



**LAUREA**  
AMMATTIKORKEAKOULU

*Uuden edellä*

# Laiterekisterin toteuttaminen

---

Rauvala, Riku

2011 Laurea Leppävaara

Laurea-ammattikorkeakoulu  
Laurea Leppävaara

## Laiterekisterin toteuttaminen

Riku Rauvala  
Tietojenkäsittelyn koulutusohjelma  
Opinnäytetyö  
Toukokuu, 2011

Riku Rauvala

### Laiterekisterin toteuttaminen

Vuosi 2011

Sivumäärä 40

---

Tämän työn aiheena oli luoda laiterekisteri, johon tallennetaan tiedot yrityksen käytössä olevista tietoteknisistä laitteista, kuten kannettavista ja pöytämallisista tietokoneista, verkkotulostimista sekä palvelimista. Tietoa oli yrityksen verkossa sijoitettu eri paikkoihin niin, että sen hallinta ja käyttö oli hankalaa.

Opinnäytetyössä käydään läpi yhtenäisen laiterekisterin suunnittelu, kehitys, testaus ja käyttöönotto alusta loppuun. Laiterekisteri toteutettiin PHP-kielellä ja tiedot ovat MySQL-tietokannassa. Näitä ohjelmistoja käytetään web-palvelimella, joka on asennettu Ubuntu Linux -käyttöjärjestelmään.

Suunnitteluvaiheessa käytiin läpi, mitä tietoja tietokantaan on oleellista tallentaa ja miten niitä hallitaan. Joitakin tarpeettomia, ylläpitoa hankaloittavia tietoja jätettiin pois. Tämän lisäksi valittiin palvelimen käyttöjärjestelmä ja ohjelmistot.

Varsinaisen toteutuksen osia olivat tietokanta ja sen rakenne, laiterekisterin ulkoasu sekä itse ohjelmakoodin ja toiminnallisuuden teko. Valmis toteutus siirrettiin testiympäristöstä palvelimelle, jossa sitä tullaan käyttämään.

Työ on tehty Bilia Oy Ab:n IT-osaston käyttöön. Tavoitteena oli rakentaa helppokäyttöinen ja tehokas rekisteri, joka kuitenkin sisältää tarpeellisen tiedon laitteiston ylläpitoa varten. Valmiin rekisterin tietokanta sisältää tiedot kaikista käytössä olevista laitteista.

Asiasanat: PHP, MySQL, tietokanta

Riku Rauvala

### Creating a device registry

Year 2011

Pages 40

---

The subject of this project is to create a registry of devices that contains information about the company's IT equipment, such as notebooks, desktops, network printers and servers. Initially, the information was located in several different places, which made managing and using it difficult.

This thesis contains the planning, development, testing and implementation phases of the registry throughout the project. The registry was created using the PHP scripting language, and the information is stored in a MySQL database. This software is run on a web server that has been installed on the Ubuntu Linux operating system.

At the planning stage it was reviewed what information is necessary to be stored in the database and how it is managed. Some unnecessary information that makes managing the database more complex was left out.

The parts of the actual implementation were the database and its structure, the user interface of the registry as well as the code and the functionality of the software. The completed work was transferred from the test environment into the server in which it will be used.

The registry was created for the IT department of Bilia Oy Ab. The purpose of the project was to build a practical and effective device registry that contains the necessary information that is required to maintain the devices. The database of the completed registry contains information about all of the devices that are in use in the company.

Key words: PHP, MySQL, database

## Sisällys

1	Johdanto.....	7
2	Tietoa kohdeyrityksestä .....	7
3	Projektin tausta ja tarkoitus.....	8
3.1	Lähtötilanne.....	8
3.2	Työn tyyppi.....	8
3.3	Käyttötarkoitus .....	9
3.4	Erot kaupallisiin toteutuksiin .....	9
3.5	Projektin tavoitteet ja työvaiheet .....	10
4	Käytetyt työkalut ja kehitysympäristöt .....	11
5	XAMPP-ohjelmistopaketti .....	11
6	Laiterekisterin sisältämät tiedot .....	12
6.1	Laitteen merkki, malli ja tyyppi.....	13
6.2	Laitteen nimi verkossa .....	13
6.3	Käyttäjä ja sijainti .....	13
6.4	Sarjanumero.....	14
6.5	IP-osoite .....	14
6.6	Lisätietokenttä .....	15
7	Tietoturva .....	15
7.1	Active Directory .....	15
7.2	Tunnistautuminen .....	16
8	PHP-tiedostot .....	17
9	Muut tiedostot .....	18
10	Tietokannan rakenne .....	19
11	Yhteys tietokantaan ja sen käsittely .....	21
12	Testaus.....	22
13	Laiterekisterin käyttöönotto .....	23
14	Tietojen siirto testiympäristöstä .....	23
15	Tietojen varmuuskopiointi .....	25
16	Laiterekisterin käyttö ja toiminta .....	27
16.1	Kirjautuminen ja etusivu.....	27
16.2	Tietojen listaus ja haku .....	29
16.3	Uuden laitteen tietojen lisäys .....	30
16.4	Muokkaus ja poisto.....	32
17	Yhteenveto .....	33
	Lähteet .....	35
	Kuvat .....	36
	Kuviot .....	37
	Taulukot .....	38

Taulukot .....	38
Liitteet.....	39

## 1 Johdanto

Kun yrityksessä on käytössä useita satoja työasemia ja muita tietoteknisiä laitteita, on laitteista oltava selkeät tiedot niiden hallintaa ja ylläpitoa varten. Tämän opinnäytetyön aiheena on yhtenäisen laiterekisterin toteutus yrityksen IT-osaston käyttöön.

Toimeksiantajana on Bilia Oy Ab, jossa työskentelen.

Työhön sisältyy sen kehitysympäristön ja alustan valinta sekä selvitys, mitä eri tietoja sinne on tarkoituksenmukaista tallentaa ja ylläpitää. Valmiin laiterekisterin toiminnan lisäksi on esitelty sen siirtäminen testaus- ja kehitysympäristöstä lopulliseen käyttöön omalle palvelimelleen.

## 2 Tietoa kohdeyrityksestä

Bilia Oy Ab on pääkaupunkiseudulla toimiva Volvo-henkilöautojen jälleenmyyjä. Automyynnin lisäksi Bilia huoltaa ja korjaa Volvo-, Renault- sekä Dacia-henkilöautoja sekä Renault -tavara-autoja. Yritys on perustettu 1. tammikuuta 1990.

Bilia on Volvo Auto Oy Ab:n 100-prosenttisesti omistama tytäryhtiö. Toimipisteitä on pääkaupunkiseudulla, ja ne sijaitsevat Vantaalla Kaivokselassa, Helsingissä Herttoniemessä sekä Espoossa Olarissa. Yrityksessä työskentelee noin 300 henkilöä, joista suurin osa Kaivokselassa.

Jokaisessa toimipisteessä toimii vaihtoautojen ja uusien Volvo-henkilöautojen myynti sekä valtuutettu Volvo-, Renault- ja Dacia-huoltokorjaamo. Lisäksi Kaivokselassa toimii vaurikorjaamo. Henkilöstö jakautuu prosesseittain kolmeen osaan: automyynnissä työskentelee henkilöstöstä 21,6 prosenttia, teknisessä palvelussa 67,8 prosenttia ja liiketoiminnan tukipalveluissa 10,6 prosenttia.

Bilialle on myönnetty ISO 14001 -ympäristösertifikaatti sekä ISO 9001 -laatusertifikaatti. Yritys kehittää jatkuvasti toimintojaan ympäristöä vähemmän kuormittavaksi yhteistyössä Volvo Auto Oy Ab:n kanssa, ja huolenpito ympäristöstä on yksi organisaation perusarvoista. Laatujärjestelmän tarkoitus on varmistaa prosessien laadun ja tehokkuuden jatkuva parantaminen.

### 3 Projektin tausta ja tarkoitus

#### 3.1 Lähtötilanne

Kävimme työpaikallani läpi kehitettäviä aihealueita, joista voisin löytää opinnäytetyöksi soveltuvan aiheen. Päädyimme siihen, että toteuttaisin yrityksen IT-osaston sisäiseen käyttöön tietoteknisiä laitteita varten laiterekisterin, jossa pidettäisiin tietoja kaikista käytössä olevista laitteista.

Yrityksen perustietotekniikka, sen ylläpito- ja tukipalvelut oli ulkoistettu usean vuoden ajan. Palvelusopimuksen päättyessä vuonna 2008 päädyttiin siirtämään nämä takaisin oman henkilöstön hallintaan.

Tästä hetkestä lähtien yrityksellä ei ole ollut käytössä yhtenäistä rekisteriä, johon olisi selkeästi dokumentoitu tiedot kaikista laitteista. Tietoja on ollut eri paikoissa vaihtelevilla yksityiskohdilla, esimerkiksi Excel-listauksissa ja laitetuimittajien omilla rekistereillä, ja tiedot ovat olleet osittain puutteellisia ja vanhentuneita. Tiedon etsiminen ja analysointi on tällä menetelmällä hidasta ja hankalaa, eikä selkeää kokonaiskuvaa yrityksen laitteistokannasta saa muodostettua. Esimerkiksi yksittäisen laitteen tietoja paikannettaessa ei aluksi välttämättä edes tiedä, mistä paikasta niitä lähtisi etsimään, ja lopulta löytyvä tieto saattaa olla vanhentunutta tai puutteellista.

Yrityksellä oli siis tarve laiterekisterille, johon saisi tallennettua tiedot sen käytössä olevista atk-laitteista. Niistä on saatavilla useista kaupallisista toteutuksista, joiden hinta on kuitenkin suhteellisen korkea. Niissä on myös paljon ominaisuuksia, joille yrityksen IT-osastolla ei ole käyttöä. Näitä ovat esimerkiksi laitteiden automaattinen etsiminen yrityksen sisäverkosta ja yksityiskohtaisten tietojen haku näistä, kuten listaus työasemille asennetuista ohjelmista. Parhaimmaksi vaihtoehdoksi osoittautui, että toteutan yksinkertaisen ja manuaalisesti päivitettävän laiterekisterin itse.

#### 3.2 Työn tyyppi

Opinnäytetyö on tyypiltään toimintakeskeinen työ. Se on tuotesuunnitteluhanke yrityksen sisällä, ja sen tulos tulee olemaan yrityksen apuvälineenä todellisessa käytössä.



