

Mikko Heiskanen

Tietohallinnon vuorojärjestelmän kehittäminen

Metropolia Ammattikorkeakoulu

Insinööri (AMK)

Tietotekniikka

Insinöörityö

22.4.2015

Tekijä Otsikko	Mikko Heiskanen Tietohallinnon vuorojärjestelmän kehittäminen
Sivumäärä Aika	28 sivua 22.4.2015
Tutkinto	insinööri (AMK)
Koulutusohjelma	tietotekniikka
Suuntautumisvaihtoehto	ohjelmistotekniikka
Ohjaajat	lehtori Kimmo Saurén ATK-palvelupäällikkö Mikko Mäkelä
<p>Insinööriytyön tarkoituksena oli tehdä parannuksia vuorojärjestelmäpalveluun, joka on Metropolia Ammattikorkeakoulun Tietohallinnon työntekijöiden sisäisessä käytössä. Työntekijöiden ajan tasalla pitäminen Tietohallinnon organisaatiossa on helpompaa sellaisen vuorojärjestelmän avulla, jolla vuorojen muokkaaminen on nopeampaa ja työntekijät voivat tarkistaa toistensa paikallaolon myös sähköpostin välityksellä.</p> <p>Vuorojärjestelmän uusinta perustui työntekijöiden asettamiin tarpeisiin ja parannustoimenpiteet tehtiin heidän antamansa palautteen perusteella.</p> <p>Työ toteutettiin käyttämällä CodeIgniterin ohjelmistokehystä, jonka todettiin olevan helpporakenteinen ja yksinkertainen. Käyttäjäystävällisyys oli tärkeä vaatimus vuorojärjestelmän parantamisessa. Ohjelmointikielenä käytettiin web-sivujen suunnitteluun tarkoitettua PHP-kieltä ja dynaamisuuden lisäämiseksi käytettiin JavaScript-kieltä. Toimivuuden testaamiseksi käytettiin XAMPP-nimistä ohjelmistoa, joka tukee MySQL-tietokantoja.</p> <p>Vuorojärjestelmän toimintojen parantamisprojektin lopputuloksena oli toimiva ja yksinkertainen järjestelmä, joka täytti käyttäjien ja työn tilaajan odotukset.</p>	
Avainsanat	CodeIgniter, PHP, MySQL, JavaScript, vuorojärjestelmä

Author Title	Mikko Heiskanen Improving Metropolia IT administrative schedule system
Number of Pages Date	28 pages 22 April 2015
Degree	Bachelor of Engineering
Degree Programme	Information and Communications Technology
Specialisation option	Software Engineering
Instructors	Kimmo Saurén, Senior Lecturer Mikko Mäkelä, IT Service Manager
<p>The goal of this thesis was to improve the personnel scheduling system of the IT services of Helsinki Metropolia University of Applied Sciences. Keeping personnel up-to-date in the organization is easier when such a schedule system exists that provides a quick way of editing shift information and that enables the personnel to check each other's presence by e-mail.</p> <p>Renewing of the scheduling system was based on needs set by the personnel and enhancements were done according to their feedback.</p> <p>Programming was performed by using the CodeIgniter framework which was found to have a simple structure and to be easy to use. User-friendliness was a crucial requirement for the improvement work. The PHP programming language was used for coding web-pages and the JavaScript programming language was used for increasing the dynamism of the system. A program called XAMPP was used for testing purposes because it supports MySQL databases.</p> <p>The outcome of the project for improving the personnel scheduling system was a well-functioning and user-friendly system, which fulfilled the expectations of the users and the commissioner.</p>	
Keywords	CodeIgniter, MySQL, PHP, JavaScript, scheduling system

Sisällys

Lyhenteet

1	Johdanto	1
2	Web-sovellusten kehitys	1
2.1	Sovelluskehukset	1
2.2	Sisällönhallintajärjestelmät	2
2.3	Sovelluskehysten valinta	3
2.4	Sovelluskehysten hyödyt	5
2.5	PHP	6
2.6	JavaScript	7
2.7	CSS	7
2.8	XAMPP	7
3	Projektinhallintamenetelmät	8
3.1	Vesiputousmalli	8
3.2	Scrum	9
4	Toiminnot ja ominaisuudet	13
4.1	Vuorot	13
4.2	Suodatus	14
4.3	Lomakkeet	16
4.4	Sähköposti	16
4.5	Loki	17
5	Tietoturva	18
6	Projektin toteutus	19
6.1	Suunnitteluvaiheet	19
6.2	Käytetyt menetelmät	20
6.3	Projektin suoritusvaiheet	20
6.4	Käytetyt ohjelmistot	24
7	Yhteenveto	24
	Lähteet	26

Lyhenteet

Apache	PHP- ja SQL- kieliä tukeva avoimen lähdekoodin palvelinalusta.
CMS	Content management system. Sisällönhallintajärjestelmä.
CSRF	Cross-site request forgery, tietoturva-aukko, joka mahdollistaa sähköpostiviestin lähettämisen toisen henkilön nimissä.
CSS	Cascading Style Sheets. Web-sivuston muotoilusta vastaava osio, joka määrittelee sisällön asettelun WWW-sivulla.
HTTP	Internetin tiedonsiirtoprotokolla, jota selaimet ja palvelimet käyttävät.
MVC	Ohjelmistoarkkitehtuurityyppi, jonka avulla eri osat, malli, näkymä ja ohjain yhdistetään toimivaksi kokonaisuudeksi.
PHP	Web-ohjelmoinnissa käytettävä ohjelmointikieli.
SMTP	Sähköpostiviestin lähetysprotokolla kahden palvelimen välillä, perustuu TCP-protokollaan.
SQL	Structured Query Language. Relaatiotietokannan kyselykieli, jolla haetaan tietoa tietokannasta.
XSS	Cross Site Scripting. Tietoturva-aukko, jonka rikollinen pystyy hyödyntämään ja syöttämään haitallista ohjelmistokoodia.

1 Johdanto

Insinööriyön aiheena on Metropolia Ammattikorkeakoulun Tietohallinnon käyttäjätuen sisäisessä käytössä olevan työvuoropalvelusivuston uudistaminen. Työ toteutettiin kesätyöharjoitteluna Metropolia Ammattikorkeakoulun tilauksesta. Tavoitteena oli saada helppokäyttöinen, monipuolinen, vakaa ja yksinkertaisia toimintoja sisältävä järjestelmä. Tietoturvan kannalta uudistetun järjestelmän tulisi olla turvallinen, sillä se sisältää tärkeitä tietoja työntekijöiden työvuoroista. Työvuorojärjestelmää käyttävät pääasiassa Metropolian Helpdeskissä vakituisesti työskentelevät työntekijät ja he voivat päivittää kalenteria oman ajoituksensa mukaan. Vuorovaikutus työntekijöiden kesken tulisi olla sujuvaa ja heidän työvuoronsa jatkuvasti ajan tasalla. Käytettävyyden parantamiseksi pohdittiin erilaisia vaihtoehtoja ja toteuttamisen aikana tehtiin kyselyitä työntekijöiltä kartoittaen heidän mielipiteistään liittyen projektin vaatimuksiin.

Vuorojärjestelmän uudistaminen toteutettiin työharjoitteluna kahtena kesänä peräkkäin vuosina 2013 ja 2014 ja muutostyö ajoitettiin toukokuun lopusta elokuun loppuun. Tarkoituksena oli toteuttaa järjestelmämuutos aikataulun puitteissa kehittämällä samalla osaamistasoa ohjelmointitaidossa. Osaamista parannettiin ja työkokemusta kerrytettiin käyttäen sekä PHP- että JavaScript-kieliä.

Projektin toteuttamiseen käytettyjä ohjelmistoja, työvaiheita ja niiden osa-alueiden määrittelyjä tullaan käsittelemään tässä insinööriyössä.

2 Web-sovellusten kehitys

2.1 Sovelluskehikset

Suunniteltaessa toimivaa web-sivustoa ensisijainen tavoite on usein saada järkevä ja toimiva kokonaisuus. Siirtyminen toiminnosta toiseen ilman pitkiä odotusaikoja ja viiveitä parantavat käytettävyyttä. Ohjelmistokehyksen kaikki osa-alueet ovat siististi jaoteltuna yhdessä paketissa eikä suunnittelijalta vaadita suurta tietämystä web-ohjelmoinnista, mutta PHP-kieltä olisi hyvä hallita riittävästi alkuun pääsemiseksi. Aloitteijalla on mahdollisuus päästä alkuun, koska ohjelmistokehyksen ohjelmistokoodi sisältää valmiiksi luotuja luokkamäärittelyksiä, joista jokainen ohjelmoija voi valita

halutunlaisen lähestymistavan asian ytimeen. Valmiita ohjelmointikehyksiä on saatavilla internetistä maksuttomina versioina ja valtaosa niistä on toteutettu avoimella lähdekoodilla, johon jokainen web-ohjelmoinnista kiinnostunut voi lisätä omia ideoitaan ja parannusehdotuksiaan. Sovelluskehukset ovat valmiiksi rakennettuja ohjelmistopohjia, joihin ohjelman suunnittelija ja ohjelmoija voi kehittää halutunlaisen ohjelmakokonaisuuden tai tietyn tyyppisen osan ohjelmistosta olemassa olevan ohjelmistokehyksen päälle käyttäen yleensä PHP-ohjelmointikieltä. Näitä ohjelmistokehyksiä, jotka soveltuvat PHP-kielellä ohjelmoimintiin, ovat esimerkiksi Bootstrap, Yii, CodeIgniter, CakePHP ja ZooP. [1.]

