että ohjeiden talletus ja hakeminen onnistuu samalla kun mahdollisten SQL-injektioiden mahdollisuus poistetaan. Tietokantayhteys tulisi luoda pääjärjestelmä E-sensen käyttämään tietokantaan, jonne tekisin modulini käyttämät taulut erikseen.

Kun turvallinen tallennus tietokantaan olisi suunniteltu, olisi aika suunnitella itse php-funktiot manuaali-luokan sisälle. En näytä lähdekoodia php-funktioista, vaan selvitän seuraavaksi, että mitä SQL-lauseita tulee käyttää kussakin tapauksessa. Asioiden kertominen yleisellä johtuu siitä, että en halunnut tehdä opinnäytetyötäni malliksi, että miten rakennetaan järjestelmä, vaan tutkielmaksi miten sisällönhallintajärjestelmää kannattaa lähteä suunnittelemaan ja, että mitkä asiat tulee ottaa huomioon.

Sisällönhallintajärjestelmään suunniteltiin vakio ominaisuudet: julkaisu, muokkaus ja poisto Ensimmäiseksi tarvitaan manuaalin tallentaminen tietokantaan. Tämä tapahtuu käyttämällä SQL:n INSERT-lausetta tallennusfunktion sisällä. Seuraavaksi julkaistua manuaalia tulisi pystyä muokkaamaan halutessaan. Tähän tarkoitukseen käytettäisiin SQL:N UPDATE-lausetta. Viimeiseksi tarvitaan vielä poistofunktio manuaalien poistamiseen järjestelmästä. Poistamistoiminnallisuus toteutetaan SQL:N DELETE-lauseella. Kaikki funktiot tehdään yhden php-luokan sisälle, jotta niiden käyttäminen olisi turvallista ja luontevaa. Tietokantaan tallentaminen ja tiedon sieltä pois hakeminen tuli tutuksi erinäisten projektien kautta, joita yrityksessä tein siellä ollessani.

5 KÄYTETTÄVÄT TEKNIIKAT JA OHJELMISTOT

5.1 Yleistä

Käyttöohjelmoduulin ulkoasu toteuttiin uusimmilla responsiivisilla tekniikoilla käyttäen JavaScriptin jQuery-kirjastoa tuomaan sivustolle näyttävyyttä ja dynaamisuutta. Toiminnallisuus ja käytettävät luokat toteuttiin palvelimella pyörivän PHP-ohjelmointikielen avulla.

5.2 JavaScript

JavaScript on Netscapen kehittämä Web-ympäristössä ajettava komentosarjakieli. Kielen nimi oli alunperin LiveScript, mutta se vaihdettiin JavaScriptiin myöhemmin. JavaScript on kevyt ohjelmointikieli jolla luodaan staattisiin HTML-sivuihin dynaamisuutta sekä responsiivisuutta manipuloiden selaimen DOM-objekteja (Document Object Model). Manipuloinnissa toiminnallisuus sidotaan tapahtumiin (engl. event). (Tutorialspoint C 2014.)

5.2.1 JQuery-kirjasto

JQuery on suuren käyttäjäjoukon saavuttanut, avoimeen lähdekoodiin perustuva JavaScript-kirjasto. Kirjasto on rakennettu helpottamaan DOM-objektien käyttöä. Käyttöliittymän rakennus sekä dynaamisten web-sivujen rakentaminen on pelkkää JavaScriptiä käytettäessä erittäin aikaa vievää, ja juuri tähän jQuery tarjoaa helpotusta. (Dubois, J 2010.)

5.3 PHP

PHP on laajalti käytössä oleva avoimeen lähdekoodiin perustuva yleiskäyttöinen ohjelmointikieli, joka soveltuu erinomaisesti web-kehitykseen, ja jota voidaan käyttää HTML-sivujen koodin seassa. PHP soveltuu hyvin aloittelijoille, koska se on helposti opittavissa oman kokemuksenkin perusteella.

PHP suoritetaan palvelimella (engl. Server-side) ja eroaa täten JavaScriptistä, joka suoritetaan asiakaspuolella (engl. Client-side). Asiakaspuolella näkyy vain PHP-koodin tulos, mutta itse koodia ei pääse tarkastelemaan. (PHP 2014.)

5.4 HTML5 ja CSS3

HTML5 on uusin versio ja tuleva standardi HTML-merkkaukielestä. Uusin versio sisältää paranneltuja ominaisuuksia sekä keinoja web-sivuilla esitettävään tietoon.

HTML5 poistaa aikaisemmat yhteensopivuusongelmat eri laitteilla, ja tekee web-sivut käyttäjäystävällisemmiksi mobiililaitteille. (Tutorialspoint B 2014.)

CSS3 on uusin standardi CSS-tyyliohjekielestä (engl. Cascading Style Sheets), jossa vanha CSS-spesifikaatio on jaettu moduuleiksi, ja johon myös lisätty uusia moduuleja. (w3schools B 2014.) Responsiivisuuden kanssa läheisimmin tekemisissä oleva moduuli on nimeltään Mediahaku (engl. media query), joka käyttää mediaominaisuuksia kuten korkeutta sekä leveyttä. Mediahaku säättää sisällön sopivaksi aina tietyn laitteen leveyden sekä korkeuden mukaan, ilman että itse sisältöä pitää muuttaa. (Mozilla 2014.)

5.4.1 Bootstrap-kirjasto

Bootstrap-kirjasto on kokoelma työkaluja joilla voidaan luoda web-sivuja sekä web-sovelluksia. Ideana on tarjota sovelluskehittäjille nopea tapa luoda hyvännäköisiä käyttöliittymiä. Versiosta 2.0, jota kutsutaan Twitter Bootstrapiksi, lähtien kirjasto on tukenut responsiivista suunnittelua, eli web-sivujen ulkonäkö säätyy automaattisesti käytettävän laitteen näytön koon ja leveyden mukaan. Uusin versio on tällä hetkellä Twitter Bootstrap 3. (Tutorialspoint A 2013.)