### 3.3 Käyttötarkoitus

Yksi oleellisimmista laiterekisterin käyttötarkoituksista on yksittäisen laitteen paikantaminen ja sen tietojen katselu tuki- tai vikatilanteessa. Esimerkiksi kun yksittäinen yrityksen työntekijä ilmoittaa työasemassaan ilmenneestä ongelmasta, saadaan toimivasta laiterekisteristä haettua käyttäjän nimen perusteella hänen käytössään olevien laitteiden tiedot. Näin säästyy huomattavasti aikaa ja vaivaa verrattuna siihen, että laitteiden tietoja jouduttaisiin selvittämään joka kerta erikseen.

Jos jokin laite rikkoutuu, rekisterin avulla voi tarkistaa, onko siinä takuuta jäljellä, jolloin siihen saa tilattua korjauksen maksutta. Jos takuu on loppunut, voidaan miettiä, kannattaako sitä yrittää korjata, vai onko järkevämpää hankkia kokonaan uusi laite. Myös uusia laitehankintoja suunniteltaessa nähdään, paljonko eri-ikäisiä laitteita on käytössä, ja mitkä mahdollisesti tarvitsisi jo uusia.

Ajantasainen ja toimiva laiterekisteri helpottaa ja nopeuttaa koko IT-osaston päivittäistä työntekoa ja rutiineja. Tietoa ei tarvitse etsiä useasta eri paikasta, vaan kaikki tarvittava tieto on helposti saatavilla. Loppukäyttäjiltä ei tarvitse kysellä heidän kannaltaan hankalia kysymyksiä, vaan IT-osaston työntekijät löytävät vastaukset itse.

### 3.4 Erot kaupallisiin toteutuksiin

Laiterekisteri pienimuotoisena yhden henkilön projektina toteutettuna jää luonnollisesti selvästi suppeammaksi kuin yritysten myymät laiterekisterisovellukset, joita kehittää kokonainen joukko ihmisiä. Esimerkiksi joihinkin laiterekistereihin sisältyy ominaisuus, jossa sovellus tutkii määritettyä verkkoaluetta jatkuvasti, osaa automaattisesti havaita uuden verkkoon lisätyn laitteen ja hakea siitä lisätietoja. Näitä tietoja ovat esimerkiksi tarkka listaus työasemalle asennetuista ohjelmistoista, koneelle viimeksi kirjautunut käyttäjä, laitteen nimi ja nykyinen IP-osoite verkossa sekä tietoja työaseman osista, kuten suorittimen nopeus, muistin määrä ja käytössä olevan ja vapaan tilan määrä kovalevyllä.

Yrityksen työasemaympäristö on melko laajalti vakioitu, joten näiden ominaisuuksien tuoma hyöty hankintahintaan verrattuna jää melko pieneksi. Yrityksessä on siis käytössä muutamia eri laitemalleja, joiden kokoonpano on identtinen, ja käytössä olevat ohjelmistotkin ovat suureksi osaksi samoja joitakin lisäohjelmia lukuun ottamatta.

Myöskään verkon automaattinen tutkiminen ja laitteiden etsiminen ei ole erityisen tarpeellinen ominaisuus, sillä kaikki uudet laiteasennukset ovat tarkoin harkittuja ja

suunniteltuja, eikä tietojen dokumentointi tehtävien asennusten yhteydessä tuo kovin suurta lisätyömäärää yrityksen IT-osastolle.

Kaupallisissa laiterekistereissä on myös mahdollisuus erilaisten raporttien luomiseen, esimerkiksi kuinka moneen työasemaan tietty ohjelmisto on asennettu tai listaus tietyllä komponentilla tai käyttöjärjestelmällä varustetuista työasemista. Tämäkään ominaisuus ei ole tarpeellinen vakioidusta ympäristöstä johtuen.

### 3.5 Projektin tavoitteet ja työvaiheet

Tämän projektin tavoitteena on luoda mahdollisimman selkeä, hyvin toimiva ja kaikki oleelliset tiedot sisältävä laiterekisteri yrityksen tietoteknisistä laitteista. Selvitän, mitkä laitteiden yksityiskohtaisista tiedoista on hyvä sisällyttää rekisteriin, ja mitä voi mahdollisesti jättää pois.

Työn valmistuttua rekisteri jää jatkuvaan käyttöön, ja siitä on hyötyä koko IT-osastolle. Rekisteriä myös kehitetään jatkuvasti paremmaksi ja tarpeita vastaavaksi. Kokonaiskuva tietoteknisistä laitteista selkeytyy.

Alun suunnittelun ja vaatimusmäärittelyn jälkeen aloitan työn teknisen toteutuksen. Kun ensimmäinen käytettävä versio on valmis, otan sen useamman henkilön testattavaksi. Lopullinen työ siirretään pois testiympäristöstä ja otetaan käyttöön sille tarkoitettulla palvelimella.



Kuvio 1: Toteutusprosessin vaiheet

#### 4 Käytetyt työkalut ja kehitysympäristöt

Laiterekisteri toteutettiin PHP-ohjelmointikielellä, ja tiedot on tallennettu MySQL-tietokantaan. PHP (HyperText Preprocessor) on WWW-palvelimelle asennettava ohjelmisto, joka tulkitsee PHP-ohjelman. PHP tulkitsee koodin vasta sen suoritusvaiheessa, ja sen palauttama tulos palautuu asiakkaalle (yleensä WWW-selaimelle) esitettäväksi.

PHP-kieli on korkean tason kieli. Tällä tarkoitetaan sitä, että ohjelmoinnissa käytettävät käsitteet ovat ihmisen tuntemia, eikä esimerkiksi tietokoneen toimintaa ohjaavia käskyjä tarvitse tuntea (Kolehmainen 2006, 3). Se on avoimen lähdekoodin kieli, joka soveltuu erityisesti web-kehitykseen, ja sen voi sisällyttää HTML-kieleen (The PHP Group 2011).

SQL on lyhenne sanoista Structured Query Language, ja se on kieli, jolla käsitellään relaatiotietokantoja. MySQL on hallintajärjestelmä SQL-tietokannoille. Se sisältää myös joitakin omia funktioita SQL-funktioiden lisäksi (Kolehmainen 2006, 285-286).

Laiterekisteriä käytetään WWW-selaimella, ja sen käyttöliittymä toteutetaan HTML-kielellä (Hypertext Markup Language) sekä CSS-määrittelytiedoilla (Cascading Style Sheets). PHP-ohjelma on tekstitiedosto, jonka tiedostotunnus on PHP. Yksittäinen tiedosto sisältää sekä HTML-kuvauskieltä että PHP-ohjelmakoodia. Ohjelmakoodi on kirjoitettu tavallisella tekstieditorilla, Windows-käyttöjärjestelmään sisältyvällä Muistiolla.

#### 5 XAMPP-ohjelmistopaketti

PHP vaatii toimiakseen myös WWW-palvelimen. PHP, MySQL sekä WWW-palvelinohjelmistot ovat kaikki yksittäisiä ohjelmistoja, jotka täytyisi normaalisti asentaa ja konfiguroida erikseen.

XAMPP on WWW-palvelinpaketti, joka sisältää kaikki nämä ohjelmistot yhtenä pakettina. Siitä on versiot useille eri käyttöjärjestelmille: Windows (2000, 2003, XP ja Vista), Linux (testattu SuSE-, RedHat-, Mandrake- ja Debian-jakeluilla), Mac OS X sekä Solaris. (Seidler 2011.)

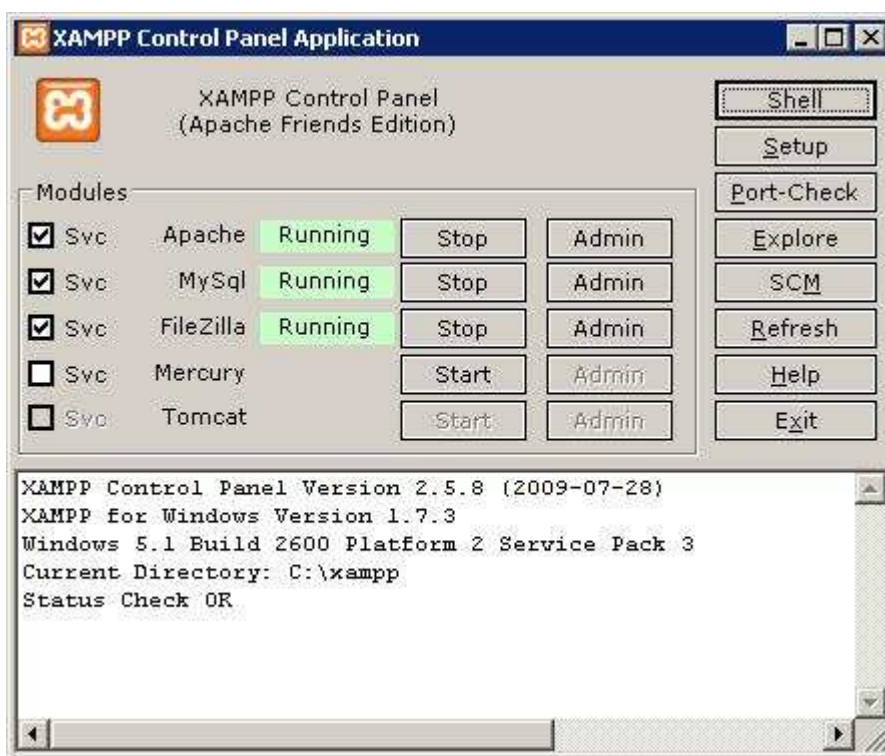
Laiterekisterin kehitysympäristöksi valitsin XAMPP:n Windows-version, joka sisältää seuraavat osat: Apache WWW-palvelin, MySQL, PHP, phpMyAdmin, Perl, FileZilla FTP-palvelin sekä Mercury-sähköpostipalvelin.

Apache HTTP Server on laajalti käytössä oleva WWW-palvelinohjelmisto, jota kehittää Apache Software Foundation. Se on ollut Internetin suosituin web-palvelinohjelmisto vuodesta 1996 lähtien (The Apache Software Foundation 2011).

phpMyAdmin on WWW-selainen kautta käytettävä, myös PHP-kielillä toteutettu MySQL-tietokannan hallintajärjestelmä. Se nopeuttaa ja helpottaa tietokannan hallintaa, sillä sen avulla voi käsitellä tietokantoja graafisella käyttöliittymällä MySQL:n tarjoaman komentorivin sijaan. Se on käännetty 62 eri kielelle, mukaan lukien suomeksi. (phpMyAdmin 2011.)