2.2 Sisällönhallintajärjestelmät

Sisällönhallintajärjestelmiä (engl. Content management system) on monenlaisia, joista web-sovelluksen suunnittelija voi valita sopivimman vaihtoehdon omaan tarkoitukseensa. Hallittavaa tietojärjestelmää voidaan hallita yhtenä kokonaisuutena, että jokaista elementtiä ei tarvitse hallita erikseen. Käyttökohteina voivat olla esimerkiksi www-sivusto, dokumentin tapainen sivusto tai verkkokaupan sivusto. Sisällönhallintajärjestelmän elementit ovat helposti muokattavissa ja rakenteeltaan yksinkertaisia peruskäyttäjälle, joka haluaa saada aikaan suunnittelemansa näköisen sivuston ilman hankalia toimenpiteitä. [2; 3.]

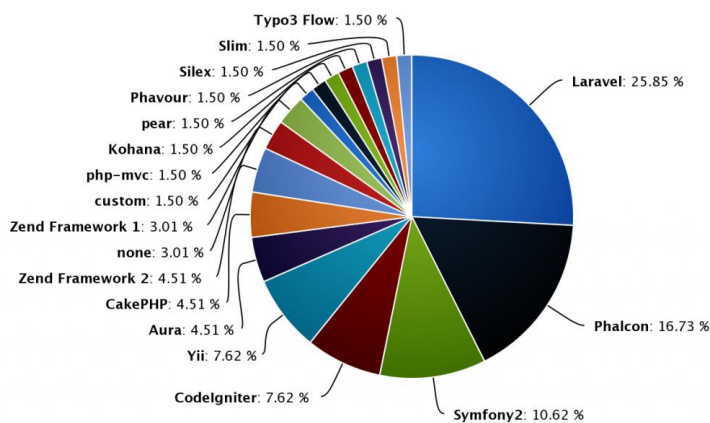
Yleisimpiä maksuttomia ja avoimella lähdekoodilla toteutettuja sisällönhallintajärjestelmiä ovat WordPress, Joomla! ja Drupal. Mainitut sisällönhallintajärjestelmät sisältävät ominaisuuksia ja toimintoja, joita aloittelija voi hyödyntää hyväkseen käyttämällä yleisimpiä internet-selaimia. Ohjelmointia vaativia taitoja tai web-sovelluksiin liittyviä aikaisempaa kokemusta ei tarvitse olla lainkaan, vaan sivuston sisältö ja tietokannan asetusten määrittely ovat tehty mahdollisimman yksinkertaisiksi. Käyttäjä voi syöttää MySQL-tietokannan tarvittavat tiedot tietokantaan käyttäen internet-selainta. Käyttöliittymä on hallittavissa internet-selaimen kautta ja vapaasti internetistä ladattava sisällönhallintajärjestelmä sisältää valmiiksi PHP-kielellä ohjelmoidun paketin, joka asennetaan tietokoneelle. [2; 3.]

2.3 Sovelluskehityksen valinta

Tietohallinnon vuorojärjestelmä oli toiminnaltaan hyvin hidas. Kun alettiin suunnitella vuorojärjestelmän rakennetta uudelleen, huomattiin, että käyttöliittymä oli huonosti toteutettu. Käyttäjystävällisyys ja tietoturva olivat vanhassa vuorojärjestelmässä kaukana vaatimuksista, joita eheästä kokonaisuudesta vaaditaan. Metropolian työntekijöitä haastateltiin ja pyydettiin ideoita ja parannusehdotuksia toiminnallisuuksista, joita voitaisiin lisätä. Ohjelmistosuunnittelijan toimesta ehdotettiin, että PHP:tä käyttävä ohjelmistokehitys nimeltä Codelgniter voisi olla hyvä vaihtoehto järjestelmän uudelleenrakentamiseksi.

Yhdysvaltalaisen ohjelmistoyrityksen EllisLab Inc.:n kehittämä ohjelmistokehitys nimeltään Codelgniter on julkaistu vuonna 2006 [4]. Maksuttomuuden vuoksi se onkin yksi suosituimmista ohjelmistokehityksistä, kuten näkyy kuvan 1 kaaviosta [5]. MVC-malliin perustuva Codelgniterin ohjelmistokehitys tukee PHP:n lisäksi MySQL-tietokantoja. Tietoturvallisuutta parantavat ominaisuudet on otettu huomioon, ja ne estävät haitallisten koodipätkien syöttämisen järjestelmään, kuten verkkohyökkäykset (XSS, CSRF).

Framework popularity, end of 2013; SitePoint

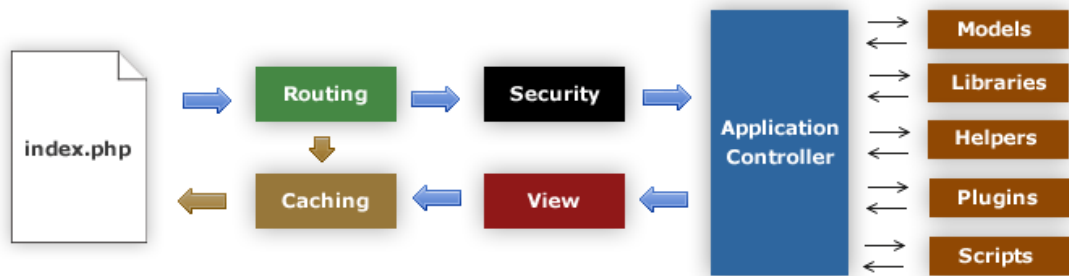


Highcharts.com

Kuva 1. Suosituimmat ohjelmistokehitykset [5].

Codelgniterin valttina olevat helppo hakemistorakenne ja asetukset ovat suurimmaksi osaksi valmiiksi määritettyjä. Havaittiin, että uusi ohjelmistokehitys reagoi nopeammin

käyttäjän antamiin pyyntöihin. CodeIgniter käyttää hyväksi MVC-arkkitehtuurin toimintaperiaatetta. Model-View-Controller-arkkitehtuurin rakenne ja toiminta esitetään kuvassa 2. CodeIgniter on suunniteltu toteuttamaan verkkosovellusten (applikaatioiden) rakennetta ja sovellus sisältää jo valmiiksi rakennettuja kirjastoja. Ohjelman rakenne alkaa index.php-nimisestä tiedostosta, jonka pystyy halutessaan poistamaan ja korvaamaan sen .htaccess-tiedostolla. [6.]



Kuva 2. CodeIgniter MVC-arkkitehtuurin toiminta (EllisLab 2015)

CodeIgniterin ohjelmistokehykseen ytimeen kuuluu kolme osaa:

- Model (malli)
- View (näkyvä)
- Controller (ohjain).

Malliosio huolehtii järjestelmän rakenteesta ja antaa ohjeita ohjaimelle. Funktiot määrittellään mallissa, jotka määrittävät järjestelmän tietokantarakenteen. Tietokannassa olevien tietojen lisäys, poistaminen ja muokkaus tapahtuvat mallin kautta.

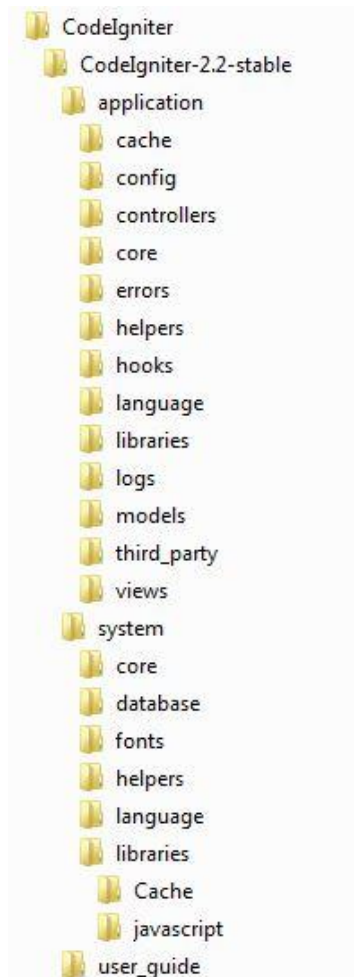
Tiedon (datan) käsittelijänä toimiva ohjain käsittelee funktioita ja tietoja, jotka liikkuvat näkymän ja ohjaimen välillä. Ohjain on CodeIgniterin pääydin, ja se vastaa kaikista tulevista HTTP-protokollan kyselyistä.

Näkyvä määrittelee ja huolehtii asettelusta eli siitä, minkälainen ja millä tavalla käyttäjälle tullaan näyttämään järjestelmän rakenne esim. www-selaimessa. HTML-kieli käännetään mallissa sopivaan muotoon, jotta www-selain ymmärtää ja osaa näyttää sivun muotoilut oikein jäseneltynä. [7]. CSS-tyylitiedoston kautta määrittellään sivun

näkymää ja jäsentelyä, jotta tekstit ja elementit eivät mene päällekkäin toistensa kanssa.

2.4 Sovelluskehiksen hyödyt

Hakemistorakenne on selkeä ja yksinkertainen. Sen alakansioissa on erilaisia tiedostoja, joihin ohjelmoija voi lähteä lisäämään ja muokkaamaan haluamiaan toimintoja. Tietokannan määrittelyasetukset ovat yhdessä tiedostossa. Turvallisuutta huolehtivat osa-alueet, kuten SQL-injektion ehkäisemiseksi suunnatut osiot on sijoitettu CodeIgniterin ytimeen. Kuvassa 3 on CodeIgniterin hakemistorakenne, alkuperäisessä muodossaan, ja sen tiedostot ovat ladattavissa EllisLab:in kotisivuilta yhtenä pakettina. [8.]