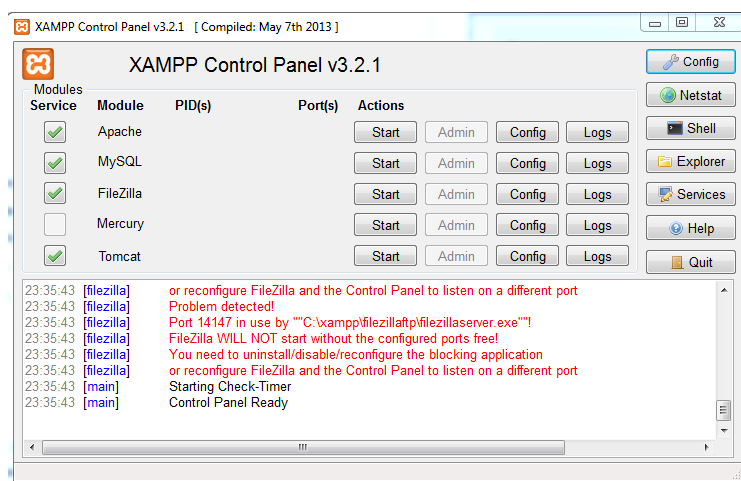
5.5 Ajax

Ajax (engl. Asynchronous JavaScript And XML) koostuu joukosta erilaisia tekniikoita, joiden avulla web-sovelluksesta voi tehdä dynaamisen. Selaimessa pyörivä ohjelma vaihtaa dataa palvelimen kanssa niin, ettei koko sivua tarvitse jokaisen pyynnön kohdalla ladata uudelleen.

Ajax on oiva apu ongelmaan, joka koskee PHP:n ja JavaScriptin suorituspaikkaeroa, eli palvelinpuoli ja asiakaspuoli. Koska PHP:tä ei voi suorittaa JavaScriptin seassa suoraan, voidaan Ajaxin avulla hakea suoritustulos erillisestä PHP-tiedostosta, ja suorittaa palvelimelta saatu vastaus JavaScriptin seassa. (W3schools A 2014.)

5.6 XAMPP

PHP-koodi tarvitsee suoritukseensa palvelimen. Käytin tässä työssä XAMPP-nimistä ohjelmistoa (kuva 8), joka sisältää tämän työn tarvitsemat välineet palvelinympäristön rakennukseen. XAMPP-ohjelmisto asentaa Apachen, MySQL:n, PHP:n käyttäjän koneelle, jolloin PHP ja MySQL toimivat lokaalisti kotikoneella (engl. localhost). Tämä helpottaa työn aloitusta, koska ei tarvitse erikseen rakentaa erillistä palvelinympäristöä. (Apache Friends 2014.)

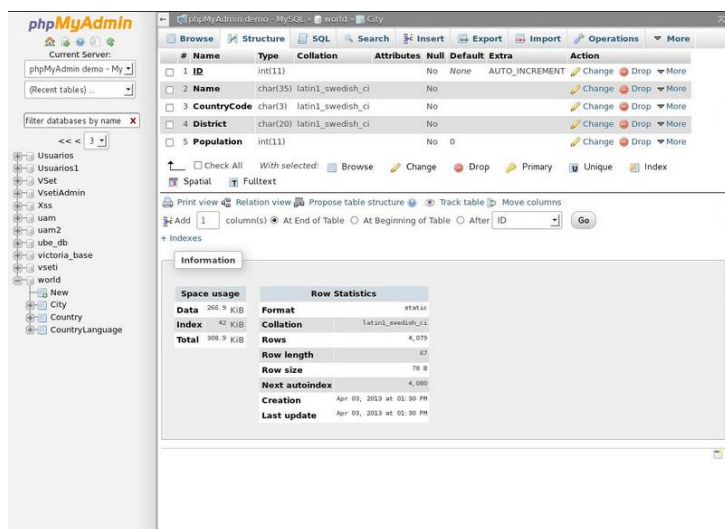


Kuva 8. XAMP-ohjelmisto.

5.7 MySQL ja PhpMyAdmin

MySQL on tietokantaohjelmisto, joka perustuu avoimeen lähdekoodiin. MySQL-tietokantaa voidaan käyttää suoraan sivustolta PHP-kielillä. PHP- tagien `<?php ?>` sisälle määritellään yhteys tietokantaan. Koodiin määritellään käytettävä palvelin, käyttäjä, salasana ja lopuksi käytettävä tietokanta. Tämän jälkeen asetetaan vielä mahdolliset virheilmoitukset ja yhteyden katkaisu, jos yhteyttä ei voitu jostain syystä muodostaa, tai jos yhteyden muodostus kestää liian kauan. (Oracle 2014.)

PhpMyAdmin (kuva 9) on MySQL-tietokannan hallintatyökalu. Tämäkin työkalu on ilmainen, ja sen avulla MySQL-serverin hallinnointi on kätevää. Työkalun avulla voidaan suorittaa erilaisia operaatioita kuten, tietokantojen, kenttien, indeksien käyttöoikeuksien hallintaa. Hallinta onnistuu joko käyttämällä suoraan graafista käyttöliittymää, tai tekemällä suoraan SQL-kyselyjä tietokantaan. (Wikipedia B 2014.)



Kuva 9. PhpMyAdmin, joka on MySQL-tietokannan hallintatyökalu.

(<http://a.fsdn.com/con/app/proj/phpmyadmin/screenshots/phpmyadmin-structure.png>)

6 TIETOKANTA

6.1 Tietokannat yleisesti

Tietokannat tarjoavat organisoidun mekanismin tiedon tallennukseen, hallintaan, käyttämiseen. Tähän tarkoitukseen käytetään tauluja, joihin tieto tallennetaan. Tietokannassa olevia tietoja hallitaan tietokannanhallintajärjestelmällä, TKHJ (engl. Database Management System). MySQL on tässä työssä käytettävä esimerkki tietokannanhallintajärjestelmästä. (Chapple 2014.)