FTP (File Transfer Protocol) on tiedostonsiirtomenetelmä kahden tietokoneen välille. Sillä voi siirtää esimerkiksi PHP-tiedostoja henkilökohtaisen tietokoneen ja WWW-palvelinkoneen välillä. FileZilla FTP Server on FTP-palvelinohjelmisto.

Alkuvaiheessa laiterekisterin suunnittelua, kehitystä ja testausta varten XAMPP-paketti asennettiin tavalliselle työasemalle, jossa on käyttöjärjestelmänä Windows XP Professional. XAMPP asentuu yhdestä asennuspaketista. Asennuksen jälkeen määritetään tarvittavat salasana, sekä käyttöön otettavat ohjelman osat. XAMPP-paketti sisältää myös ohjauspaneelin, josta voi tarkistaa eri sovellusten tilan:



Kuva 1: XAMPP-ohjauspaneeli

## 6 Laiterekisterin sisältämät tiedot

Työn alkuvaiheessa käytiin läpi, mitä tietoja laitteista tulisi syöttämään laiterekisteriin, sekä miten ne järjestellään ja esitetään. Kaikkia mahdollisia yksityiskohtia ei tähän rekisteriin

haluttu, sillä tiettyjä yksityiskohtia jouduttaisiin edelleen pitämään yllä muissa sijainneissa. Laiterekisterin käyttö tuli olla nopeaa ja yksinkertaista, ettei ylläpitoon kuluisi liikaa aikaa.

Näiden tietojen perusteella suunniteltiin tietokannan rakenne, eli se, kuinka paljon, minkälaisia ja millä tavalla yhdistettyjä kenttiä tietokanta sisältää. Tämän lisäksi alkuvaiheessa hahmoteltiin, minkälainen WWW-selaimella oleva käyttöliittymä tulee olemaan. Tähän sisältyy sivun rakenne ja ulkoasu väreineen, linkkien, navigoinnin, sisällön ja tietokenttien sijainti sekä mahdollinen grafiikka.

Kaikkia laiterekisterin sisältämiä tietoja täytyy pystyä muokkaamaan helposti sen käyttöliittymästä. Tähän sisältyy uusien laitteiden tietojen lisäys, olemassa olevien tietojen muokkaus sekä laitteiden poisto tietokannasta. Tämän lisäksi laitteita etsitään tietokannasta erilaisilla hakukriteereillä, joista oleellisimmat ovat laitteen nimellä ja käyttäjän nimellä haku.

#### 6.1 Laitteen merkki, malli ja tyyppi

Laitteista merkitään laitteen valmistajan nimi ja malli. Vakioidussa ympäristössä osataan jo tämän perusteella nopeasti kertoa, minkälaisesta laitteesta on kyse (esimerkiksi minkä ikäinen laite on, ja kuinka suorituskykyinen se on laitteistoltaan). Tämän lisäksi merkitään erikseen minkä tyyppinen laite on kyseessä. Eri laitetyppejä ovat esimerkiksi pöytätietokone, kannettava tietokone, palvelin ja verkkotulostin.

#### 6.2 Laitteen nimi verkossa

Jokaisella yrityksen verkkoon liitettyllä laitteella on yksilöllinen nimi verkossa. Selkokielisen nimen avulla laitteet on helpompi tunnistaa, ja niihin saa myös yhteyden tämän nimen avulla. Esimerkiksi tukitilanteessa voidaan etsiä rekisteristä työasema käyttäjän nimen perusteella, ja työasemaan saadaan muodostettua yhteys laitteen nimellä.

Yrityksen Active Directory -toimialueella on käytössä Server 2003 -palvelimet. Pääasiallinen nimenselvityspalvelu tässä käyttöjärjestelmässä ja sen uudemmissa versioissa on DNS (Domain Name System) (Microsoft 2003). DNS-palvelun avulla verkon laitteisiin saadaan yhteys helposti, kun pelkkä laitteen nimen tietäminen riittää ja osoitteen selvittäminen tapahtuu automaattisesti.

#### 6.3 Käyttäjä ja sijainti

Yrityksellä on kolme eri toimipistettä: Kaivoksela, Herttoniemi ja Olari. Laitteesta merkitään rekisteriin toimipiste, jossa se sijaitsee, sekä mahdollisesti sen käyttäjä tai tarkempi sijainti. Useimmissa tapauksissa laitteen käyttäjä riittää määrittelemään sen sijainnin. Joillakin laitteilla, esimerkiksi verkkotulostimilla, taas ei ole tiettyä käyttäjää, joten niistä voidaan merkitä tarkempi sijainti.

#### 6.4 Sarjanumero

Jokaisella laitteella on valmistajan määrittämä yksilöllinen sarjanumero. Myös tämä tieto sisällytetään rekisteriin. Sarjanumeroa tarvitaan useimmiten silloin, kun laitteeseen tulee jokin vika ja sitä varten täytyy tilata huolto. Sarjanumeron avulla voidaan myös useimmiten tarkistaa, onko laitteessa takuuta jäljellä.

#### 6.5 IP-osoite

Suurin osa yrityksen verkon työasemista saa IP-osoitteen automaattisesti DHCP-protokollan (Dynamic Host Configuration Protocol) avulla. DHCP on protokolla, joka antaa asetusparametreja verkon tietokoneille. Sen avulla on mahdollista määrittää asiakaskoneen IP-osoite, aliverkon peite ja yhdyskäytävä sekä tarvittaessa myös esimerkiksi DNS- ja WINS-palvelimet (Casad & Willsey 1999, 280).

Käynnistyessään verkon asiakaskone lähettää yleislähetystenä DHCP-etsintäsanoman (discover) paikallisessa verkossa. Verkossa oleva DHCP-palvelin vastaanottaa etsintäsanoman, ja palauttaa vastauksena tarjoussanoman (offer). Tarjoussanoma sisältää asiakaskoneelle tarjottavan IP-osoitteen ja aliverkon peitteen DHCP-palvelimeen määriteltyjen osoitealueiden mukaisesti. Vastaanotettuaan tarjoussanoman asiakaskone muodostaa sen perusteella pyyntösanoman (request). Kun palvelin on saanut pyyntösanoman, se vahvistetaan erillisellä vahvistussanomalla (acknowledgement). Vahvistussanomassa on mukana oletusyhdyskäytävä sekä DNS- ja WINS-palvelimien osoitteet. Tämän lisäksi siihen on määritetty aika, jonka osoitelaina on voimassa. (Casad & Willsey 1999, 281-282.)

DHCP helpottaa suuren työasemamäärän ylläpitoa huomattavasti. Laitteille ei tarvitse määritellä verkon osoitetietoja yksitellen, vaan ne saadaan automaattisesti. Hyöty on erityisen suuri kannettavia tietokoneita käyttäessä, sillä yrityksen eri toimipisteissä on käytössä eri osoitealueet. Ilman DHCP:tä asetuksia joutuisi muuttamaan joka kerta liikuttaessa toimipisteiden välillä.

DHCP:n ollessa käytössä IP-osoite ei siis ole laitteilla pysyvä, vaan se vaihtelee sen mukaan, miten palvelin lainaa osoitteita määritellyltä alueelta. Osoitteita siis ei ole näissä tapauksissa

järkevää merkitä rekisteriin. Joillakin laitteilla, esimerkiksi palvelimilla ja verkkotulostimilla, IP-osoite kuitenkin on kiinteästi määritetty, eikä DHCP ole niissä käytössä. Tämän takia myös IP-osoitteelle on hyvä olla oma tietokenttänsä rekisterissä.

## 6.6 Lisätietokenttä

Joihinkin laitteisiin voi liittyä erityistä lisätietoa, esimerkiksi jos työaseman kokoonpano on muutettu jälkikäteen, kuten asennettu uusi näytönohjain. Tätä varten rekisterissä on erillinen vapaa tekstikenttä lisätiedoille ja kommentteille.

## 7 Tietoturva

Laiterekisteri on käytettävissä vain yrityksen sisäisessä tietoverkossa. Verkko on yhteydessä internetiin, mutta se on suojattu ulkopuolisilta yhteyksiltä palomuurilla. Laiterekisterin palvelin saa siis käyttöönsä sisäverkon IP-osoitteen, ja siihen voi muodostaa vain yrityksen sisäisessä verkossa olevalta työasemalta. Internetistä ei ole suoraa pääsyä laiterekisterin palvelimelle.

IT-osastolla on myös käytössä VPN-yhteys (Virtual Private Network). Tämä tarkoittaa, että yrityksen sisäverkkoon on mahdollista muodostaa turvallinen yhteys myös ulkopuolisesta verkosta. Tällä tavoin pääsy laiterekisteriin on myös esimerkiksi työntekijän kotoa, kun käytössä on kannettava tietokone.

### 7.1 Active Directory

Verkossa on käytössä Microsoft Active Directory -hakemistopalvelu, johon jokaisella työntekijällä on henkilökohtainen käyttäjätunnus ja salasana. Active Directory sisältää tietoja yrityksen verkon tietokoneista, palveluista, käyttäjistä ja käyttöoikeuksista. Sen avulla käyttäjät ja tietokoneet pääsevät käsiksi yrityksen verkon resursseihin. Näitä resursseja ovat esimerkiksi tiedostot, tietokannat, ohjelmat ja sähköposti. (Holme, Ruest & Ruest 2008, 3.)

Active Directory -ympäristössä käyttäjien tunnistautumiseen käytetään Kerberos-nimistä protokollaa. Kun käyttäjä tai tietokone kirjautuu verkkoon, Kerberos varmentaa tämän käyttöoikeudet. Tämän varmennuksen jälkeen käyttäjällä tai tietokoneella on pääsy niihin resursseihin, joihin tälle on määritetty käyttöoikeus. (Holme, Ruest & Ruest 2008, 3.)

## 7.2 Tunnistautuminen

Vaikka laiterekisteriin pääsy on mahdollista vain yrityksen omasta sisäverkosta, ei sitä voi kuitenkaan pitää täysin avoimena, sillä tällöin kaikilla työntekijöillä olisi mahdollisuus päästä katselemaan ja muokkaamaan rekisterin tietoja. Laiterekisterin käyttöön vaaditaan siis käyttäjätunnus ja salasana.

Eritasoisia käyttöoikeuksia laiterekisteriin ei ole tarvinnut määrittää. Kaikki, joilla rekisteriin on pääsy, pääsevät muokkaamaan kaikkia sen sisältämiä tietoja.