Kuva 3. CodeIgniterin hakemistorakenne [8].

Tärkeimpiä kansioita CodeIgniterin hakemistorakenteessa ovat sovelluskansio (application) ja järjestelmäkansio (system). Sovelluskehittäjät voivat luoda uusia kirjastoja ja muokata valmiita olemassa olevia kirjastoja sovelluskansion alakansioihin, jotka vaikuttavat vain uudelle luotavalle sovellukselle. Muutokset eivät vaikuta kehyksen rakenteeseen.

Sovelluskansio sisältää model-, controller- ja view-kansioiden lisäksi config-kansion, jossa voidaan määrittää palvelun sivun ohjaus oikeaan paikkaan tietojen haun tietokannasta web-sivustolle. Config.php – tiedoston avulla voidaan määrittää aloitussivu, joka on yleensä index.php-niminen tiedosto. Ohjelman suorittamiseen vaadittavat ohjaimet haetaan index.php-tiedostosta. Prosessin jälkeen CodeIgniter ohjaa suorituksen config-kansiossa olevaan routes-tiedostoon, samalla tarkistaa käsittelemällä HTTP-pyynnön ja käy läpi, että kaikki on kunnossa. Lopulta sivun perusnäkyvä ladataan web-selaimeen. Mikäli sivun sisältö ei ole samaan aikaan välimuistissa, CodeIgniter lataa tarvittavat mallit ja resurssit. System-kansio sisältää kokonaisuudessaan CodeIgniterin kehysrakenteen, joka ylläpitää CodeIgniterin toimintaa.

2.5 PHP

Yleisin web-sovelluksen toteuttamiseen käytetty kieli on PHP-ohjelmointikieli (Hypertext Preprocessor). Avoimeen lähdekoodiin perustuva PHP-ohjelmointikieli, joka muistuttaa paljolti Perl-ohjelmointikieltä. Sitä käytetään dynaamisten web-sivujen luomiseen. PHP-ohjelmointikieleen sisältyy monipuolisia luokkia ja kirjastoja. Ohjelmointikielen hyötypuolena on se, ettei koodin rakennetta pysty saamaan selville www-selaimen kautta. Ohjelmointikielen käskyä ei suoriteta etukäteen missään vaiheessa, vaan se käydään läpi suorituksen aikana. Viimeisin vakaa julkaisu PHP-ohjelmointialustasta on versio 5.6.8, joka on julkaistu huhtikuussa 2015. Haavoittuvuusiakin löytyy, joten uudet päivitykset parantavat tietoturvaa. [9; 25.] PHP on CodeIgniterin sovelluskehityksen kanssa käytettävä ohjelmointikieli, jos tehdään web-sovelluksia.

2.6 JavaScript

WWW-sivustoissa sivun dynaamiset komponentit on toteutettu JavaScript-kielessä, joka on oliopohjainen komentosarjakieli. JavaScriptillä voidaan rakentaa web-sivustolle palkkeja, josta napautetaan hiirtä painamalla ja esimerkiksi uusi valikko aukeaa näkyville. Valmiita JavaScriptin kirjastoja on saatavilla ja yksi niistä on jQuery-kirjasto. Tietohallinnon käyttämät vuorojärjestelmän alasvetopalkit (engl. dropdown menus) ja päivämäärän valitsemiseen toteutettu kalenteri (engl. datepicker) on rakennettu käyttäen jQuery-kirjastoja. [10.] CodeIgniterin ohjelmistokehyksessä jQuery-kirjastot lisätään omiin kansioihin, kuten Tietohallinnon vuorojärjestelmän hakemistorakenteessa on tehty. Kuvassa 14 sivulla 21 näytetään, kuinka JavaScriptin jQuery-kirjasto on lisätty alikansioon osaksi CodeIgniterin toimintaa. JavaScript-moduulien osia kutsutaan CodeIgniterin määrittelyissä.

2.7 CSS

Sivustojen ulkoasujen määrittely luodaan käyttäen CSS:n (engl. Cascading Style Sheets) -tyylimäärittelyä, jotka voivat sisältää allekkain yhdessä tiedostossa. Jokaiselle sivun komponentille on luotu määrittely, esimerkiksi kuinka leveä yläpalkista halutaan saada, jotta se näyttäisi halutun kokoisena www-sivustolla. Värienhallinta toteutetaan sivun kullekin elementille määrittämällä haluttu värisävy käyttämällä joko värin nimeä tai heksakoodistoa. [11.] CodeIgniterin hakemistorakenteeseen on lisätty erikseen alikansio, jossa on sivuston ulkoasun määrittelyyn vaadittavat elementit.

Sivuston skaalautuvuutta näytön resoluution mukaan kutsutaan toisella nimellä responsiivisuus. Kannettavien laitteiden, kuten älypuhelinien ja tablettien yleistymisen myötä sama sivusto tullaan näyttämään siistissä muodossa ja navigointi sivujen välillä on sujuvaa. [12.]

2.8 XAMPP

Palvelinteknologiaa käytettäessä Apachen palvelinohjelma nimeltään XAMPP sopii hyvin testausympäristöksi, jossa on mahdollista rakentaa oma web-sovellusympäristö riippumatta työskentelyalustasta tai palvelinalustasta. Luotu testausjärjestelmä voidaan

testata XAMPP:illa paikasta riippumatta. XAMPP sisältää valmiiksi määriteltyjä PHP-kirjastoja ja MySQL-tietokantatuken. Käyttöjärjestelmiä, joihin XAMPP on saatavilla, ovat Windows, Linux, Mac OS X. [13.] CodeIgniterin ohjelmistokehys on yhteensopiva käytettäväksi XAMPP:in kanssa. CodeIgniterin kanssa käytettiin XAMPP-palvelinohjelmaa testausympäristön toteuttamiseksi.

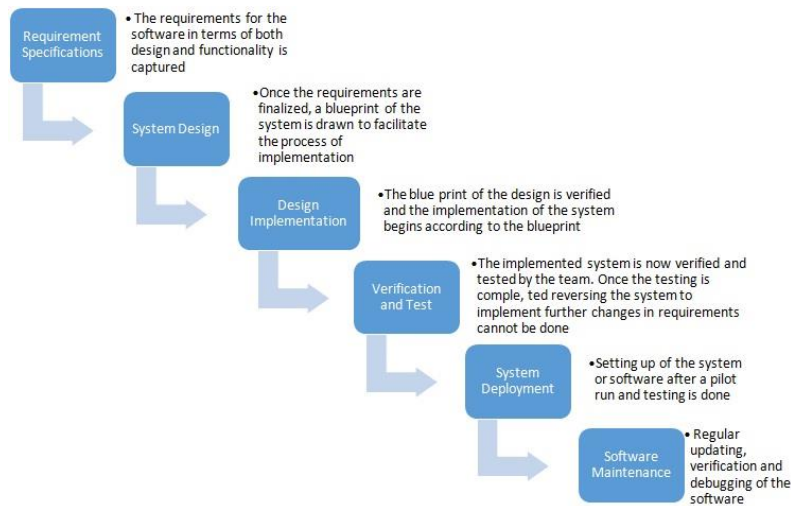
3 Projektinhallintamenetelmät

Ohjelmistosuunnittelun projektivaiheisiin kuuluu myös erilaisia tapoja hallita projektin etenemisen kulkua eri vaiheiden kautta toimivaan ja eheään lopputuotteeseen. Projektin suoritusvaiheisiin kuuluvat suunnitelman luominen, aikataulun laatiminen, seurantapalaverit, asiakirjojen dokumentointi ja tuotteen tai palvelun toteutus. Tavoitteena on saada aikaan määrättyssä ajassa hinta-laatusuhteeltaan mahdollisimman tehokas ja tarpeeksi hyvä tuote, joka palvelee laaja-alaisesti suurta käyttäjäkuntaa. Tarvittaviin korjauksiin ja parannuksiin tulee varata aikaa projektin prosessivaiheiden välissä. [14.]

Projektinhallintamenetelmiä on kehitetty monenlaisia, mutta tässä luvussa esitellään yleisimpiä tapoja ohjata projektin etenemistä.

3.1 Vesiputousmalli

Suoraviivaisuus on vesiputousmallin keskeisin ominaisuus, Projektin vaiheet suoritetaan edeten yhteen suuntaan eikä paluuta edelliseen vaiheeseen välttämättä tehdä, kuten kuvassa 4 esitellään.