6.2 Relaatiotietokannat

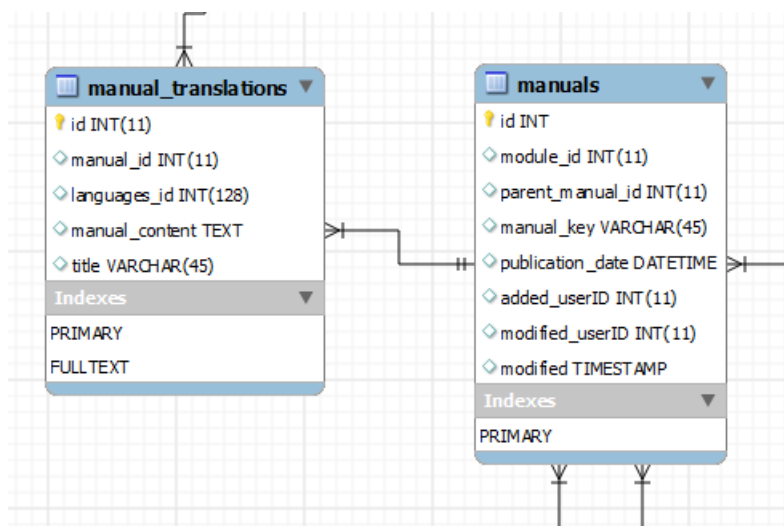
Relaatiotietokanta on joukko datakokonaisuuksia, jotka on järjestetty tauluiksi (engl. table), tietueiksi (engl. record), sarakkeiksi (engl. column) ja riveiksi (engl. row). Taulujen tunnistamiseksi, niissä on tunnistena perusavain (eng. Primary Key, PK). Lisäksi relaatiotietokannassa korostuu hyvin määritelty suhde tietokannan taulujen välillä.

Tämän tyyppisessä tietokannassa on käytössä SQL-kieli (engl. Structured Query Language), joka mahdollistaa ohjelmointirajapinnan ja tietokannan väliset kanssakäymiset sovelluksen muodossa.

(Janssen 2014.)

6.3 Käyttöohjemo-moduulin tietokannan rakenne

Aloitin tietokannan suunnittelun miettimällä, mitä tauluja ja kenttiä tarvitaan sisällönhallinnassa järjestelmässä. Tämän jälkeen toteutin suunnitelman ohjelmalla nimeltä MySQL Workbench, joka on tietokannan suunnittelutyökalu.



Kuva 10. Käyttöohjeluoduulin tietokannan päätaulut.

Tietokannasta löytyy kaksi päätaulua `manual_translations` sekä `manuals` (kuva 10), joihin on kytkettyä relaatioiden kautta kolme muuta ulkoista taulua, jotka ovat käytössä e-sensen pääjärjestelmässä. Nämä taulut on jätetty tarkoituksella pois tarkemmasta kuvauksesta.

6.3.1 Manuals-taulu

Manuals-tauluun talletetaan perusavain (`id`), jolla haetaan manuaalit yksilöllisesti, ja tämän perusteella saadaan haettua oikeat tiedot manuaalista. Manuaalien näyttämisen moduuleittain mahdollistaa moduulinumero (`module_id`), johon on talletettuna oikea moduulinumero. Jos manuaaleja lisättäessä havaitaan että manuaalit liittyvät toisiinsa, käytetään kytketymskenttää (`parent_manual_id`), josta löytyy samaan aiheeseen liittyvän manuaalin `id`. Oikea manuaali saadaan liitettyä pääjärjestelmään käyttäen manuaaliavainta (`manual_key`), jonka arvon perusteella oikea ohje saadaan. Esimerkiksi käyttäjä ohjataan e-sensen alihankkijamoduulista manuaaliavaimen (`manual_key`) perusteella oikeaan ohjeeseen.

Manuaaleja lisätessä tarvitaan myös tietoa siitä, kuka on manuaalin lisännyt ja milloin se on lisätty. Tarvitaan julkaisupäivä (`publication_date`), johon talletetaan lisäyspäivä sekä manuaalin lisännyt käyttäjä (`added_userID`). Tämän lisäksi tarvitaan manuaalien muokkauksen tehnyt käyttäjä (`modified_userID`) sekä muokkauspäivämäärä (`modified`).

6.3.2 Manuals_translations-taulu

Manuals_translations-taulu sisältää manuaalien käännökset. Perusavain (`id`) yksilöi jokaisen käännöksen. Manuals taulu kytketään käännöksiin manuaalitunnistenumeron (`manual_id`) kautta. Kieli, jolla manuaali on kirjoitettu saadaan selville kielitunnisteen (`language_id`) avulla, jossa kaikkia tuettuja kieliä vastaa numero.

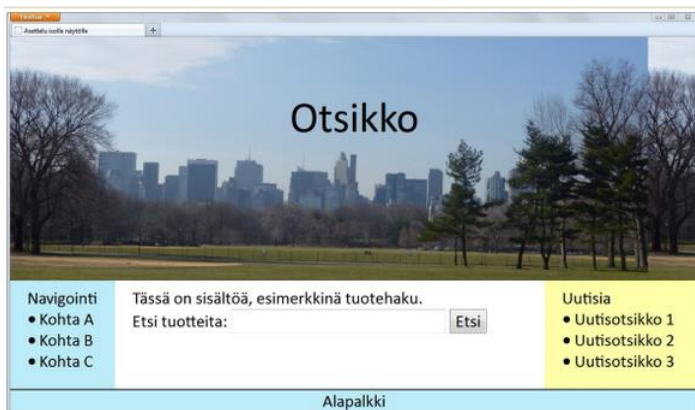
Uuden manuaalin lisäyksen yhteydessä tietokantaan talletetaan otsikko (`title`), joka kertoo manuaalin aiheen, sekä ohjeen sisältö (`manual_content`), joka tehdään tekstieditorilla.