Laiterekisterien käyttäjien tunnistamiseen käytetään olemassa olevia Active Directory -käyttäjätunnuksia. Yhteys laiterekisteristä Active Directoryyn on toteutettu avoimen lähdekoodin PHP-luokan, adLDAP:n, avulla. Tämän luokan avulla voidaan ottaa yhteys toimialueen ohjauskoneeseen ja varmentaa käyttäjä.

Tunnistautumiseen käytettävä adLDAP ei vaadi erityistä asennusta. Se on tehty yhteen PHP-tiedostoon, johon viitataan, kun sitä halutaan käyttää. Tähän tiedostoon täytyi määrittää toimialueen ja sen ohjauskoneen tiedot sekä käyttäjätili, jolla on tarvittavat käyttöoikeudet. Tämän lisäksi PHP:n määrittämistiedostoa piti muuttaa niin, että LDAP-lisäosa otettiin käyttöön.

Kirjautumisessa on määritetty pääsy laiterekisteriin vain IT-osaston työntekijöiden käyttäjätunnuksille. Kirjautumissivulla on lomake, jossa on kentät käyttäjätunnukselle ja salasanalle. Käyttäjän syöttämät tiedot lähetetään toiselle PHP-tiedostolle, joka ensin tarkistaa, onko käyttäjä sallittujen käyttäjien listassa, joka sisältää IT-osaston henkilöstön käyttäjätunnukset. Tämä lista on tehty PHP:n array-funktiolla:

```
$kayttajat = array("kayttaja1", " kayttaja2", " kayttaja3", " kayttaja4", " kayttaja5");
```

Jos käyttäjätunnus on listassa, kutsutaan adLDAP-luokka:

```
if ($kayttaja!=NULL && $kayttaja!=NULL && in_array($kayttaja,$kayttajat)){
include ("adLDAP.php");
$adldap = new adLDAP();
```

adLDAP ottaa yhteyden toimialueen ohjauskoneeseen ja tarkistaa, vastaavatko syötetty käyttäjätunnus ja salasana palvelimelle määritettyjä. Tähän käytetään funktiota authenticate (Barnett & Hyland 2011). Funktioon määritellään lomakkeelta saatu käyttäjätunnus ja



salasana. Jos tunnukset ovat oikein, kirjautuu käyttäjä sisälle laiterekisteriin. Muussa tapauksessa seuraa virheilmoitus.

Kun käyttäjä kirjautuu sisään onnistuneesti, luodaan tälle PHP-istunto (session). Istunnon ollessa voimassa laiterekisteriä voi käyttää. Laiterekisteri koostuu useista PHP-tiedostoista, ja jokaiseen tiedostoon on määritetty vaatimus siitä, että istunto on voimassa. Jos tiedostoa yrittää käyttää ilman voimassaolevaa istuntoa, ohjataan käyttäjä virhesivulle.

Laiterekisteristä uloskirjautuminen tuhoaa istunnon, ja uusi kirjautuminen vaaditaan.

Tietoturvan takia uloskirjautuminen onkin suotavaa, kun käyttäjä on lopettanut laiterekisterin käytön.

## 8 PHP-tiedostot

Laiterekisterin varsinainen toiminnallisuus ja tietokannan käsittely on sisällytetty PHP-tiedostoihin, jotka sijaitsevat WWW-palvelimen hakemistossa. Seuraavassa taulukossa on eritelty PHP-tiedostot ja niiden käyttötarkoitukset.

Tiedosto	Käyttötarkoitus
adLDAP.php	adLDAP-kirjasto
alku.php	Laiterekisterin etusivu, joka näkyy onnistuneen sisäänkirjautumisen jälkeen
auth.php	Tämä PHP-tiedosto on sisällytetty kaikkiin niihin tiedostoihin, jotka vaativat kirjautumisen. Tiedosto tarkistaa, onko istunto voimassa, ja jos ei ole, ohjaa käyttäjän ulos sivulta.
haku.php	Tietojen haku laiterekisteristä erilaisilla hakuehdoilla
index.php	Laiterekisterin alkusivu, johon syötetään kirjautumistiedot
lisaa.php	Tietojen lisäys tietokantaan
lisaa_check.php	Tarkistaa, onko kaikki pakolliset tiedot täytetty, näyttää käyttäjälle hänen syöttämänsä tiedot ja pyytää vahvistuksen
lisaa_do.php	Tekee varsinaisen lisäyksen tietokantaan, kun käyttäjä on hyväksynyt sen
listaus.php	Listaa koko tietokannan sisällön
loggaa.php	Tarkistaa index.php -tiedostosta ohjatut

	sisäänkirjautumistiedot adLDAP-kirjaston avulla ja ohjaa käyttäjän eteenpäin
logout.php	Kirjaa käyttäjän ulos laiterekisteristä ja tuhoaa istunnon
muok.php	Olemassa olevien tietokannan tietojen muokkaus
muok_check.php	Tarkistaa, onko kaikki pakolliset tiedot täytetty, näyttää käyttäjälle hänen syöttämänsä tiedot ja pyytää vahvistuksen
muok_do.php	Tekee varsinaisen muokkauksen tietokantaan, kun käyttäjä on hyväksynyt sen
poista.php	Tietojen poisto tietokannasta
virhe.php	Virhesivu, johon käyttäjä ohjataan, jos tunnistautuminen ei onnistu tai istunto on vanhentunut

Taulukko 1: PHP-tiedostot

## 9 Muut tiedostot

PHP-tiedostojen lisäksi laiterekisteri käyttää muita tiedostoja, jotka vaikuttavat sen ulkoasuun.

Tiedosto	Käyttötarkoitus
bilialit.gif	GIF-muotoinen kuvatiedosto, jossa on Bilian logo
lr.css	CSS-tyylitiedosto
valikko.phtml	Laiterekisterin navigoinnissa käytettävä valikko, joka on sisällytetty joka tiedostoon

Taulukko 2: Muut tiedostot

Suurin osa laiterekisteriin ulkoasuun vaikuttavista määrittelyistä sijaitsee lr.css -tiedostossa. Se on muodoltaan CSS (Cascading Style Sheets) -tiedosto, joka on sisällytetty jokaiseen tarvittavaan PHP-tiedostoon yhdellä HTML-koodirivillä:

```
<link rel="stylesheet" type="text/css" href="lr.css" />
```

Tiedosto sisältää määrittelyjä esimerkiksi väreistä, kirjasintyypeistä ja niiden koosta, taulukoiden ja lomakkeiden ulkoasusta sekä reunojen ja elementtien koosta. Määrittelyjen sijainti yhdessä tiedostossa helpottaa myös helpottavasti muutosten tekemistä ulkoasuun. Kun jokin tieto muuttuu, päivittyy se samalla kaikkiin muihin tiedostoihin, eikä muutoksia tarvitse

tehdä useaan kertaan. Esimerkiksi laiterekisterin sivun taustaväri, sivuille jäävän reunuksen koko sekä pääasiallisesti käytettävä kirjasin ja sen koko on määritetty seuraavasti:

```
body {
background-color:#f5f5f5;
font-family:Tahoma, Helvetica, sans-serif;
font-size: 12px;
margin:10;
}
```

Myös laiterekisterin navigointiin käytettävä valikko on tehty niin, että se koostuu vain yhdestä tiedostosta, joka haetaan sitä käyttävien sivujen sisälle. Itse valikko on yhdessä tiedostossa, joka on sisällytetty PHP-tiedostoihin PHP-komennolla:

```
<?php include 'valikko.phtml'; ?>
```

Näin valikon rakenne ja ulkoasu päivittyvät automaattisesti laiterekisterin jokaiseen osaan, kun sitä muokataan. Eri osilla ei ole tarvetta erilaisille valikoille, joten yhdessä tiedostossa sijaitseva valikko on sopivin ratkaisu.

## 10 Tietokannan rakenne

Tiedostojen lisäksi laiterekisteri käyttää yhtä MySQL-tietokantaa, johon kaikki laitetiedot on säilötty. Tietokannan rakenne pyrittiin suunnittelemaan niin, että se olisi helppo ylläpitää. Suunnittelussa pyrittiin ottamaan huomioon se, että syötettyihin tietoihin saattaa tulla muutoksia jälkikäteen. Rakennetta muokattiin työn edetessä tarpeiden mukaan. Seuraavassa taulukossa on esitetty tietokannan sisältämät taulut.

Taulu	Kuvaus
valmistajat	Laitevalmistajat
mallit	Laitemallit
valm_malli	Välitaulu, joka yhdistää laitemallit valmistajiin
kiinteistot	Kiinteistöt (yrityksen toimipisteet)
tyypit	Laitetyypit
laitteet	Laitetiedot

Taulukko 3: Tietokannan taulut

Tauluista valmistajat, mallit, valm\_malli, kiinteistot ja tyypit sisältävät jokainen kaksi kenttää. Taulut valmistajat, mallit, kiinteistot ja tyypit sisältävät kaikki yksittäisen tiedon sekä tätä vastaavan id-numeron. Taulussa valm\_malli on ainoastaan kaksi id-numerokenttää, jotka vastaavat taulujen valmistajat ja mallit id-numeroita.

Laitteet-taulu sisältää yhdeksän kenttää. Näissä kentissä viitataan id-numerolla tauluihin mallit, kiinteistot ja tyypit. Muita kenttiä ovat laitteen yksilöllinen id-numero, laitteen nimi, käyttäjä tai tarkempi sijainti, sarjanumero, IP-osoite sekä lisätietokenttä. Tähän tauluun ei tarvita tietoa valmistajasta, sillä taulu valm\_malli liittyy tietyn laitemallin tiettyyn valmistajaan.

Laitteet-taulun kentistä tyhjiä (null) voivat olla IP-osoite- ja lisätietokentät. Tyhjä IP-osoitekenttä tarkoittaa, että laite saa IP-osoitteen automaattisesti palvelimelta DHCP-protokollaa käyttäen. Muussa tapauksessa kenttä sisältää laitteen kiinteän IP-osoitteen. Laitteen lisätieto ei ole pakollinen tieto, joten kenttä voi olla tyhjä.