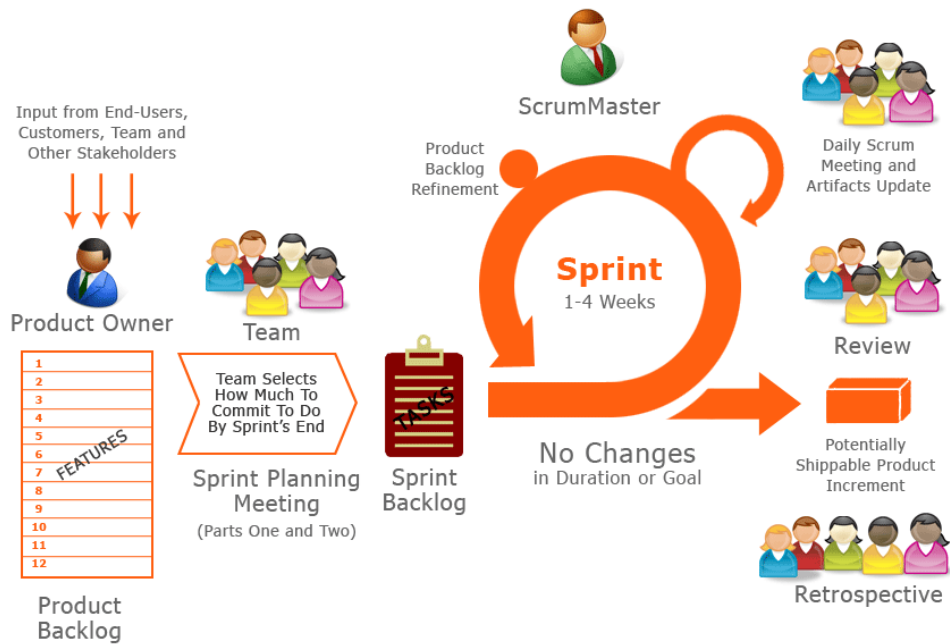


Kuva 4. Vesiputousmalli projektinhallintamenetelmänä [15].

Vesiputousmallin haittapuoli on se, että yhden osa-alueen kanssa työskenteleminen saattaa venyä liian pitkäksi, eikä aikataulu riitä jäljellä oleviin vaiheisiin. Mikäli suunniteltavassa ohjelmistossa tai toteutuksessa ilmenee ongelmia, jotkut ongelmat on jätettävä korjaamatta ja tuotantoprosessia on jatkettava eteenpäin. [15.]

3.2 Scrum

Ohjelmiston kehitykseen paremmin soveltuva projektinhallintamenetelmä on Scrum, joka valittiin Tietohallinnon vuorojärjestelmän parannusprojektissa käytettäväksi menetelmäksi. Kyseisen menetelmän mukaisesti projektin jokainen vaihe käydään yksityiskohtaisesti läpi ennen kuin pienikin osa-alue on tehty loppuun asti. Scrum on uusimpia projektinhallintamenetelmiä, ja sen on kehittänyt vuonna 1993 yhdysvaltalaiset Jeff Sutherland ja Ken Schwaber [16.] Scrumin toimintavaiheita kuvaillaan kuvassa 5. Keskeinen toimintaperiaate on, että muodostetaan ryhmä henkilöitä, jotka valvovat projektin vaiheet ja heidän joukostaan valitaan yksi henkilö ”Scrum-mestari”, joka kontrolloi syklien tilannetta ja katsoo, kuinka projektin läpiviemiseksi asetetut tehtävät toteutuvat. Hänen tehtävänä on johtaa projektia ja katsoa, että työskentely ryhmän sisällä on sujuvaa.



Kuva 5. Scrum-projektinhallinnan kaavio [17].

Suunniteltavalta tuotteelta vaadittavat ominaisuudet listataan taulukkomuotoiseen asiakirjaan, jota voidaan kutsua ”toteutuslistaksi” (engl. Product backlog). Siihen on listattuna ohjelmiston sisäisiä ominaisuuksia ja toimintoja, joita lopputuotteen odotetaan sisältävän. Tuotteen ominaisuudet on listattu tärkeysjärjestyksessä allekkain, jotta tärkeimmät asiat saadaan toteutettua huolellisesti. Ryhmän jäsenten kanssa keskustellaan yhdessä asetetuista tavoitteista ja asetetaan projektin tehtäville aikataulu, jota ryhmän jäsenten tulee noudattaa. [17; 18.] Metropolian Tietohallinnon vuorojärjestelmän parantamisprojektin työn aikatauluttamista varten tehtiin kuvan 6 mukainen toteutuslista, jossa kunkin tärkeysasteen tehtävän parissa työskentely oli rajoitettu kolmeen viikkoon.

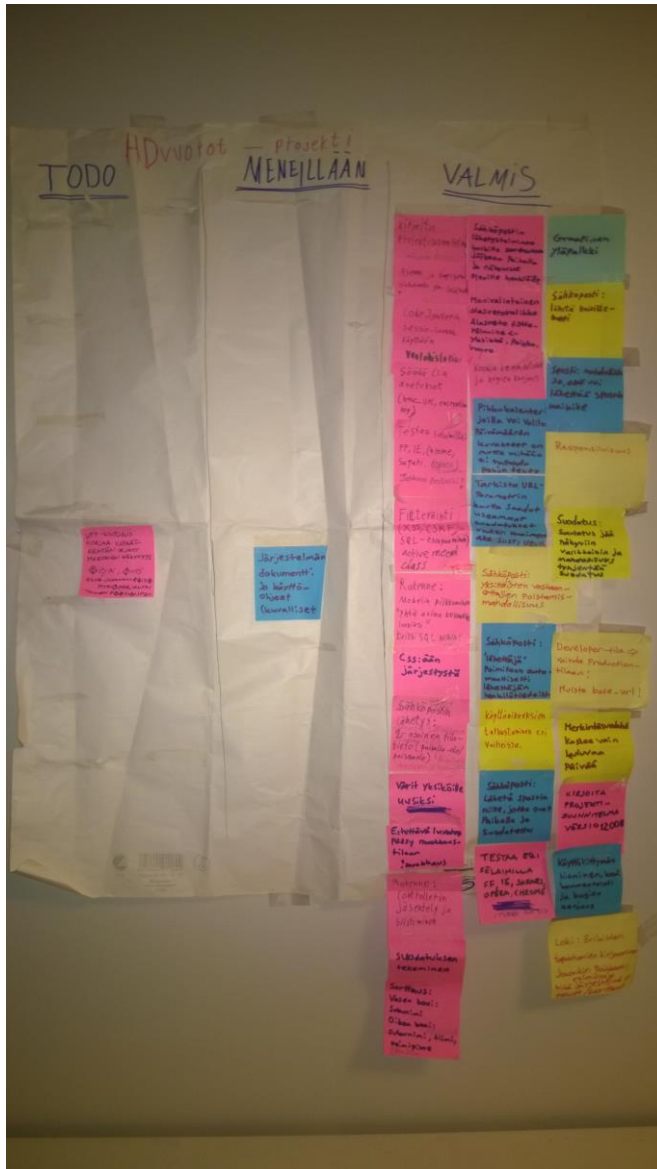
HDVUOROT PRODUCT BACKLOG							
VALMIS	AIKAIKKUNA	TÄRKEYSASTE	TEHTÄVÄ	VAIHE	VALMIS	TEHDYT TEHTÄVÄT (LKM)	Prosenttiosuus / osa
07.07.14	1	1	Codeigniterin oma sessio-luokka käyttöön.	Menilläään	1	8	
	1	1	Filterointi (mm. XSS, CSRF, SQL escapaaminen ja tai prepared statements) lisättävä kaikkiin input-kenttiin, sekä globaalisti järjestelmään.	Menilläään	1		
	1	1	Paranna rakennetta. Controllerin pilkkominen ja jäsentely. Modelin pilkkominen yhtä asiaa kuvaaviin tiedostoihin.	Menilläään	1		
	1	1	Kätkö SQL-komennot siirretään Modeliin.	Menilläään	1		
	1	2	Säädä codeigniterin asetukset (base_url, encryption_key).	Menilläään	1		
	1	2	Koodin kommentointi ja bugien korjaus.	Menilläään	1		100.00%
	1	2	Testaus eri selaimilla (FF, IE, Chrome, Safari).	Menilläään	1		
	1	3	Z osainen tilatelo s-postin lähehtykseen, paikallaolo merkintä / poissaolo merkintä erotus	Menilläään	1		
21.07.14	1	3	Kirjoita projektsuunnitelmaa.	Menilläään	1		
	2	1	Estettävä kuvien pääsy muokaus-tilaan.	Menilläään	1	7	
	2	1	Suodatuksen korjaaminen	Menilläään	1		
	2	2	CSS:ään järjestystä.	Menilläään	1		
	2	2	Koodin kommentointi ja bugien korjaus.	Menilläään	1		100.00%
	2	2	Sortaus. Vasen boksi päätetään sukunimi. Oirea listahäkymis sukunimi, nimi, toimipiste.	Menilläään	1		
	2	2	Sähköpostin lähehtymiento kaikille sortauksen jälkeen paikalla ja näkyvissä oleville henkilöille.	Menilläään	1		
	2	3	Monivalintainen alasvetovalikko. (Alasvetovalikon filteroinnille <- yksikkö, paikka, vuoro)	Menilläään	1		
04.08.14	3	2	Järjestelmän dokumentti ja käyttöohjeet (kuvalliset).	Menilläään	1	7	
	3	2	Käyttöliittymän hiominen, koodin kommentointi ja bugien korjaus.	Menilläään	1		
	3	2	Pikavalintatila jota voi valita päivämäärän, tuvatieset on mutta mitään ei tapahdu.	Menilläään	1		
	3	2	Testaus eri selaimilla (FF, IE, Chrome, Safari, Opera).	Menilläään	1		
	3	2	Sähköposti lähettäjä poimitaan automaattisesti lähettäjä henkilö tiedoista.	Menilläään	1		87.50%
	3	3	Värit yksiköille uusiksi.	Menilläään	1		
	3	3	Sähköposti suodatuksen ottaminen huomioon	Menilläään	1		
	3	4	Tarvitta URL-parametrin kautta saatut useammat suodatukset yhden avaimen alle. Siisti URL:ia.	Menilläään	1		
11.08.14	4	1	Käyttöoikeuksien tarkastaminen eri vaiheissa.	Menilläään	1	8	
	4	1	Loki	Menilläään	1		
	4	1	Sähköposti lähete kaikille-nappi	Menilläään	1		
	4	2	UTP-koodus vapaaapäivissä. korjaa vapaaapäivien ä ja ö näkyvyys oikein.	Menilläään	1		
	4	2	Sähköposti yksittäisten vastaanottajien poistamismahdollisuus	Menilläään	1		
	4	3	Merkintäsuodatus: koskee vain kuluva nykyistä päivää.	Menilläään	1		88.89%
	4	4	Responsiivisuus	Menilläään	1		
	4	4	Muuta base_url Codeigniter jäänyt developer-tilaan. Pitäisi olla production-tila.	Menilläään	1		
	4	4	Suodatus: suodatus ja näkyvät valitoin ja mahdollisuus tyhjentää suodatus.	Menilläään	1		
	4	5	(Käyttäjien yhteisen taulun väliaikainen tallentaminen palvelimelle jota useampi käyttäjä käyttää.)	Menilläään	1		
	4	5	(Paikoista ja yksiköistä dynaamisten taulujen tekeminen.)	Menilläään	1		
	5	1	Graafinen välpäkki	Menilläään	1	1	100.00%
						Kokonaisuuseteneminen:	31
						Prosenttiosuus:	93,94%

Kuva 6. Metropolian Tietohallinnon vuorojärjestelmän toteutuslista.