7 RESPONSIIVINEN WEB-SUUNNITTELU

7.1 Yleistä

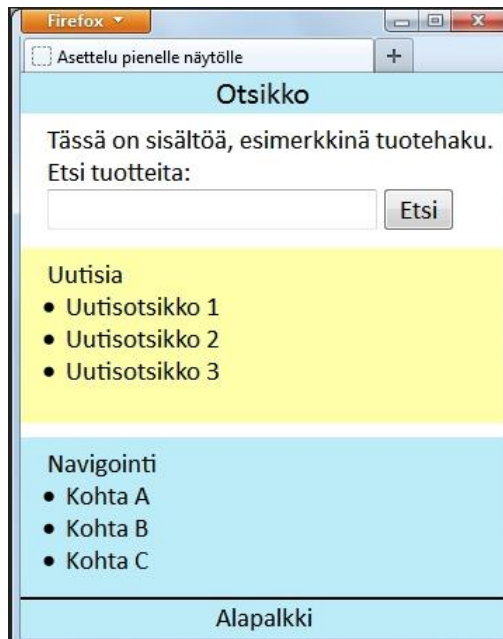
Responsiivinen web-suunnittelu tarkoittaa sitä, että web-sivu rakennetaan toimimaan jokaisessa laitteessa ja jokaisessa ruudun koossa niin, että sivu näyttää ja sen käyttäminen tuntuu luontevalta. Suunnittelussa keskitytään hyvään käyttäjäkokemukseen riippumatta siitä käytetäänkö pöytätietokoneen näyttöä tai mobiililaitetta. Responsiivisuus tarkoittaa yleisesti nopeaa reagoitua muutoksiin, kuten näyttölaitteen leveyden muutoksiin. Näin saadaan luotua sama toiminnallisuus kuin alkuperäisessä leveydessä. (Howe 2014.) Kuvassa 11 nähdään web-sivu tavallisen pöytätietokoneen suurelta näytöltä. Siirryttäessä pienemmän näytön laitteeseen (kuva 12) sisältö pysyy muuttumattomana, mutta asettelu muuttuu, mahdollistaen sivuston sujuvan käytön

Mobiilisuunnittelu eroaa responsiivisuudesta siten, että ne on rakennettu vain ainoastaan mobiilikäyttäjää ajatellen, omalle verkkotunnukselleen, domainille. Mobiilisivut ovat kevyitä, mutta alkuperäinen toteutus vain mobiilikäyttöön tuo ongelmia niin kehittäjille kuin käyttäjillekin, koodiriippuvuuksiensa takia. (Howe 2014.)



Kuva 11. Asettelu isolla näytöllä.

(<http://html5kirja.files.wordpress.com/2012/08/respons1.jpg>)



Kuva 12. Siirryttäessä kuvan 11 isosta näytöstä pienempään, sisältö pysyy samana, mutta asettelu muuttuu.

(<http://html5kirja.files.wordpress.com/2012/08/respons2.jpg>)

7.2 Toiminnallisuus ja komponentit

Responsiivinen web-suunnittelu voidaan jakaa kolmeen pääkomponenttiin, joustaviin asetteluihin (engl. flexible layout), mediahakuihin (engl. media queries) ja joustavaan mediaan (engl. flexible media). (Howe 2014.)

7.2.1 Flexible layout

Joustava asettelu on tekniikka jolla rakennetaan sivuston ulkoasu joustavalla ruudukolla (engl. flexible grid), joka pystyy dynaamisesti muuttamaan kokoaan halutulle leveydelle. Ruudukot rakennetaan käyttäen suhteellisia pituuden yksiköitä, useimmiten prosentteja tai em-yksiköitä. Näillä suhteellisilla pituuksilla määritellään ruudukon ominaisuuksien arvoja kuten leveyttä (engl. width) tai marginaalia (engl. margin). (Howe 2014.)

Joustavat ulkoasut eivät käytä suoraan määriteltyjä mittayksiköitä, kuten pixeleitä tai tuumia, siitä syystä että ikkunan korkeus ja leveys muuttuu jatkuvasti laitteesta laitteeseen. Web-sivujen ulkoasun pitää sopeutua tähän muutokseen, ja täten valmiiksi valituilla arvoilla on liian paljon rajoitteita. (Howe 2014.)

Joustavan ulkoasun kaava perustuu siihen, että otetaan kohde-elementin leveys, joka jaetaan pääelementin (engl. parent) leveydellä, jonka sisässä haluttu elementti on. Laskutoimituksen tulos on kohde-elementin leveys, joten aina kun pääelementin leveys muuttuu, muuttuvat sen sisässä olevien elementtien ominaisuuksien koot myös. (Howe 2014.)

Joustavan ulkoasun lähestymistapa ei yksinään kuitenkaan riitä. Toisinaan selaimen ikkuna voi olla kooltaan niin pieni että ulkoasun skaalaaminen suhteellisesti voi luoda ruudukoita, jotka ovat liian pieniä sisällön tehokkaaseen näyttämiseen. Jotta vältettäisiin tekstin näkyminen viallisesti, voidaan käyttää mediahakuja. (Howe 2014.)

7.2.2 Media queries

Mediahaut rakennettiin lisäosana medityypeillä, kun määritellään ja lisätään tyylejä tyyli tiedostosta (CSS). Mediahaut tarjoavat määrittelytavan eri tyyleille yksittäisille

selaimille ja laitteistotyypeille, kuten ruudun leveys tai laitteen suuntaus. Tyyliä voidaan ottaa käyttöön yksittäisesti. (Howe 2014.)

Mediahakujen käyttöönotto tapahtuu joko käyttämällä @media-sääntöä tyyli-tiedostossa, tuomalla uuden tyylin käyttäen @import-sääntöä tai yhdistämällä erillinen tyyli-tiedosto HTML-dokumentin sisällä. Yleensä on parasta käyttää @media-sääntöä jo olemassa olevan tyyli-tiedoston sisällä, jotta välttyttäisiin ylimääräisiltä HTTP-pyyntöiltä. (Howe 2014.)