Seuraavassa on taulujen ja niiden kenttien tarkat nimet:

Taulu: valmistajat		
Kenttä	Tyhjä	Sisältö
valm_id	Ei	Yksilöllinen valmistajan id-numero
valmistaja	Ei	Valmistajan selkokielineen nimi

Taulukko 4: Valmistajat-taulu

Taulu: mallit		
Kenttä	Tyhjä	Sisältö
malli_id	Ei	Yksilöllinen mallin id-numero
malli	Ei	Mallin selkokielineen nimi

Taulukko 5: Mallit-taulu

Taulu: valm_malli		
Kenttä	Tyhjä	Sisältö
valm_id	Ei	Yksilöllinen valmistajan id-numero
malli_id	Ei	Yksilöllinen mallin id-numero

Taulukko 6: Valm\_malli-taulu

Taulu: kiinteistot		
Kenttä	Tyhjä	Sisältö
kiint_id	Ei	Yksilöllinen kiinteistön id-numero
kiinteisto	Ei	Kiinteistön selkokielen nimi

Taulukko 7: Kiinteistot-taulu

Taulu: tyypit		
Kenttä	Tyhjä	Sisältö
tyyppi_id	Ei	Yksilöllinen laitetypin id-numero
tyyppi	Ei	Laitetyypin selkokielen nimi

Taulukko 8: Tyypit-taulu

Taulu: laitteet		
Kenttä	Tyhjä	Sisältö
laite_id	Ei	Yksilöllinen laitteen id-numero
malli_id	Ei	Yksilöllinen mallin id-numero
tyyppi_id	Ei	Yksilöllinen laitetypin id-numero
kiint_id	Ei	Yksilöllinen kiinteistön id-numero
nimi	Ei	Laitteen nimi verkossa
kayttaja	Ei	Käyttäjä tai tarkempi sijainti
sn	Ei	Laitteen sarjanumero
ip	Kyllä	Laitteen IP-osoite
muut	Kyllä	Lisätietokenttä

Taulukko 9: Laitteet-taulu

Taulukoiden sarake ”tyhjä”, tarkoittaa, saako kenttä olla täysin tyhjä, vai onko siinä oltava jokin arvo.

Tietokantaan muodostetaan yhteys ja sitä käsitellään PHP-tiedostoihin sisällytetyillä komennoilla. Jokaisessa tiedostossa, jolla käsitellään tietokantaa, yhdistetään tietokantaan tiedoston alussa. Tämän jälkeen tulee muu ohjelmakoodi sekä tarvittavat tietokannan käsittelyt, ja tiedoston lopussa suljetaan yhteys.

Yhteyden avaus:

```
$yhteys = mysql_connect("localhost", "käyttäjätunnus", "salasana");
```

Tietokannan valinta:

```
mysql_select_db("rekisteri", $yhteys) or die("Virhe tietokantaa avatessa");
```

Yhteyden sulkeminen:

```
mysql_close($yhteys);
```

## 12 Testaus

Laiterekisteriä ja sen toimintaa pyrittiin myös testaamaan mahdollisimman monipuolisesti koko sen kehityksen ajan, kun ominaisuuksia tuli vähitellen lisää. Puhekielessä termi testaus voi tarkoittaa mitä tahansa kokeilemistä, mutta ohjelmistotestauksessa sillä tarkoitetaan suunnitelmallista virheiden etsimistä suorittamalla ohjelmaa tai sen osaa (Haikala & Märijärvi 2004, 284).

Laiterekisteriä kaikkia osia käytettiin järjestelmällisesti, ja syötettiin sinne erilaisia tietoja mahdollisten virheiden löytämiseksi. Ennen kuin varsinaisia tietoja alettiin syöttämään tietokantaan, sinne syötettiin mahdollisimman paljon erilaisia yhdistelmiä laitetiedoista, jotta mahdolliset virheet saatiin havaittua. Näitä koetietoja muokattiin ja poistettiin eri tavoilla niin, että mahdollisimman paljon eri variaatioita tiedoista saatiin käytyä läpi.

Tein testausta suurimmaksi osaksi itse. Kun ohjelma oli käyttökelpoinen niin, että tietoa sai muokattua ja lisättyä, tuli myös muu IT-osasto mukaan testaukseen. Heidän kommenttiansa perusteella tein useita muutoksia, korjauksia ja lisäyksiä laiterekisteriin.

Myös käyttöliittymän toimivuuden testaus oli oleellista. Käyttöliittymää testattiin Microsoft Internet Explorerin versioilla 6, 7 ja 8 sekä Mozilla Firefoxin versioilla 3.6.

### 13 Laiterekisterin käyttöönotto

Kun laiterekisterin suunnittelu, kehitys, ja testaus olivat riittävän pitkällä niin, että rekisteri voitiin ottaa käyttöön, se täytyi siirtää testiympäristönä toimineelta työasemalta palvelimelle, jolla sitä tullaan käyttämään.

Yrityksellä oli ylimääräisenä käytöstä poistunut, mutta vain parin vuoden ikäinen palvelinkone, joka oli täysin käyttökelpoinen. Koneeseen jouduttiin ostamaan uusi kovalevy, eikä sille ollut vapaata käyttöjärjestelmälisenssiä.

Palvelimen käyttöjärjestelmäksi valittiin Ubuntu Server Edition 10.04 LTS, joka on täysin ilmainen, eikä siitä tule lisenssikuluja. Tämän Ubuntu-version asennuksessa on myös vaihtoehtona asentaa LAMP-palvelin, joka tarkoittaa, että asennuksessa tulee valmiina Apache, MySQL ja PHP.

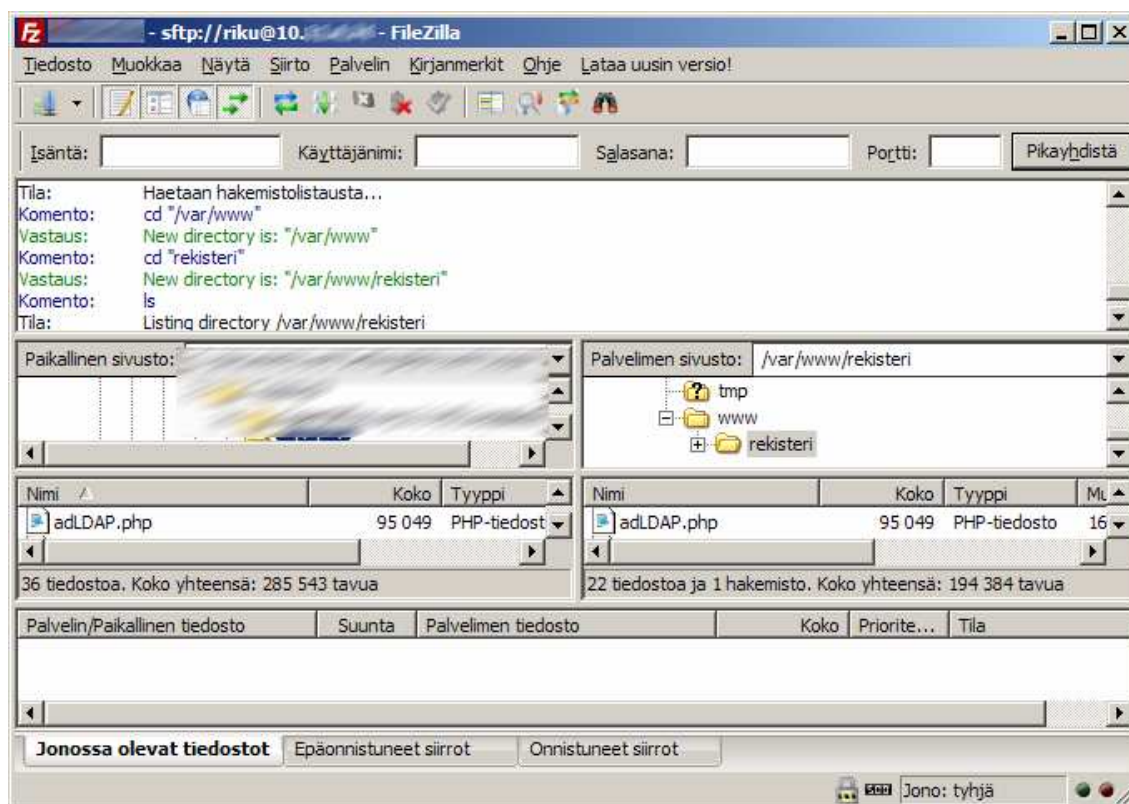
Asennusohjelma myös tekee tarvittavat asetukset näihin ohjelmiin valmiiksi, joten konfiguroinnin tarve jää minimaaliseksi.

Ubuntun asennuslevyn levykuva ladattiin internetistä, kopioitiin CD-levylle ja käynnistettiin asennus palvelinkoneelta. Asennuksessa valittiin LAMP-palvelimen asennus. Tämän jälkeen asennuksessa tehtiin tarvittavat kovalevyosioinnit, määritettiin pääkäyttäjän tunnus ja salasana sekä määritettiin koneen verkkoasetukset. Palvelimelle määritettiin sisäverkon kiinteä IP-osoite, eli DHCP ei ole käytössä.

### 14 Tietojen siirto testiympäristöstä

Asennuksen jälkeen täytyi laiterekisterin sisältämät tiedot kopioida testiympäristöstä uudelle palvelinkoneelle. Laiterekisterin käyttämiä tietoja ovat www-palvelimella sijaitsevat PHP-, kuva- ja tyylitiedostot, sekä MySQL-tietokanta kokonaisuudessaan.

WWW-palvelimen tiedostot kopioitiin testikoneelta palvelimelle suoraan SFTP-yhteyttä käyttäen. SFTP, eli SSH File Transfer Protocol, on tiedonsiirtoprotokolla, jolla saadaan siirrettyä tiedostoja kahden tietokoneen välillä salatusti.



Kuva 2: Tiedostojen siirto FileZilla-ohjelmistolla

Testiympäristön työasemalla avattiin siis SFTP-yhteys tulevaan palvelinkoneeseen, valittiin siirrettävät PHP-, kuva- ja tyyli-tiedostot ja siirrettiin ne palvelimelle määritettyyn hakemistoon, jossa web-palvelinohjelmisto käyttää niitä.

MySQL-tietokannan siirtoon käytettiin phpMyAdmin-ohjelmistoa. Ohjelmassa on tuonti- ja vientitoiminnot, joilla saadaan tallennettua tietokantojen sisältö erilliseen tiedostoon ja tallennettua tiedoston sisältö tietokantaan. Testiympäristön koneella kirjaututtiin phpMyAdminiin, valittiin laiterekisterin tietokanta toimintovalikosta Vienti. Vienti-toiminnon asetuksissa on valittuna kaikki tietokannan taulut ja tiedostomuotona SQL. Rakenteessa ja tiedoissa käytettiin oletusasetuksia. Tästä saatiin suoraan tallennettua SQL-muotoinen tiedosto omalle työasemalle.