Scrum-menetelmän mukaisesti suoritettavat tehtävät, jotka tulee tehdä huolella esimerkiksi seuraavan kahden viikon aikana, merkitään kehitystaululle, jota voidaan kutsua sprinttitauluksi (engl. Sprint backlog). Toteutuslistalta siirretään työtehtäviä, jotka voidaan kirjoittaa pieniin muistilappuihin ja alkuvaiheessa laput laitetaan keskeneräisten tehtävien (engl. todo) kohdalle. [17; 18.] Kun tiettyä tehtävää ryhdytään suorittamaan, kyseinen lappu siirretään ”meneilläään”-sarakkeeseen. Valmiiksi saatujen osa-alueiden tehtävät siirretään ”valmis”-sarakkeeseen.

Projektissa jokaviikkoisen lyhyen palaverin aikana projektipäällikölle ja ryhmän jäsenille esiteltiin valmiiksi tehdyt tehtävät. Samalla esiteltiin vuorojärjestelmässä olevan toiminnon toimintaperiaate. Saatua palautte auttoi ymmärtämään edistyttiinkö siten, että ryhmä oli tyytyväinen tuloksiin.

Kehitysvaiheessa sprinttitaulussa käytettiin erivärisiä muistiolappuja, joiden tarkoituksena oli helpottaa ymmärtämään tehtävien tärkeysjärjestys. Kuvassa 7 näkyy Tietohallinnon vuorojärjestelmän muistilaput, jotka siirrettiin sarakkeelta toiselle projektin etenemisen mukaan. Vaaleanpunaisiin lappuihin kirjoitettiin aikaikkunassa ensimmäisenä olevien asioiden tehtävät, jotka oli määritelty tärkeimmiksi. Sinisiin lappuihin kirjoitettiin seuraavaksi tärkeimmät asiat. Keltaiset laput olivat viimeistelyä kaipaaviin asioita varten, joita ei saa unohtaa ennen projektin päättymistä.



Kuva 7. Tietohallinnon vuorojärjestelmän sprinttitaulu.

Kaikki asiat käytiin läpi yhdessä ryhmän henkilöiden kesken, ja jos todettiin, että asiat ovat kunnossa, niin siirryttiin seuraavaan vaiheeseen. Tässä tapauksessa siirryttiin erivärisen muistilappuun.

4 Toiminnot ja ominaisuudet

4.1 Vuorot

Metropolian Tietohallinnon vuorojärjestelmässä käytetään nimilistan kaltaista sivustoa, jossa on kalenterinäkymä kuukausittain. Näkymiä on kaksi erilaista, jossa toisessa näkymässä pääsee muokkaamaan vuoroja kirjaututtaessa järjestelmän ylläpitäjän admin-tunnuksilla, jotka ovat tässä tapauksessa samat kuin Metropolian Tuubi-portaalissa. Perusnäkymä avautuu kirjaututtaessa omilla Tuubi-käyttäjätunnuksilla ja silloin aukeaa oletuksena ainoastaan vuorojen katselutila, josta pääsee tarkistamaan nopeasti vuoronsa. Jokaisella työntekijällä on oma vuoronsa merkittynä taulukkomuotoiseen tauluun. Sivupalkissa hallinnoidaan vuoroja yksityiskohtaisemmin ja sinne siirrytään valitsemalla muokkaus-napista. Henkilöiden lisääminen järjestelmään ja heidät poistamisensa järjestelmästä tapahtuu pääasiassa sivupalkin kautta. Myös kalenterinäkymän aukeamaan pystyy lisäämään vuoron valitsemalla ensin rastin sen päivän kohdalle, johon kyseinen henkilö haluaa merkitä vuoronsa. Sitten valitaan kuukauden nimen yläpuolella olevasta pudotusvalikosta sopiva vuorokuvaus, esimerkiksi loma. Käyttäjä voi myös asettaa jonkun muun vuoron edellisen vuoron tilalle. Työntekijöiden tiedot ovat tallennettuna MySQL-tietokantaan, joka sijaitsee Metropolian palvelimella. Viikonloput ja muut vapaapäivät on merkitty harmaalla palkilla eikä niitä kenttiä pysty valitsemaan ja asettamaan vuoroa.

Helppokäyttöisyyden parantamiseksi vuorojärjestelmässä voi valita kerralla useita henkilöitä listalta ja asettaa vuoron tietyille päiville. Kalenterinäkymän kuukausia pystyy vaihtamaan edellisen ja seuraavan kuukauden näkymään. Työntekijöiden vuorohistoriaa voi tarkastella vuosi kerrallaan napsauttamalla henkilön nimeä listalta ja vasemmalle puolelle tulee näkyviin kuinka monta lomapäivää vuodessa hän on pitänyt. Kuva 8 esittää minkä näköinen on Tietohallinnon vuorojärjestelmän etusivu käyttäjän kirjautuessa sisään järjestelmään.

Kuva 8. Tietohallinnon vuorojärjestelmän etusivu.

CodeIgniterin sovelluskehys käyttää hyväksi SQL-tietokantaa, jossa tiedon käsittelyssä käytetään komentoja SELECT, WHERE, FROM, ORDER BY ja INSERT INTO. Vuorojärjestelmän tietokantaan on tallennettu taulut, joka sisältää työntekijöiden etunimen, sukunimen, työtunnuksen (henkID), käyttäjätunnuksen, roolinumeron nimikkeen ja puhelinnumeron. Vuorojen tyypit on tallennettu tauluun, jonka nimi on vuorot. Sairasloma, puhelinpäivystäjä, saldovapaa, loma, iltavuoro, helpdesk kiinni, valvontatukihenkilö, koulutuksessa ja puhelinvastaajan purku ovat näkyvissä palkkeina Tietohallinnon vuorojärjestelmän alaosassa eri väreinä. Tietokannasta haetaan oikea vuorotyyppi vuoroID:n avulla, jotka ovat numeroitu numeroin 1 - 12. Juhlapyhät, kuten helatorstai ja muut vapaapäivät, ovat taululla, jonka nimi on "vapaattaulu". Vapaiden lisääminen tehdään lomakkeella ja tiedot tallennetaan vapaattauluun. Taulun tiedot ovat merkitty samalla tavalla kuin vuorojen tyypit.

4.2 Suodatus

Tiedon haku nopeasti vähentää turhaa etsimistä, ja haluttu tulos saadaan näkyville hetkessä. Suodatustoiminto on vuorojärjestelmän yksi käytetyimmistä toiminnoista, jotta saadaan haettua haluttu työntekijä nimilistasta. SQL-kyselyillä muodostetaan rajaukset tietokannassa olevista tiedoista, jotta haluttu tietomäärä pystytään tulostamaan näytölle. Nimilistan järjestys voidaan määrittää nousevaksi tai laskevaksi ja lajittelujärjestyksenä on sukunimi, yksikkö ja toimipiste.

Haun tuloksia voidaan eritellä yksityiskohtaisemmin erilaisten kriteerien kautta, kuten kuvissa 9 ja 10 havainnollistetaan.



Kuva 9. Suodatus asetettujen valintojen perusteella.

Siirtyminen suodatuslaatikoiden välillä on suoraviivaista ja nopeaa, sillä käyttäjä voi valita haluamansa määrän valintoja jokaisesta alasvetovalikosta ja hyväksyä rajauksensa suodatusnapista. Tietohallinnon työntekijät voidaan suodattaa vielä tarkemmin toimipaikkojen, yksikön ja nimikkeen mukaan. Jokainen alasvetovalikko toimii itsenäisesti, hakutulos voidaan kohdistaa haluttaessa pelkästään toimipaikkoihin kuvan 10 osoittamalla tavalla. Merkintöiden suodattaminen koskee ainoastaan kyseistä päivää, jolloin vuorojärjestelmää käytetään. Tarkoituksena on täsmentää hakua, jossa nähdään, onko työntekijä paikalla vai esimerkiksi lomalla. Mikäli henkilöllä ei ole lainkaan merkintöjä, hän on töissä ja tavoitettavissa työpäivän aikana.



Kuva 10. Työntekijöiden rajaaminen toimipaikoittain.