Jokainen mediahaku voi sisältää mediatyyppin, jossa on yksi tai useampi ilmaisu (engl. expression). Tavalliset mediatyypit sisältävät all, screen, print, tv and braille ominaisuudet. Mediahaku ilmaisu, joka tulee mediatyyppin jälkeen voi sisältää erilaisia mediaominaisuuksia ja arvoja, jotka tämän jälkeen eritellään joko todeksi (engl. true) tai vääräksi (engl. false). Kun mediaominaisuus ja arvo osoitetaan todeksi, tyyliä otetaan käyttöön. Jos arvo osoitetaan vääräksi, tyyliä ei huomioida ollenkaan. (Howe 2014.)

Loogisia operaattoreita (and, not, or) voidaan käyttää mediahauissa, jotta saadaan varmistus siihen, että selain tai laite näyttäytyy halutulla tavalla eri ikkunakoossa. Kuvassa 2 valitaan kaikki mediatyypit, joiden leveys on väliltä 320 - 780 pikseliä. (Howe 2014.)

```
@media all and (min-width: 320px) and (max-width: 780px) { ... }
```

Kuva 2. Mediatyyppien valinta käyttäen loogisia operaattoreita.

Yksi yleisimmistä mediaominaisuuksista on pituuden tai leveyden määrittely laitteelle tai selaimen ikkunalle. Leveydet ja pituudet löydetään käyttäen height, width, device-height ja device-width mediaominaisuuksia. Jokainen näistä

ominaisuuksista voidaan etuliitteistää käyttäen min tai max-tarkennetta, kuten esimerkiksi max-device-width, joka määrittelee käytettävän laitteen maksimi leveyden. (Howe 2014.)

Responsiivisessa suunnittelussa yleisin käytetty ominaisuus sisältää min-width (suurempi tai yhtäsuuri) ja max-width (pienempi tai yhtäsuuri) määrittelyn. (Howe 2014.)

7.2.3 Flexible media

Kun ikkunan kokoa muutetaan, mediat (kuten videot) eivät usein pysy perässä. Myös kuvien, videoiden ja muiden mediatyyppien tulee olla skaalautuvia, jotta ne muuttaisivat kokoaan ikkunankoon muutoksien mukana. (Howe 2014.)

Median skaalaamiseksi voidaan käyttää max-width ominaisuutta siten, että asetetaan sen arvoksi 100%. Tämä varmistaa sen että, kun ikkunan kokoa pienennetään, niin myös pääelementin sisällä oleva media pienenee suhteessa pääelementin kokoon (kuva 3). (Howe 2014.)

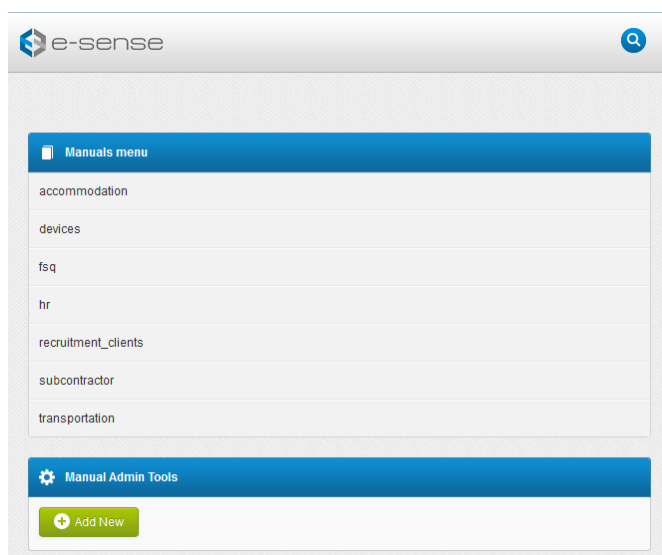
```
kuva1, video1, canvas {  
  max-width: 100%;  
}
```

Kuva 3. Median skaalaus pääelementin muuttumisen mukaan.

8 KÄYTTÖLIITTYMÄ JA TOIMINNALLISUUS

8.1 Sivuston ulkoasu

Sivuston käyttöliittymä on responsiivinen, eli sitä voidaan käyttää myös mobiililaitteilla. Lisäksi värimaailma ja typografia on E-Sensen kanssa yhteensopiva (kuva 13).



Kuva 13. Käyttöohjelmoduulin etusivu mobiilinäkymässä.

Kuvassa 4 on käyttöohjelmoduulin etusivu, josta voi valita moduuleittain haettavat manuaalit. Käyttäjähallinta tarkistaa, onko käyttäjällä muokkausoikeus, ja jos oikeudet täsmäävät niin alapuolelle ilmestyy manuaalien lisäysoikeus.

Sivuston päävalikon ja ohjeidenlisäämispainikkeen yhteyteen ladataan sisältö page-parametrin mukaan. Etusivulle kiinteästi sisällytetty getContent.php sivun sisältö vaihtuu aina haettavan moduulin mukaan (koodi 1).

```
<?php

include('getContent.php');

$page = '';

if(isset($_GET['page'])){
    $page = $_GET['page'];
}

if(isset($_GET['manual_key'])){
    include 'manual_info_page.php';
}

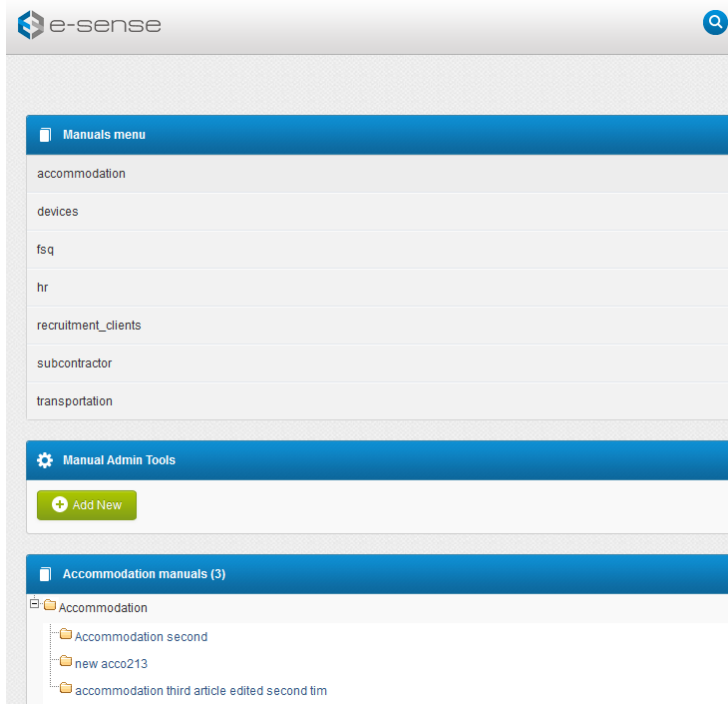
if($page == 'manualSearch'){
    include 'manualSearchPage.php';
}
if($page == 'manualAdd'){
    include 'add_manuais.php';
}

if($page == 'manualEdit'){
    include 'edit_manuais.php';
}

if($page == 'manualDelete'){
    include 'delete_manuais.php';
}

?>
```