Uudella palvelimella ei ollut phpMyAdmin-ohjelmaa esiasennettuna, mutta sen asennus onnistuu yksinkertaisesti seuraavalla komennolla:

```
sudo apt-get install phpmyadmin
```

Asennus kysyy pääkäyttäjän salasanan ja ohjelma asentuu automaattisesti. Tämän jälkeen phpMyAdmin on käytettävissä web-selaimella MySQL:n käyttäjätunnuksella ja salasanalla.



Tietokannan sai tuotua uudelle palvelimelle kirjautumalla phpMyAdminiin, luomalla uuden tyhjän tietokannan ja valitsemalla valikosta Tuonti. Asetukset muutettiin vastaamaan testiympäristöstä tallennettua SQL-tiedostoa, ja liitettiin tiedosto tänne. Näin tietokanta siirtyi kokonaisuudessaan.

The screenshot shows the phpMyAdmin interface for a database named 'rekisteri'. The left sidebar shows the database structure with tables: kiinteistot, laitteet, mallit, tyypit, valmistajat, valm\_malli. The main area displays a table list with columns: Taulu, Toiminnot, Rivit, Typpi, Aakkosjärjestys, Koko, Ylijaama. Below the table list is a 'Luo uusi taulu' form with fields for 'Nimi' and 'Kenttien määrä'.

Taulu	Toiminnot	Rivit	Tyyppi	Aakkosjärjestys	Koko	Ylijaama
kiinteistot		5	MyISAM	latin1_swedish_ci	2,1 kt	-
laitteet		304	MyISAM	latin1_swedish_ci	22,2 kt	-
mallit		14	MyISAM	latin1_swedish_ci	3,3 kt	-
tyypit		6	MyISAM	latin1_swedish_ci	2,1 kt	-
valmistajat		3	MyISAM	latin1_swedish_ci	2,1 kt	-
valm_malli		14	MyISAM	latin1_swedish_ci	2,1 kt	-
<b>6 taulu(a)</b>	<b>Summa</b>	<b>346</b>	<b>MyISAM</b>	<b>latin1_swedish_ci</b>	<b>33,9 kt</b>	<b>0 tavua</b>

Kuva 3: Tietokannan hallinta phpMyAdmin-ohjelmalla

## 15 Tietojen varmuuskopiointi

Laiterekisterin sisältämistä tiedoista täytyy luoda varmuuskopiot säännöllisesti, jotta esimerkiksi laiterikon tai muun virheen sattuessa ne voidaan palauttaa. Laiterekisterin varmuuskopioitavia tiedostoja ovat WWW-palvelimen sisältö ja tietokanta.

WWW-palvelimen tiedostojen, eli PHP-, kuva- ja tyyli-tiedostojen, sisältö muuttuu vain, jos laiterekisterin toiminnallisuuteen tehdään muutoksia. Tästä syystä näiden tiedostojen automaattinen varmuuskopiointi ei ole tarpeen. Näistä tiedostoista on tehty varmuuskopiot verkkolevylle asennusvaiheessa, ja niistä tehdään varmuuskopiot manuaalisesti aina, kun niiden sisältö muuttuu. Varmuuskopiointi onnistuu SFTP-yhteyttä käyttäen: työasemalla otetaan yhteys palvelinkoneeseen, ja siirretään sieltä tiedostot työaseman kovalevylle. Työasemalta ne vielä kopioidaan erikseen varmistetulle verkkoasemalle.

Tietokannan sisältö taas muuttuu jatkuvasti, kun rekisteristä poistetaan, muokataan ja lisätään tietoa. Tästä syystä sen sisällöstä on syytä luoda varmuuskopio säännöllisin väliajoin. Päädyin ratkaisuun, jossa tietokannasta luodaan joka yö kopio palvelinkoneen omalle kiintolevylle sekä erilliselle verkkolevylle, josta otetaan erikseen omat varmuuskopiot nauhavarmistusasemalle. Paikallisen koneen varmuuskopio on helposti ja nopeasti saatavilla,

jos tietokannasta esimerkiksi poistetaan vahingossa tarpeellista tietoa. Erillisellä verkkolevyllä on hyvä olla varmuuskopio sen varalta, että palvelinkoneen kiintolevy rikkoutuu.

Verkkolevy, johon varmuuskopiot tehdään, on osa yrityksen Active Directory -verkkoa, ja sen käyttämiseen tarvitaan oma käyttäjätunnus ja salasana. Nämä tunnistetiedot on tallennettu palvelinkoneelle tiedostoon `/root/.smbcredentials` seuraavasti:

```
username=TOIMIALUE\käyttäjätunnus
password=salasana
```

Tiedostossa on siis yrityksen Active Directory -toimialueen nimi, sekä tähän verkkolevyn osioon pääsyyn oikeuttava käyttäjätunnus ja salasana.

Verkkolevyn polku on määritetty tiedostoon `/etc/fstab` niin, että palvelin yhdistää siihen automaattisesti käynnistyksen yhteydessä:

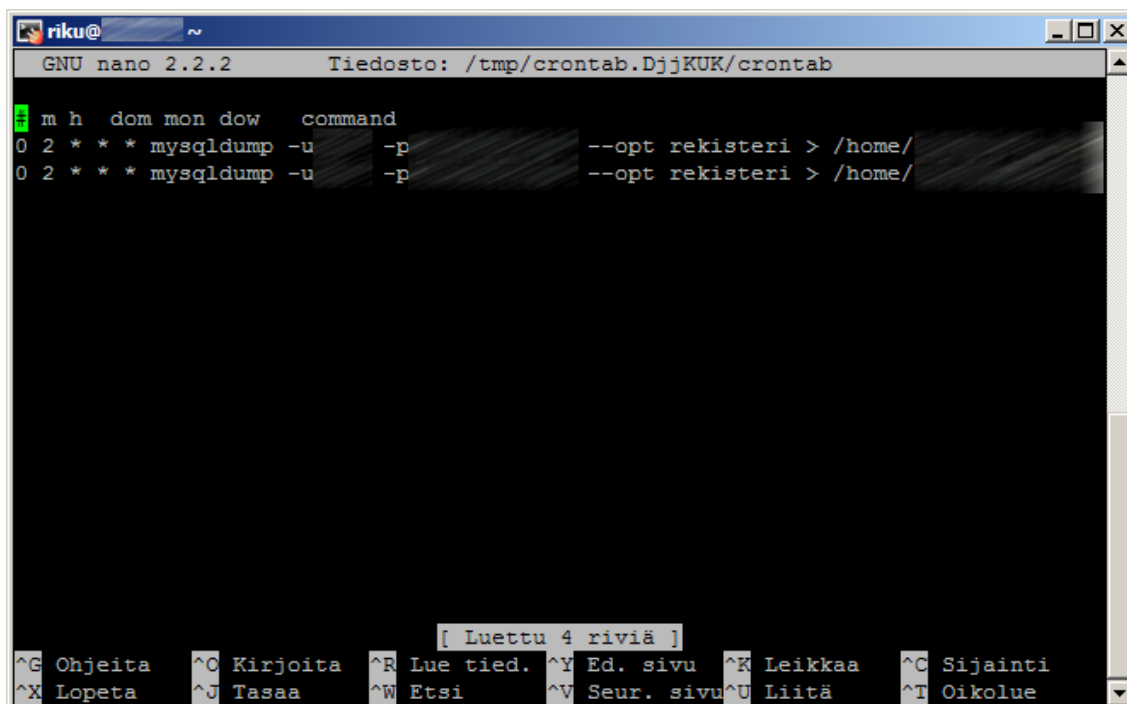
```
//verkkolevy/polku/linux_backup /home/riku/backup cifs
sec=ntlmv2,credentials=/root/.smbcredentials,icharset=utf8,file_mode=0777,dir_mode=0777
0 0
```

Varmuuskopioinnin saa ajastettua mysqldump-komennolla crontab-tiedostoon. Seuraavilla komennoilla kopioidaan MySQL-tietokannan sisältö joka yö kello 02.00 tiedostoon, jonka nimi on rekisteri<päivämäärä>.sql, esimerkiksi rekisteri20110401.sql. Ensimmäinen komento on paikalliseen kopiointiin, toinen verkkosemalle.

```
0 2 * * * mysqldump -ukäyttäjätunnus -psalasana --opt rekisteri >
/home/riku/sql_backup/rekisteri`/bin/date +%Y%m%d`.sql
```

```
0 2 * * * mysqldump -ukäyttäjätunnus -psalasana --opt rekisteri >
/home/riku/backup`/bin/date +%Y%m%d`.sql
```

Ubuntu Linux -käyttöjärjestelmässä kaikkia näitä tiedostoja saadaan muokattua nano-tekstieditorilla, esimerkiksi komennolla `sudo nano /etc/fstab/`. Ensimmäinen komento `sudo` tarkoittaa, että sen jälkeinen komento `nano` suoritetaan pääkäyttäjän oikeuksin.



```

GNU nano 2.2.2 Tiedosto: /tmp/crontab.DjjKUK/crontab
m h dom mon dow  command
0 2 * * * mysqldump -u [redacted] -p [redacted] --opt rekisteri > /home/[redacted]
0 2 * * * mysqldump -u [redacted] -p [redacted] --opt rekisteri > /home/[redacted]

[ Luettu 4 riviä ]
^G Ohjeita ^C Kirjoita ^R Lue tied. ^Y Ed. sivu ^K Leikkaa ^C Sijainti
^X Lopeta ^J Tasaa ^W Etsi ^V Seur. sivu ^U Liitä ^T Oikolue

```

Kuva 4: nano-tekstieditorin käyttö crontab-ajastusten muokkaukseen

## 16 Laiterekisterin käyttö ja toiminta

### 16.1 Kirjautuminen ja etusivu

Kun WWW-selaimeen syötetään laiterekisterin osoite, aukeaa kirjautumissivu, johon syötetään Active Directory -toimialueen käyttäjätunnus ja salasana. Tälle sivulle on tehty JavaScript-kielellä määrittäminen, joka siirtää näppäimistön kohdistimen automaattisesti käyttäjätunnus-kenttään. Näin käyttäjätunnuksen kirjoittamisen voi aloittaa heti, kun sivu on latautunut, eikä kohdistinta tarvitse erikseen siirtää käyttäjätunnuksen kohdalle. JavaScript-koodi on seuraava:

```

<script language="javascript">
function kursori(kayttajatunnus) {
    var n = document.getElementById(kayttajatunnus);
    n.focus();
}


</script>

<body onload="kursori('kayttajatunnus');">