Työntekijöiden rajaamista toimipaikkojen mukaan on helpotettu ja nopeutettu erivärisillä palkeilla sopien ennalta työntekijöiden kanssa, minkä värinen suodatuspainike

määrätyllä toimipaikalla on käytössä, kuten kuva 10 havainnollistaa toimintaperiaatteen. Visuaalisuuden lisääminen parantaa jäsentelyn kokonaisuuden hahmottamista ja tällöin erottelukyky paranee.

4.3 Lomakkeet

Henkilöiden ja vuorojen lisäämiseksi ja niiden muokkaamiseksi vuorojärjestelmän sivun vasemmalle puolelle on luotu tekstikenttiä. Käyttäjiä voidaan lisätä sivupalkissa sijaitsevalla lomakkeella, ”formilla”, johon syötetään vaadittavat tiedot uudesta käyttäjästä palvelimelle käyttäen POST-metodia. Työtunnus määräytyy automaattisesti järjestelmän antaessa sen silloin kun henkilötauluun lisätään uusi käyttäjä. Luotuaan uuden käyttäjätunnuksen itselleen, hän ei voi enää muuttaa työtunnustaan. Listalla olevat nimet näytetään sukunimen mukaan.

Käyttäjän poistaminen vuorojärjestelmästä tapahtuu valitsemalla henkilön nimi listalta ja vahvistamalla valinta. Poistetun käyttäjän oikeudet päästä vuorojärjestelmään on evätty, koska hänen käyttäjätunnuksaan ei löydy enää tietokannasta. Ainoa menetelmä käyttöoikeuden palauttamiseksi on pyytää, että joku tietokannasta löytyvä käyttäjä lisää poistetun käyttäjän uudelleen.

4.4 Sähköposti

Tiedonkulku yksinkertaisella tavalla ja tilanteiden pitäminen ajan tasalla on tärkeää, jotta saadaan tieto välitettyä nopeasti tietyille henkilölle tai käyttäjäryhmälle. Tietohallinnon vuorojärjestelmän sähköpostitoiminto on suunniteltu siihen tarkoitukseen, että paikalla oleville työntekijöille voidaan lähettää sähköpostia kuvan 11 esittämällä lomakkeella. Viestin aiheena voi olla lyhyt tärkeä tiedotus esimerkiksi tulevasta yhteisestä palaverista. Viestin sisällön kannattaa olla tyyliltään napakka ihan kuin viesti olisi suunnattu suuremmalle vastaanottajaryhmälle.

Kuva 11. Tietohallinnon sähköpostin lähetykslomake.

Sähköpostilomakkeessa olevien henkilöiden nimet haetaan automaattisesti palvelimella olevasta tietokannasta ja samalla katsotaan onko henkilö paikalla vai poissa esimerkiksi lomalla sinä päivänä, jolloin viesti halutaan lähettää. Vuorojärjestelmä lähettää oletuksena sähköpostiviestin paikalla oleville henkilöille, mutta viestin lähettäjä voi halutessaan poistaa valinnan tietyn henkilön nimen kohdalta. Poissaoleville työntekijöille ei lähetetä automaattisesti sähköpostia, koska he eivät ole merkittyinä tietokannassa paikalla oleviksi kyseisenä päivänä. CodeIgniterin ohjelmistokehykseen sisältyy kirjasto, joka mahdollistaa sähköpostin lähettämisen. Valmiiksi määritelty pohja sähköpostin lähettämiseen löytyy ohjelmistokehyksen järjestelmäkansiosta. Vaaditut parametrit, kuten sähköpostiosoite, SMTP-portin osoite, käyttäjätunnus ja salasana syötetään heittomerkkien väliin ja niihin valmiiksi määritelyihin kenttiin. [19.]

4.5 Loki

Järjestelmän ylläpitämiseen kuuluu jonkinlainen tietojen ja tapahtumien rekisterinhallintatiedosto. Lokitiedostossa säilytetään käytössä olevan järjestelmän viimeaikaiset suoritukset ja tapahtumat. Olennaisimmat tapahtumat, jotka merkitään lokitiedostoon, ovat vuorojen lisääminen ja niiden muokkaaminen.

Henkilöstä, joka on muokannut tietokantaa vuorojärjestelmää käyttäen tai kirjautunut ulos järjestelmästä, rekisteröidään lokitiedostoon hänen nimensä, käyttäjänimi, päiväys ja kellonaika. Tulevaisuudessa on helpompi saada selville, kuka on käynyt muokkaamassa tapahtumaa vuorojärjestelmän sisällä ja milloin se on tapahtunut. Tietoturvan kannalta rekisteritiedoista pystytään katsomaan, onko vuorojärjestelmää käyttänyt Metropolian Tietohallintoon kuulumaton henkilö vai onko järjestelmään tapahtunut odottamaton verkkohyökkäys.

5 Tietoturva

Tietoturvasta huolehtiminen on erittäin tärkeä asia, jotta palvelua tai sivustoa ei väärinkäytetä. Haavoittuvuuden paljastuessa rikollisille, salaiseksi luokitellut tiedot paljastuvat heille. Tietokantaan pääsy voidaan mahdollistaa syöttämällä haitallista ohjelmistokoodia ja tekemällä kyselyitä. Pahimmassa tapauksessa työntekijöiden luottamuksellisia tietoja paljastuu siten, että siitä koituu vahinkoa. Toisten henkilöiden nimien väärinkäytön ja identiteettivarkauden uhan vuoksi tärkeimpiä tietoja ei kannata tallentaa palvelimen tietokantaan ollenkaan.

Käyttäjän tunnistautumiseen käytettyjä menetelmiä on monenlaisia ja yleisimpiä niistä on käyttäjänimi-salasana-yhdistelmä. Metropolian sisäänkirjautumismenetelmänä käytetään kertakäyttökirjautumista (engl. Single Sign-On-Service (SSO)). [20.] Kuvassa 12 on vuorojärjestelmän sisäänkirjautumisikkuna, joka on samanlainen kuin Metropolian Tuubi-portaalissa. Käyttäjätunnuksia ei tarvitse olla monta, koska tunnistautuminen eli käyttäjän todennus suoritetaan nopeasti yksinkertaisella tavalla CAS-moduulin avulla.



Kuva 12. Metropolia Single sign-on service-kirjautumisikkuna.

6 Projektin toteutus

6.1 Suunnitteluvaiheet

Työ toteutettiin parityönä ja osaksi kesätyöpaikkaharjoitteluna kahtena peräkkäisenä kesänä, jotta ohjelmointi tulisi tutuksi molemmille harjoittelijoille. Tietohallinnon vuorojärjestelmä oli aikaisemmin toteutettu opiskelijoiden toimesta, mutta järjestelmän kokonaisrakenne oli hajanainen. Toiminnallisuus ja tietoturvallisuus oli huonosti toteutettu.

Työn aikataulu suunniteltiin taulukoihin käyttäen Microsoft Excel- taulukkolaskentaohjelmaa ja sovittiin projektipäällikön kanssa sähköpostin välityksellä palaverit pidettäväksi viikon välein. Tulevissa palavereissa tultiin käymään läpi jo suoritettuja töitä ja sen jälkeen suunniteltiin tulevat työtehtävät vuorojärjestelmän parissa. Projektin tuli olla kokonaisuudessaan valmiiksi saatettuna työsuhteen päättyessä. Keskusteltiin Metropolian Tietohallinnon henkilökunnan kanssa siitä, kuinka vuorojärjestelmää on ajateltu käytettäväksi ja minkälaisia ominaisuuksia siihen järjestelmään toivottaisiin. Sovittiin haastatteluajoja työntekijöiden kanssa, mutta ongelmaksi muodostui heidän kesälomiensa ajoitus. Suurin osa henkilöistä oli lomalla heinäkuun ajan, joten tapaamisajat täytyi järjestää alkukesällä tai elokuun alkupuolen aikana.

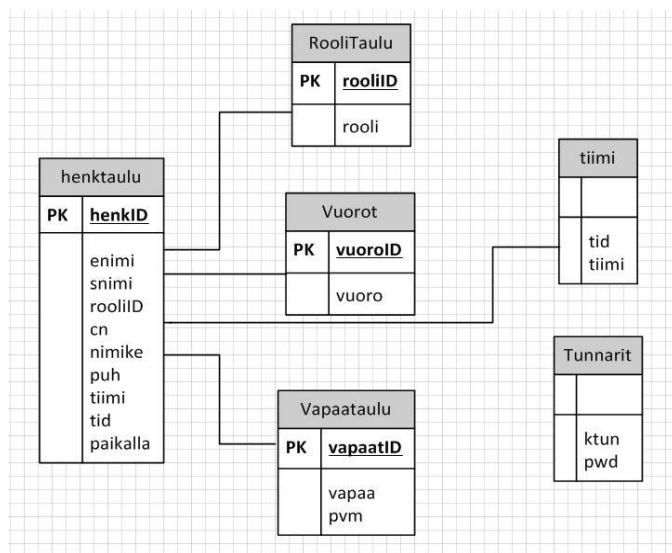
Tietohallinnon vuorojärjestelmän sisältö, rakenne ja toiminnallisuus täytyi ensiksi ymmärtää, ja sen takia luotiin dokumentaatiot ja vuokaaviot relaatiotietokannasta ja järjestelmän tiedostoista. Lainattiin Metropolian kirjastosta kirjallisuutta PHP-ohjelmointiin ja siihen tarvittavat materiaalit. Internetistä löytyy paljon hyviä oppaita ja esimerkkejä alkuun pääsemiseksi. Avoimen lähdekoodin kirjastoja voi käyttää hyödykseen katsomalla mallia ohjelmistokoodista ja muokkaamalla funktioita omaan projektiin.