Koodi 1. index.php:ssä sisällyttävät sivut page-parametrien mukaan.



Kuva 14. Käyttöohjeiden hakeminen moduuleittain.

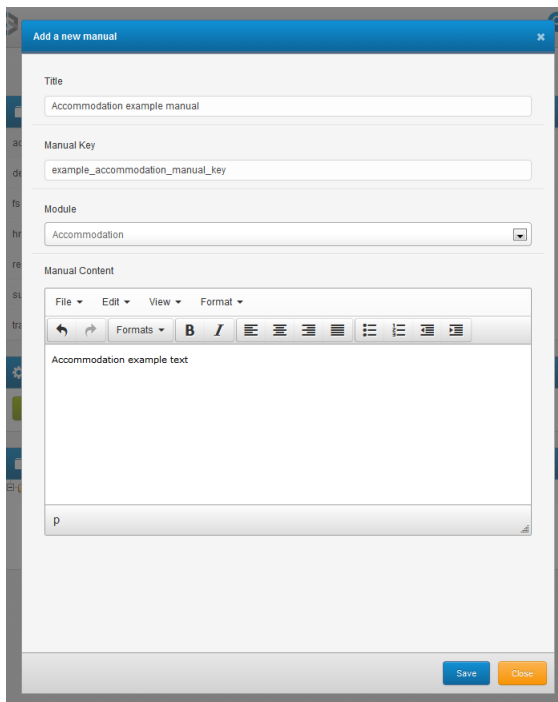
Kuvassa 14 haetaan alapuolelle sisältöä valittavan moduulin mukaan. Sivuston päävalikko pysyy samanlaisena, mutta moduulista riippuen getContent.php tiedosto hakee tietokannasta oikean ohjesisällön käyttäen manuaali-luokkaa. Haettavat manuaalit esitetään puurakenteena.

8.2 Sivuston toiminnallisuus

Sivuston toiminnasta ja sen ulkoasusta haluttiin selkeä ja toimiva kokonaisuus, jossa ohjeiden selaaminen ja muokkaaminen olisi helppoa.

8.2.1 Manuaalien lisääminen

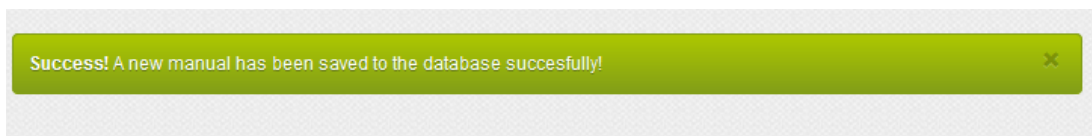
Painettaessa manuaalinlisäämispainiketta avautuu dialogi, jossa lisääminen tapahtuu (kuva 15).



Kuva 15. Manuaalin lisääminen.

Uutta manuaalia lisättäessä lisätään otsikko (title), manuaaliavain (Manual Key), moduuli johon ohje kuuluu (module) sekä itse ohjesisältö (Manual Content). Manuaalien hakeminen tapahtuu manuaaliavaimen arvon perusteella.

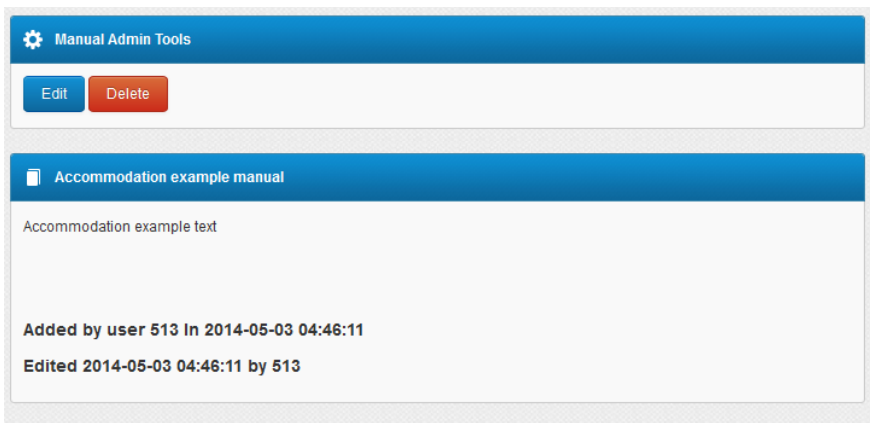
Lisäämisen voi perua painamalla close-painiketta. Sisällön ollessa oikeanlainen painetaan Save-painiketta, jolloin uusi ohje tallentuu tietokantaan. Tämän jälkeen tulee joko virheilmoitus lisäyksen epäonnistuessa, tai lisäyksen onnistuessa onnistumisilmoitus (kuva 16).



Kuva 16. Uuden ohjeen lisäyksen onnistumisen vahvistusilmoitus.