```

Tämän lisäksi tekstikenttä käyttäjätunnusta varten on nimetty id-määrittelyllä:

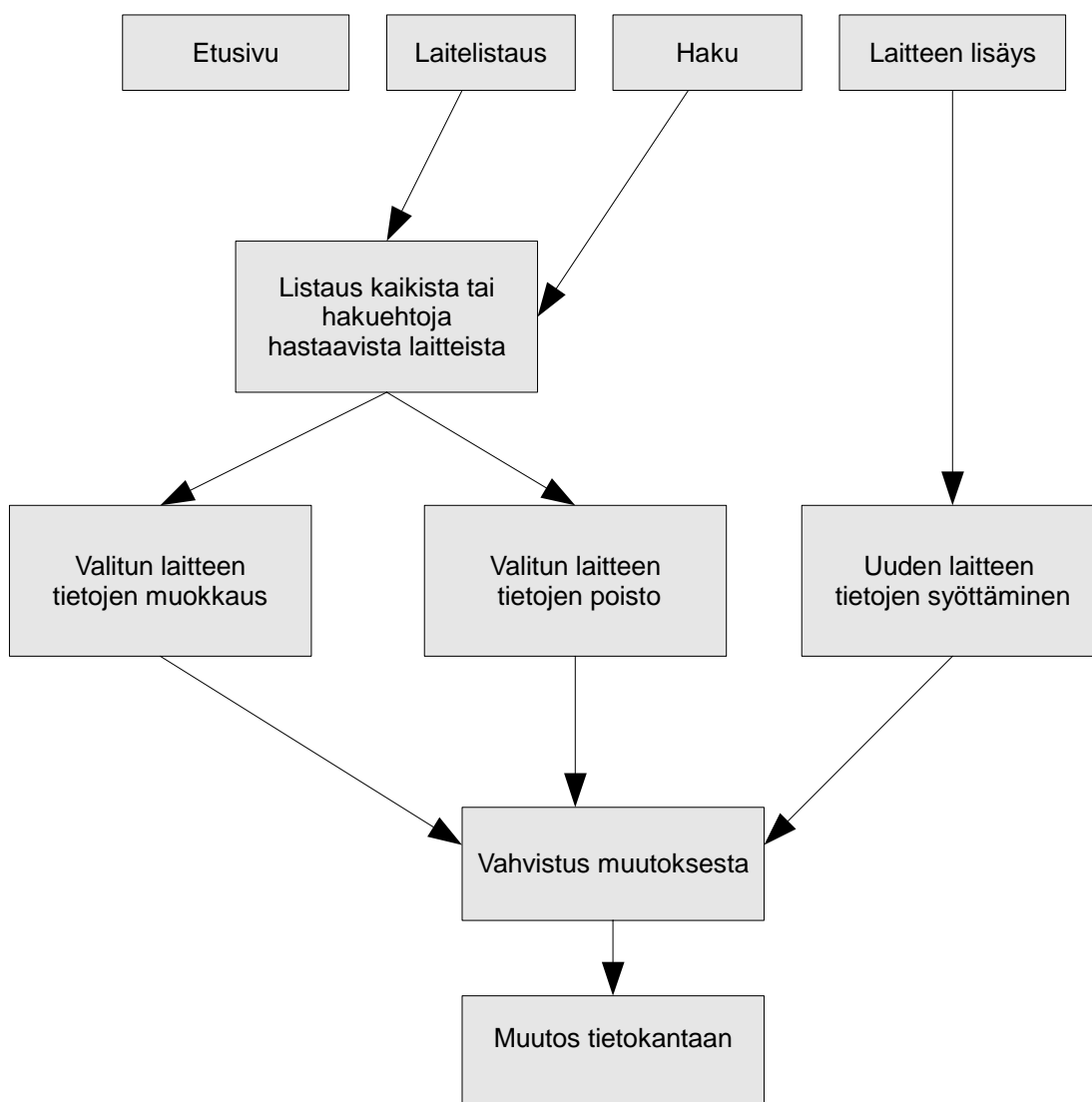
```
<input type="text" name="kayttajatunnus" id="kayttajatunnus">
```



The image shows a login form for 'Laiterekisteri' by BILIA IT. The form is contained within a light green rectangular box. At the top left of the box is the BILIA IT logo, which consists of a blue icon of a person with arms raised next to the text 'BILIA IT' in a bold, blue, sans-serif font. Below the logo, the title 'Laiterekisteri' is displayed in a bold, black, sans-serif font. The form contains two input fields: the first is labeled 'Käyttäjätunnus' (Username) and the second is labeled 'Salasana' (Password). Both labels are in a black, sans-serif font. Below the password field is an 'OK' button with a black border and black text.

Kuva 5: Laiterekisteriin kirjautuminen

Jos kirjautuminen epäonnistuu, esimerkiksi jos salasana on väärin, saa käyttäjä siitä ilmoituksen, ja mahdollisuuden palata linkistä takaisin kirjautumissivulle. Onnistuneen kirjautumisen jälkeen avautuu laiterekisterin etusivu, jossa on laiterekisteriin liittyviä ajankohtaisia tiedotteita sekä valikko laiterekisterin käyttöä varten. Valikko on näkyvissä sivun vasemmassa reunassa samanlaisena riippumatta siitä, mitä laiterekisterin osiota ollaan käyttämässä. Navigointi laiterekisterissä tapahtuu seuraavan kaavion mukaisesti:



Kuvio 2: Laiterekisterin valikon käyttö

## 16.2 Tietojen listaus ja haku

Laiterekisterissä on mahdollista listata koko tietokannan sisältö ja selata sitä, tai etsiä tietoa eri hakuehdolla. Mahdollisia hakuehtoja ovat laitteen nimi, käyttäjän nimi, laitteen malli, kiinteistö, sarjanumero sekä lisätietokenttä. Sekä koko tietokannan listauksessa että hakutuloksissa on mahdollista valita yksittäinen laite ja muokata sen tietoja tai poistaa se kokonaan.

Esimerkkikuvassa on haettu hakuehdolla, jossa käyttäjän nimi on Rauvala:

Laitteita löytyi 2 kpl

Tyyppi	Valmistaja	Malli	Laitteen nimi	Käyttäjä	Kiinteistö	Sarjanumero	Lisätiedot	IP-osoite	Muokkaa	Poista
Kannettava	Lenovo			Riku Rauvala	Kaivoksela 6 A				Muokkaa	Poista
Pöytäkone	Dell			Riku Rauvala	Kaivoksela 6 A				Muokkaa	Poista

Kuva 6: Hakutulokset

Käyttäjän nimellä haettaessa tehdään kysely MySQL-tietokantaan. Hakutulokset muodostetaan toistolauseella, joka hakee tietokannasta hakutuloksia vastaavien laitteiden tiedot ja luo taulukkoon rivejä hakutulosten määrän verran. Näiden toimintojen ohjelmakoodi on liitteessä 1.

### 16.3 Uuden laitteen tietojen lisäys

Kun tietokantaan halutaan lisätä uusi laite, aukeaa lomake, jossa on kentät eri tiedoille. Kentistä ovat pakollisia kaikki paitsi IP-osoite ja lisätietokenttä. Jos jokin pakollisista kentistä on tyhjä, saa käyttäjä tästä ilmoituksen, ja joutuu palaamaan täyttämään puutteelliset tiedot. Tämän lisäksi laiterekisteri tarkistaa, onko tietokannassa jo olemassa laite sillä nimellä tai sarjanumerolla, jota ollaan lisäämässä.

### Lisää uusi laite

Tyyppi	<input type="text" value="Pöytäkone"/>
Malli	<input type="text" value="Optiplex 780"/>
Laitteen nimi	<input type="text"/>
Käyttäjä/tarkempi sijainti	<input type="text"/>
Kiinteistö	<input type="text" value="Kaivoksela 6 A"/>
Sarjanumero	<input type="text"/>
IP-osoite	<input type="text" value="DHCP"/> <input type="text"/>
Lisätiedot	<input type="text"/>

Kuva 7: Uuden laitteen lisäys

Vaihtoehdot laitteen tyyppiä, mallia ja kiinteistöä varten haetaan tietokannasta lomakkeen valmiisiin vetoalukoihin. Muita tietoja varten on tekstikentät, joihin tiedot voi syöttää. Myös IP-osoitetietoa varten on vetoalikko, jonka valinnoista DHCP tarkoittaa että IP-osoitetta ei määritetä tietokantaan. Jos laitteella on kiinteä IP-osoite, määritetään se viereiseen tekstikenttään.

Kun kaikki tiedot ovat oikein, näytetään ne vielä käyttäjälle erillisellä sivulla. Tällä sivulla on mahdollisuus palata lomakkeelle korjaamaan tietoja. Jos tiedot ovat oikeen, vahvistetaan ne ja tehdään lopullinen lisäys tietokantaan:

Onnistuneen lisäyksen jälkeen tarjotaan käyttäjälle mahdollisuutta lisätä uusi samanmallinen laite. Tämä linkki johtaa lomakkeeseen, jossa on valmiina sama tyyppi ja malli kuin juuri lisätyssä laitteessa. Tämä ominaisuus on osoittautunut hyödylliseksi, kun on hankittu useampia samanmallisia laitteita kerralla, ja syötetään näiden tietoja tietokantaan.

## 16.4 Muokkaus ja poisto

Laitteen muokkaus avaa lomakkeen, jossa on valmiina laitteen nykyiset tiedot, ja jokaista kenttää on mahdollista muokata. Lomake on siis täysin samanlainen kun lisättäessä uutta laitetta, mutta siihen avautuu valmiina muokattavan laitteen nykyiset tiedot. Kun muokkausnappia painetaan, näytetään käyttäjälle uudet tiedot. Myös tässä lomakkeessa on tarkistus, ettei mitään pakollisia tietoja jätetä tyhjäksi. Samoin muokattavat tiedot näytetään ja pyydetään vahvistus ennen lopullista lisäystä.