6.2 Käytetyt menetelmät

Aikataulun suunnittelu tehtiin varhaisessa vaiheessa. Projektin alkuvalmisteluvaiheessa sovittiin tietyt asiat, sillä vuorojärjestelmään täytyi saada toteutettua erilaisia ominaisuuksia ja toimintoja noudattaen aikataulua. Dokumentaatioita laadittiin, jotta niitä pystyttäisiin hyödyntämään jatkokehityksessä. Aikataulun pitämiseksi kurissa käytettiin taulukkolaskentaohjelmaan merkittyjä työtehtäviä ja päivämääriä jolloin määrätyt asiat tulisi olla tehtynä tietyssä ajassa. Jälkimmäisenä kesänä käytettiin Scrum-menetelmää, jossa merkitään kaikki suunnitelmat ja edistymisprosentit alusta loppuun. [17; 21.] Kaikista dokumenteista ja vuorojärjestelmän eri versioista tehtiin varmuuskopio moneen eri paikkaan. Vuorojärjestelmän jokaisessa parantamisvaiheessa tehtiin järjestelmän tiedostoista kopio varotoimenpiteenä. Syy siihen oli, että mikäli huomattaisiin jonkin osa-alueen lakkaavan toimimasta, esimerkiksi vuorojen asettaminen jollekin päivälle ei onnistuisi, voitaisiin tutkia edellisestä versiosta mikä meni väärin.

6.3 Projektin suoritusvaiheet

Tietokanta toimii vuorojärjestelmän tärkeimpänä osa-alueena, kaikkien Tietohallinnon työntekijöiden yhteystiedot on tallennettu taulukkoihin kuvan 13 esittämällä tavalla. Niihin ei tarvinnut tehdä suurempia muutoksia, mutta vanhat tiedot täytyi päivittää manuaalisesti jos käyttäjätietojen hallintapalvelussa oli jokin häiriö.



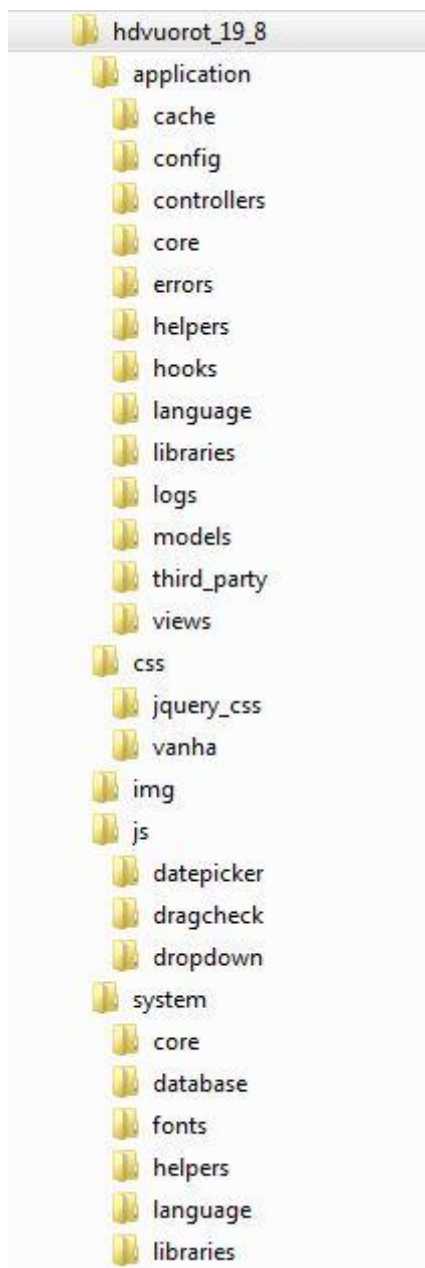
Kuva 13. Tietohallinnon vuorojärjestelmän tietokantarakenne.

Toimivuuden takaamiseksi kannattaa luoda jokin testauspalvelin, joka toimii väliaikaisena paikkana, josta järjestelmän vakautta testataan erilaisin menetelmin. Testausvaiheessa vuorojärjestelmään syötettiin erilaisia kyselyitä ja attribuutteja, jotta ehkäistään ohjelmistovirheiden esiintyminen selainversion käyttäjille.

Internet-selaimen URL-osoiterivin rakenne sisältää parametreja, joissa kerrotaan tiedon sijainti hakemistorakennepolussa. CodeIgniterin ohjelmistokehys tukee URL-osoiterivin yleisimpiä suojausmenetelmiä, kuten sisältäen escape-funktion. Näin ollen hakkeri ei pysty helposti aiheuttamaan SQL-injektiota. Muita turvallisuuteen liittyviä asetuksia CodeIgniterissä on salausavaimen määrittely (engl. encryption key) [22.] Jossain vaiheessa vuorojärjestelmässä oli vakava tietoturva-aukko, joka mahdollisti ulkopuolisten henkilöiden pääsyn muokkaamaan vuoroja. Myöhemmässä vaiheessa turvallisuutta parannettiin muodostamalla funktioita, jotka tarkistavat käyttöoikeuden erilaisissa tilanteissa. Vuorojärjestelmän autentikointia parannettiin yhdistämällä Metropolian Tuubin kirjautumisjärjestelmä osaksi järjestelmää. Työntekijöille tuli helpompi kirjautumismenetelmä, kun ei tarvita enää erillistä käyttäjätunnusta.

Vanhasta vuorojärjestelmästä kopioitiin joitakin osioita PHP-koodia uuteen järjestelmäversioon, CodeIgniterin sisältämiin tiedostoihin. Kuitenkin suurin osa funktioista ja toiminnoista on rakennettu uudelleen. Ohjelmistokoodin rakenteen tulisi olla loogisesti jäsennehtynä allekkain, kuten kuva 14 esittää ja mielellään kommentoituina. Sen vuoksi koodi kommentoitiin perusteellisesti. Jossain vaiheessa

kun joku uusi henkilö haluaa tehdä muutoksia tai parannuksia vuorojärjestelmään, hänellä on helpompi ymmärtää, mitä toimintoa järjestelmä suorittaa kyseisessä kohdassa.



Kuva 14. Vuorojärjestelmän hakemistorakenne.

Käytettävyyteen ja vastenopeuteen keskityttiin projektin jokaisessa vaiheessa. Palvelun tuli jatkuvasti vastata käyttäjän asettamiin pyyntöihin ja siirtyminen kuukausinäkymästä toiseen. Päivämäärien valitseminen valintanapeilla on työlästä, jos halutaan valita vaikkapa 16 päivää kuukaudesta ja asettaa sairausloma. Käyttäjätavallisempi ratkaisu

löytyi siitä, että asetetaan aloituspäivämäärä ja päättymispäivämäärä, jotka voidaan syöttää kirjoittamalla käsin tai Javascriptin jQuery-komponentilla toteutetulla päivämäärän valitsimella. [23.]

Ulkoasun tuli olla toteutettu mahdollisimman yksinkertaisella tavalla ja eri osien komponentit olisivat järkevästi sijoitettuna. Vuorojärjestelmän HTML-sivun rakenteen värimaailmaa uudistettiin radikaalisesti. Vanhan järjestelmän sininen tausta suunniteltiin ja muutettiin vastaamaan lähemmäksi Metropolian tämänhetkistä visuaalista ilmettä. Mobiililaitteet, kuten matkapuhelin ja tablettitietokone, ovat yleistyneet nopeaa tahtia. Vuorojärjestelmän pienet ruudut ja elementit täytyi saada mahtumaan pieneenkin tilaan ilman että tekstit menevät vaikealukuisiksi tai lukukelvottomaksi. Tyyli tiedostoa, CSS:ää muokattiin ja luotiin yksityiskohtaiset määritelmät jokaiselle vuorojärjestelmässä oleville komponenteille resoluutiosta riippumatta. Responsiivisuus tehdään lisäämällä määrittelyjä tyyli tiedoston loppuun siten, että jos näytön resoluution koko on esimerkiksi 1920 x 1080, vuorojärjestelmän kaikki elementit näkyvät mukaan lukien vasemmassa laidassa oleva sivupalkki. Resoluution ollessa 1024 x 768 sivupalkki menee piiloon ja suodatuspainikkeet siirtyvät siististi allekkain. [12.] Kaikilla verkkoselaimilla, kuten Mozilla Firefox, Microsoftin Internet Explorer, Opera, Googlen Chrome ja Macintoshin Safari, on erilaiset tavat tulkita vuorojärjestelmän lähdekoodia. Tästä johtuen sivun esitystapa saattaa olla jokaisessa selaimessa hieman erilainen.

Nimilistan suodattaminen ja lajitteluominaisuus vaati erilaisten ehtojen ja funktioiden syöttämistä, jotta saataisiin halutunmuotoinen lajittelujärjestys. Käytiin erilaisia vaihtoehtoja läpi ja keskusteltiin työntekijöiden kanssa, missä järjestyksessä he haluaisivat, että nimet näytetään.

Sähköpostin lähetystoiminnon toteuttaminen oli vuorojärjestelmän ominaisuuksista uusimpia ja siihen vaadittiin aika paljon uuden asian opettelemista. Viestilomake on toteutettu PHP-lomakkeella, johon on määritelty haettavaksi tietokannasta paikalla olevat työntekijöiden nimet ja suurin sallittu merkkimäärä, joka viestissä saa enintään olla. Liitetiedostoja voidaan lisätä sähköpostiviestiin, mutta tuetut tiedostomuodot ovat ainoastaan tekstitiedosto ja pdf-dokumentti. [19; 24.]