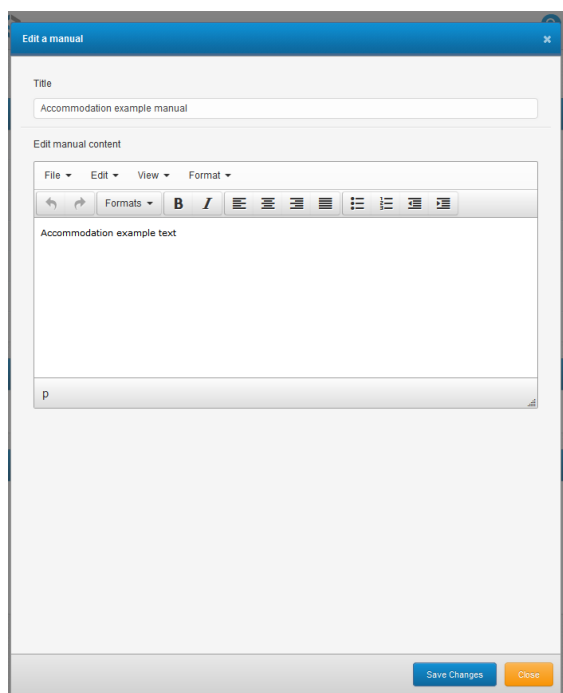
8.2.2 Manuaalin tarkastelusivu

Valittaessa jokin manuaali puunäkymästä haetaan manuaaliavaimen perusteella päävalikon alle manuaalin tarkastelusivu (kuva 17) hallinnointipainikkeineen mikäli käyttäjällä on hallintaoikeudet.



Kuva 17. Manuaalin tarkastelusivu (manual_info_page.php).

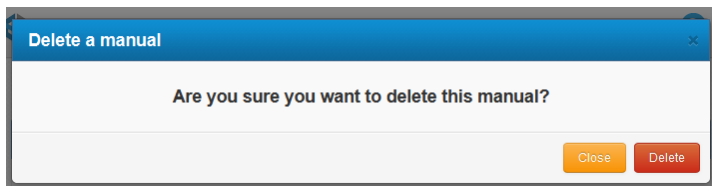
8.2.3 Muokkaus ja poistaminen



Kuva 18. Manuaalin muokkausdialogi.

Kuvassa 18 esitetään muokkausdialogi, joka avautuu muokkauspainiketta painamalla. Muokkausdialogissa voidaan vaihtaa otsikkoa sekä sisältöä, joka tallennuksen jälkeen päivittää olemassaolevan ohjeen uuteen muokattuun.

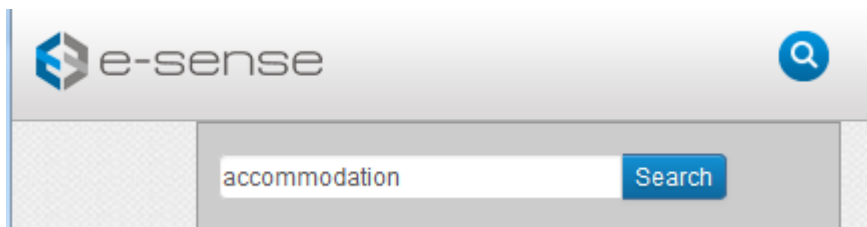
Jos olemassaoleva ohje halutaan poistaa pysyvästi painetaan poistopainiketta jonka jälkeen avautuu poistodialogi, joka pyytää vahvistusta poistolle (kuva 19).



Kuva 19. Poistodialogi.

8.2.4 Manuaalien hakeminen

Manuaalien hakemiseen löytyy hakupalkki, jota pääsee käyttämään painamalla suurennuslasin kuvaa sivuston ylälaudassa (kuva 20).



Kuva 20. Hakupalkki.

Haettaessa manuaaleja hakusanalla, sivustolle haetaan taulunäkymä, joka sisältää haettavan hakusanan (kuva 21). Manuaalien tarkastelusivulle pääsee klikkaamalla otsikkolinkkiä (Title).

Manuals search results for keyword: accommodation

Show 10 entries

Manual Search: accommodation

Manual ID	Module	Manual Key	Manual Content	Title	Publication Date	Modified Date
229	Accommodation	add_new_accommodation2	Example text.reww.	Accommodation second	2014-04-09 07:54:29	2014-04-30 17:00:28
236	Accommodation	add_new_accommodation_third_manual_key	Accommodation art.	accommodation third article edited second tim	2014-04-09 09:31:08	2014-04-09 11:40:05
239	Accommodation	example_accommodation_manual_key	Accommodation exa..	Accommodation example manual	2014-05-03 04:46:11	2014-05-03 04:46:11

Showing 1 to 3 of 3 entries

First Previous 1 Next Last

Kuva 21. Manuaalien hakunäkymä.

8.2.5 Kieltenhallinta

Sivuston kieltenhallinnassa käytetään E-Sensen käännös luokkaa, jossa on määritelty tuetut kielet. Ohjeita voidaan kirjoittaa eri kielillä tämän avulla. Kieltenhallinnassa eri kielellä E-Senseä käyttävät otetaan huomioon ja sivuston kieli säätyy käyttäjän kielitunnisteen (language_id) mukaan.

E-Sensessä käyttäjän valitsema kieli kulkee istunnon (session) mukana, ja käyttöohjelmoduuli näyttää käyttäjälle tämän mukaan sisällön englanniksi suomeksi tai saksaksi (kuva 22). Valittaessa alasvetovalikosta kieleksi esimerkiksi suomi, käyttäjä ohjataan kielenvaihtosivulle, jossa kieli vaihdetaan onnistumisilmoituksen mukana suomen kieltä vastaavaan kielitunnisteeseen.



Kuva 22. Käyttöohjelmoduulin E-Sensestä istunnon mukana tullut kieli, jonka voi vaihtaa painamalla alasvetovalikosta.

Hallintointioikeuden omaavalla henkilöllä on manuaalien lisäyksessä mahdollisuus kirjoittaa ohje eri kielellä kuin sen hetkinen session-kieli on. Manuaalien lisäyksessä session-kieli valitaan etukäteen automaattisesti pudotusvalikkoon.