Graafinen suunnittelu kuului vuorojärjestelmän parantamisprosessiin, koska ensimmäiset ulkoasuluonnokset sivusta eivät miellyttäneet kaikkia työntekijöitä.

Yläpalkin väri, toimipaikkojen, vuorojen merkintävärit ja painonappien värit jouduttiin vaihtamaan moneen kertaan, kunnes lopullinen tulos miellytti käyttäjiä.

6.4 Käytetyt ohjelmistot

Metropolian Tietohallinnon vuorojärjestelmän parantamisessa käytettiin avoimella lähdekoodilla toteutettuja, maksuttomia web-ohjelmointiin suunnattuja ohjelmistoja. Tärkeimmät ohjelmistot ovat olleet XAMPP, Notepad++ ja Sublime Text 2, joiden avulla työskenneltiin CodeIgniterin ohjelmistokehyksen parissa. Ohjelmat voidaan ladata maksuttomasti internetistä, mutta kannattaa katsoa yhteensopivuus ja valita vakaa versio ohjelmasta.

Dokumentaation kirjoittamiseen koko projektin aikana käytettiin Microsoftin Office-tuoteperheen vakio-ohjelmistoja, kuten tekstinkäsittelyohjelmaa Wordia ja taulukkolaskentaohjelmaa Exceliä. Tietokantarakenteen kokonaiskuvan hahmottamiseen käytettiin Microsoftin Visio-nimistä ohjelmaa.

7 Yhteenveto

Insinööriyön tarkoituksena oli kehittää, luoda toimiva ja interaktiivinen vuoropalvelujärjestelmä, joka palvelee Metropolia Ammattikorkeakoulun Tietohallinnon työntekijöitä. Tavoitteena oli muodostaa looginen ja toiminnoiltaan helposti hahmoteltava järjestelmä käyttäen vähäisiä resursseja, kuten maksuttomia kehitystyökaluja.

Vuorojärjestelmän rakenteen uudelleensuunnittelussa kerättiin ja otettiin esille uusia ehdotuksia ja toiveita uusista toiminnoista, joiden tuli parantaa vuorojärjestelmän käytettävyyttä. Käyttäjien palaute oli kokonaisuuden kannalta erittäin tärkeä tuotantoprosessin aikana, koska sen avulla muutettiin vuorojärjestelmän toimintoja lähemmäksi heidän toiveitaan. Projektin aikana kävi ilmeiseksi, että CodeIgniterin ohjelmistokehys oli valmiina pakettina rakenteeltaan yksinkertainen ja kevyt eikä kuormittaisi palvelinta kovinkaan paljon. Aikataulun kontrolloiminen ja siinä pysyminen oli suhteellisen sujuvaa. Työ eteni hyvin kolmessa kuukaudessa kahtena kesänä. Välillä oli tiukkoja aikoja, jolloin projektin pienet osa-alueet olivat muutaman päivän

myöhässä. Pieniä viimeistelyparannuksia jäi tekemättä, mutta ne eivät ole kokonaisuuden kannalta merkityksellisiä.

Projekti oli osallistujille opettavainen kokemus, myös henkilökohtaisesti, sillä en ollut aikaisemmin ohjelmoinut yhtään www-sivustoihin liittyviä asioita. Oli aloitettava aivan alkeista opiskelemalla ensin perusteet PHP-ohjelmoinnin ja JavaScriptin toiminnasta. Ohjelmistokehyksiin ja niiden toimintaperiaatteisiin perehtyminen oli mielenkiintoista.

Kokonaisuudessaan vuorojärjestelmän parantaminen vastasi mielestäni odotuksiani. Sain toteutettua itselleni asetettuja tavoitteita ja samalla sain kokemusta www-sivujen tekemiseen liittyvistä asioista. Loppujen lopuksi työntekijöiltä saatujen kommenttien perusteella vuorojärjestelmä on toiminut hyvin tähän mennessä ja vastannut heidän toiveitaan. Nähtäväksi jää, kehitetäänkö järjestelmästä tulevaisuudessa päivitetty versio palvelemaan laajempaakin käyttäjäkuntaa.

Lähteet

- 1 Adrian. 2011. 30+ Most Useful Web Development Frameworks. Verkkodokumentti. Designmodo. <<http://designmodo.com/web-development-frameworks>>. Luettu 10.3.2015.
- 2 Mening, Robert. 2013. WordPress vs Joomla vs Drupal+ CMS “comparison chart”. Verkkodokumentti. Website Setup. <<http://websitesetup.org/cms-comparison-wordpress-vs-joomla-drupal>>. Luettu 20.4.2015.
- 3 Compare Custom PHP Frameworks and CMS. 2015. Verkkodokumentti. Colorwhistle. <<http://www.colorwhistle.com/compare-custom-php-frameworks-and-cms>>. Luettu 20.4.2015.
- 4 A Brief History of CodeIgniter. 2015. Verkkodokumentti. EllisLab.Inc. <<https://ellislab.com/codeigniter>>. Luettu 18.2.2015.
- 5 Skvorc, Bruno. 2011. Best PHP Frameworks for 2014. Verkkodokumentti. SitePoint Pty. Ltd. <<http://www.sitepoint.com/best-php-frameworks-2014>>. Luettu 17.3.2015.
- 6 CodeIgniter User Guide Version 2.2.0. CodeIgniter URLs. Verkkodokumentti. EllisLab.Inc. <<https://ellislab.com/codeigniter/user-guide/general/urls.html>>. Luettu 20.2.2015.
- 7 CodeIgniter User Guide Version 2.2.0. Verkkodokumentti. Application Flow Chart. EllisLab. Inc. <<https://ellislab.com/codeigniter/user-guide/overview/appflow.html>>. Luettu 20.2.2015.
- 8 CodeIgniter Rocks. Verkkodokumentti. CodeIgniter framework. EllisLab.Inc. <<http://www.codeigniter.com>>. Luettu 15.2.2015.
- 9 PHP Documentation and downloads. Verkkodokumentti. 2015. The PHP Group. <<http.php.net>>. Luettu 22.3.2015
- 10 About jQuery UI. Verkkodokumentti. 2015. <<http://jqueryui.com/about/>>. Luettu 20.3.2015.
- 11 CSS Tutorial. Verkkodokumentti. W3Schools. <<http://www.w3schools.com/css>>. Luettu 17.3.2015.
- 12 Responsive Web Design. 2014. Verkkodokumentti. Shay Howe. <<http://learn.shayhowe.com/advanced-html-css/responsive-web-design/#media-queries>>. Luettu 8.3.2015.

- 13 XAMPP Apache+MySQL+PHP+Perl. Verkkodokumentti. Apache Friends. <<https://www.apachefriends.org/index.html>>. Luettu 5.3.2015.
- 14 Ohjelmistokehitys ja erilaisia projektinhallintamenetelmiä. 2015. Verkkodokumentti. Suntubi.com. <<http://hybridimenetelma.suntuubi.com/?cat=9>>. Luettu 21.4.2015.
- 15 Definition: Waterfall Model. 2011. Verkkodokumentti. Waterfall Model. MBASchool.com. <<http://www.mbaskool.com/business-concepts/it-and-systems/8658-waterfall-model.html>>. Luettu 21.4.2015.
- 16 Scrum. 2015. Verkkodokumentti. Suntubi.com. <<http://hybridimenetelma.suntuubi.com/?cat=16>>. Luettu 21.4.2015.
- 17 Jones, Catherine. 2014. Scrum Methodology & it's Practical Use at CVCE. What is Scrum? Verkkodokumentti. Digital Humanities Lab at CVCE Hypotheses. <<http://cvcedhlab.hypotheses.org/54>>. Luettu 21.4.2015.
- 18 The Scrum Product Backlog. Verkkodokumentti. Product Backlog. International Scrum Institute. <http://www.scrum-institute.org/The_Scrum_Product_Backlog.php>. Luettu 21.4.2015.
- 19 CodeIgniter User Guide Version 2.2.0. Verkkodokumentti. Email Class. EllisLab. Inc. <<https://ellislab.com/codeigniter/user-guide/libraries/email.html>>. Luettu 19.2.2015.
- 20 Rouse, Margaret. single sign-on (SSO). Verkkodokumentti. TechTarget. <<http://searchsecurity.techtarget.com/definition/single-sign-on>>. Luettu 26.3.2015.
- 21 Scrum. 2015. Frequently Asked Questions. What is Scrum? Verkkodokumentti. Mountain Goat Software. <<http://www.mountaingoatsoftware.com/agile/scrum>>. Luettu 24.3.2015.
- 22 CodeIgniter User Guide Version 2.0. Verkkodokumentti. Encryption Class. EllisLab. Inc. <<https://ellislab.com/codeigniter/user-guide/libraries/encryption.html>>. Luettu 12.3.2015.
- 23 JQuery user interface. Verkkodokumentti. 2015. Datepicker. <<http://jqueryui.com/datepicker/>>. Luettu 21.3.2015.
- 24 CodeIgniter User Guide Version 2.2.0. Verkkodokumentti. File Uploading Class. EllisLab. Inc. <https://ellislab.com/codeigniter/user-guide/libraries/file_uploading.html>. Luettu 10.3.2015.
- 25 PHP-ohjelmointikielen uudet versiot korjaavat useita haavoittuvuuksia. 2015. Verkkodokumentti. Viestintävirasto.

<<https://www.viestintavirasto.fi/kyberturvallisuus/haavoittuvuudet/2015/haavoittuvuus-2015-036.html>>. Luettu 21.4.2015.