8.2.6 Kirjautuminen

Käyttäjät tunnistetaan E-Sensen käyttäjätunnistusta käyttäen. Peruskäyttäjät voivat käyttää sivustoa ohjeiden hakemiseen ja katselamiseen. Käyttäjät, joilla on hallinnointioikeus (admin-oikeus) voivat muokata, poistaa ja luoda uusia ohjeita.

9 YHTEENVETO

Tämän opinnäytetyön tekeminen oli erittäin opettavaista ja sain soveltaa paljon käytäntöön oppimiani asioita. Työ osoittautui paljon laajemmaksi ja haastavammaksi kuin olin alunperin suunnitellut. Koin useita kertoja oivalluksen hetkiä toteutusvaiheessa. Olen lopputulokseen tyytyväinen ja moduulista tuli sellainen kuin siitä pitikin tulla, hyvän määrittelyn ja työkuvausten ansiosta. Jos aloittaisin projektin alusta, niin ottaisin projektin kuluvan ajan paremmin huomioon, ja suunnittelisin aikataulun sen mukaan.

Opin paljon PHP:stä ja MySQL:stä työtä tehdessäni, myös tietoturva piti pitää kokoajan mielessä sovellusta tehdessäni. Koska sovellus tulee oikeaan käyttöön piti siitä tehdä niin hyvä ja turvallinen kuin mahdollista. Tietoturva-asioita tullaan vielä muuttamaan ja lisäämään jatkokehityksessä, koska sovellus ei koskaan ole täysin valmis, ja parannavia asioita tulee mieleen vaatimusten muuttuessa.

On olemassa paljon valmiita sisällönhallintajärjestelmiä, mutta niissä on usein liian paljon tarpeettomia ominaisuuksia. Tekemällä oman järjestelmän sivustosta tulee paljon hallittavampi ja helppokäyttöisempi. Ohjeiden tuottaminen on helppoa ja luontevaa.

Tämä projekti oli suurin ja myös samalla mielenkiintoisin mitä olen tehnyt koskaan. PHP:n käyttö tuntuu nyt luontevalta ja sujuvalta, ja aionkin jatkaa kielen opiskelua lisää jatkossa. Myös tulevaisuudessa opinnäytetyössä käytetyt ohjelmointikieliet sekä tekniikat tulevat olemaan mukana erilaisissa projekteissa ja tietojärjestelmissä, joita tulen tekemään.

LÄHTEET

Apache Friends. 2014. About the XAMP project. Viitattu 28.5.2014. Saatavissa:

<https://www.apachefriends.org/about.html>

Chapple, Mike. 2014. What is a Database? Viitattu 8.4.2014. Saatavissa:

<http://databases.about.com/od/specificproducts/a/whatisadatabase.htm>

CS.utu. 1999. tietojärjestelmien_peruskurssi. Viitattu 27.5.2014. Saatavissa:

http://www.cs.utu.fi/kurssit/tietojarjestelmien_peruskurssi/syksy_1999/kehit.htm

Dubois, Julien. 2010. Part Two: Introduction to JQuery – a Powerful JavaScript Library.

Viitattu 28.3.2014. Saatavissa:

<http://www.oracle.com/technetwork/java/intro-jquery-166845.html>

Howe, Shay. 2014. Responsive web design. Viitattu 13.5.2014. Saatavissa:

<http://learn.shayhowe.com/advanced-html-css/responsive-web-design>

Istqbexamcertification. 2012. What are the Software Development Models? Viitattu: 27.5.2014. Saatavissa:

<http://istqbexamcertification.com/what-are-the-software-development-models/>

Janssen, Cory. 2014. Relational Database (RDB). Viitattu 10.4.2014. Saatavissa:

<http://www.techopedia.com/definition/1234/relational-database-rdb>

Mozilla. 2014. CSS media queries. Viitattu 31.3.2014. Saatavissa:

https://developer.mozilla.org/en-US/docs/Web/Guide/CSS/Media_queries

Oracle. 2014. About MySQL. Viitattu 28.5.2014. Saatavissa:

<http://www.mysql.com/about/>

PHP. 2014. What is PHP? Viitattu 31.3.2014. Saatavissa:

<http://www.php.net/manual/en/intro-what-is.php>

Salminen, Airi. 2011. Rakenteiset dokumentit. Mitä hyötyä niistä on? Viitattu 28.5.2014. Saatavissa:

<http://users.jyu.fi/~airi/presentations/AIPA-2011.pdf>

Techterms. 2013. CMS. Viitattu 13.5.2014. Saatavissa:

<http://www.techterms.com/definition/cms>

Tolvanen, Perttu. 2007. Web-sisällönhallintajärjestelmä – ominaisuudet ja käyttöönotto.

Jyväskylän yliopisto. Tietojenkäsittelytieteenlaitos. Pro gradu-tutkielma. Viitattu 13.5.2014. Saatavissa:

<http://www.projekti55.fi/tutkielmat/2007-gradu-web-sisallönhallintajarjestelma.pdf>

Tutorialspoint A. 2013. Bootstrap. Viitattu 4.4.2014. Saatavissa:

<http://www.tutorialspoint.com/bootstrap/index.htm>

Tutorialspoint B. 2014. HTML5 Overview. Viitattu 31.3.2014.

Saatavissa:http://www.tutorialspoint.com/html5/html5_overview.htm

Tutorialspoint C. 2014. JavaScript Overview. Viitattu 26.3.2014. Saatavissa:

http://www.tutorialspoint.com/javascript/javascript_overview.htm

W3schools A. 2014. AJAX Tutorial. Viitattu 27.5.2014. Saatavissa:

<http://www.w3schools.com/ajax/default.ASP>

W3schools C. 2014. CSS3 Introduction. Viitattu 31.3.2014. Saatavissa:

http://www.w3schools.com/css/css3_intro.asp

Wikipedia A. 2014. Information systems. Viitattu 27.5.2014. Saatavissa:

http://en.wikipedia.org/wiki/Information_systems

Wikipedia B. 2014. PhpMyAdmin. Viitattu 28.5.2014. Saatavissa:

<http://en.wikipedia.org/wiki/PhpMyAdmin>