**Muokkaa laitetta**

Tyyppi

Malli

Laitteen nimi

Käyttäjä/tarkempi sijainti

Kiinteistö

Sarjanumero

IP-osoite

Lisätiedot

Kuva 8: Laitteen tietojen muokkaus

Laitteen poistaminen tapahtuu laitelistauksen tai hakutulosten kautta. Poistolinkin klikkaus avaa vahvistuksen, jossa vahvistetaan laitteen poisto, ja kerrotaan vielä poistettavan laitteen nimi. Tästä näkymästä on mahdollista palata takaisin, tai vahvistaa poisto.

Tämä toiminto poistaa laitteen tiedot tietokannasta lopullisesti, ja käyttäjä saa ilmoituksen onnistuneesta poistosta.



## 17 Yhteenveto

Tätä kirjoitettaessa laiterekisteri on ollut yrityksen IT-osastolla aktiivisessa käytössä jo jonkin aikaa. Se on osoittautunut hyödylliseksi ja toimivaksi erityisesti it-tukitehtävissä, kun on selvitettävä loppukäyttäjän työaseman tiedot nopeasti. Selkeä, yhtenäinen tietokanta laitteista on tehnyt tietojen ylläpidon helpoksi, ja tietoja on käytännössä myös päivitetty tiheämmin kuin aikaisemmin.

Työ on valmistuessaan helpottanut myös omia työtehtäviäni, joihin perustietotekniikan ylläpito ja tukitehtävät kuuluvat. Käytän siis itse toteuttamaani järjestelmää omissa työtehtävissäni yhdessä muiden työntekijöiden kanssa.

Projektin myötä on käynyt ilmi, kuinka paljon ajan tasalla oleva tieto helpottaa työntekoa. Myöskään ominaisuuksia, joita mahdollisesti löytyisi vastaavista kaupallisista toteutuksista mutta joita ei ole mahdollista tämän mittakaavan työssä toteuttaa, ei ole kaivattu. Itsenäisesti toteutettu, omaan käyttöön räätälöity laiterekisteri on osoittautunut sopivimmaksi vaihtoehdoksi.

Työ oli erittäin mielenkiintoinen, mutta myös haastava. En ole tehnyt PHP- tai MySQL-ohjelmointia työkseni, vaan lähinnä harrastuksena. Jouduin siis aloittamaan koko järjestelmän rakentamisen täysin tyhjästä. Ongelmat kuitenkin ratkesivat ajan myötä opettelemalla PHP-funktioiden ja MySQL-kyselyiden toimintaa tarkemmin. Opin näiden käytöstä paljon uutta, ja minulla on valmiuksia kehittää työtä jatkossa paremmaksi.

Laiterekisteriin onkin tehty muutoksia sitä mukaa kun puutteita tai kehityskohteita on käytössä huomattu, ja sitä on tarkoitus kehittää jatkuvasti myös tulevaisuudessa. Eräs mahdollisista lisäominaisuuksista on yrityksen puhelinlaitteiden tietojen lisääminen tietokantaan tietoteknisten laitteiden lisäksi. Tämä lisäominaisuus vaatii joitakin uusia kenttiä tietokantaan ja sen käsittelyyn.

Olen kaikin puolin erittäin tyytyväinen työhön ja sen lopputulokseen. Laiterekisteri on toiminnaltaan tavoitteiden mukainen, sekä omasta mielestäni että muiden työntekijöiden, jotka sitä käyttävät. Alkuvaiheen testaus sujui hyvin, ja tein monta korjausta ohjelmaan muiden työntekijöiden kommenttien perusteella.

Henkilökohtaisena tavoitteena projektissa oli erityisesti kehittää omaa ammatillista osaamistani. En ollut aiemmin toteuttanut laajempaa kokonaisuutta, joka tulee yrityksessä muiden henkilöiden käyttöön. Vaikka pääasialliset työtehtäväni eivät liitykään ohjelmointiin

tai tietokantoihin, tulee myös perusosaamisen kehittämistä näillä osa-alueilla varmasti olemaan hyötyä tulevaisuudessa.

Jotta laiterekisteri pysyisi jatkossakin käyttökelpoisena ja sitä voitaisiin hyödyntää tehokkaasti, on tärkeää, että sen sisältämät tiedot pidetään ajan tasalla. Itse ohjelman hyvä tekninen toiminnallisuus ei riitä, jos sen sisältämä data ei ole käyttökelpoista. On siis pidettävä huolta, että tietoja ylläpidetään myös tulevaisuudessa tehokkaasti.

## Lähteet

Barnett, S & Hyland, R. adLDAP - LDAP Authentication with PHP for Active Directory. Viitattu 20.4.2011. <http://adldap.sourceforge.net/>

Casad, Joe & Willsey, Bob 1999. TCPIP Trainer. Helsinki: IT Press.

Haikala, I. & Märijärvi, J. 2004. Ohjelmistotuotanto. Helsinki: Talentum Media Oy.

Holme, D., Ruest, D. & Ruest, N. 2008. Configuring Windows Server 2008 Active Directory. Redmond: Microsoft Press.

How DNS Support for Active Directory Works. 2003. Microsoft. Viitattu 24.4.2011. <http://technet.microsoft.com/en-us/library/cc759550%28WS.10%29.aspx>

Kolehmainen, K. 2006. PHP & MySQL - Teoriasta käytäntöön. Jyväskylä: Gummerus.

PHP Manual. 2011. The PHP Group. Viitattu 20.4.2011. <http://www.php.net/manual/en/>

phpMyAdmin. 2011. The phpMyAdmin development team. Viitattu 24.4.2011. [http://www.phpmyadmin.net/home\\_page/index.php](http://www.phpmyadmin.net/home_page/index.php)

Seidler, K. 2011. XAMPP. Viitattu 6.1.2011. <http://www.apachefriends.org/en/xampp.html>

The Apache HTTP Server Project. 2011. The Apache Software Foundation. Viitattu 24.4.2011. <http://httpd.apache.org/>

## Kuvat

Kuva 1: XAMPP-ohjauspaneeli .....	12
Kuva 2: Tiedostojen siirto FileZilla-ohjelmistolla .....	24
Kuva 3: Tietokannan hallinta phpMyAdmin-ohjelmalla.....	25
Kuva 4: nano-tekstieditorin käyttö crontab-ajastusten muokkaukseen .....	27
Kuva 5: Laiterekisteriin kirjautuminen.....	28
Kuva 6: Hakutulokset .....	30
Kuva 7: Uuden laitteen lisäys .....	31
Kuva 8: Laitteen tietojen muokkaus.....	32

## Kuviot

Kuvio 1: Toteutusprosessin vaiheet .....	10
Kuvio 2: Laiterekisterin valikon käyttö.....	29

## Taulukot

Taulukko 1: PHP-tiedostot.....	18
Taulukko 2: Muut tiedostot.....	18
Taulukko 3: Tietokannan taulut .....	19
Taulukko 4: Valmistajat-taulu .....	20
Taulukko 5: Mallit-taulu .....	20
Taulukko 6: Valm_malli-taulu .....	21
Taulukko 7: Kiinteistot-taulu .....	21
Taulukko 8: Tyypit-taulu.....	21
Taulukko 9: Laitteet-taulu .....	21

## Liitteet

Tässä liitteessä on kuvattu laiterekisterin toiminnassa oleellisimpia tietokantakyselyitä sekä ohjelmakoodin osia tiedon muokkausta, lisäystä ja poistamista varten.

Esimerkki SQL-lauseesta haettaessa tietokannasta tietoa käyttäjän nimellä:

```
$laitekysely = "SELECT * FROM laitteet where $kayttaja like \"%$kayttajahaku%\"";
```

Toistolause hakutulosten taulukon rakentamiseen:

```
$tulostenmaara = mysql_num_rows($laitehaku);  
echo "Laitteita löytyi $tulostenmaara kpl<br/><br/>";  
  
echo "<table border>";  
echo  
"<tr><td><b>Tyyppi</b></td><td><b>Valmistaja</b></td><td><b>Malli</b></td><td><b>Laitteen  
nimi</b></td><td><b>Käyttäjä</b></td><td><b>Kiinteistö</b></td><td><b>Sarjanumero</b></td><td><b>Lisätiedot</b></td><td><b>IP-  
osoite</b></td><td><b>Muokkaa</b></td><td><b>Poista</b></td></tr>";  
  
if (mysql_num_rows($laitehaku) == 0) { echo "<tr><td colspan=9>Ei hakutuloksia</td></tr>"; }  
  
for ($i = 0; $i < mysql_num_rows($laitehaku); $i++) {  
    $laite_id = mysql_result($laitehaku, $i, "laite_id");  
    $tyyppi_id = mysql_result($laitehaku, $i, "tyyppi_id");  
    $kiint_id = mysql_result($laitehaku, $i, "kiint_id");  
    $malli_id = mysql_result($laitehaku, $i, "malli_id");  
    $nimi = mysql_result($laitehaku, $i, "nimi");  
    $kayttaja = mysql_result($laitehaku, $i, "kayttaja");  
    $sn = mysql_result($laitehaku, $i, "sn");  
    $muut = mysql_result($laitehaku, $i, "muut");  
    $ip = mysql_result($laitehaku, $i, "ip");  
  
    $tyyppikysely = "SELECT * FROM tyypit, laitteet WHERE laitteet.laite_id = $laite_id AND  
tyypit.tyyppi_id = $tyyppi_id";  
    $tyyppihaku = mysql_query($tyyppikysely, $yhteys) or die("Virhe kyselyssä tyyppi!");  
  
    $tyyppi = mysql_result($tyyppihaku, 0, "tyyppi");  
  
    $kiintkysely = "SELECT * FROM kiinteistot, laitteet WHERE laitteet.laite_id = $laite_id AND  
kiinteistot.kiint_id = $kiint_id";  
    $kiinthaku = mysql_query($kiintkysely, $yhteys) or die("Virhe kyselyssä kiinteistö!");  
  
    $kiinteisto = mysql_result($kiinthaku, 0, "kiinteisto");  
  
    $mallikysely = "SELECT * FROM mallit, laitteet WHERE laitteet.laite_id = $laite_id AND  
mallit.malli_id = $malli_id";  
    $mallihaku = mysql_query($mallikysely, $yhteys) or die("Virhe kyselyssä malli!");  
  
    $malli = mysql_result($mallihaku, 0, "malli");  
  
    $valmkysely = "SELECT * FROM valmistajat, valm_malli, mallit WHERE valmistajat.valm_id =  
valm_malli.valm_id AND mallit.malli_id = valm_malli.malli_id AND mallit.malli_id =  
$malli_id";  
    $valmhaku = mysql_query($valmkysely, $yhteys) or die("Virhe kyselyssä valmistaja!");
```

```
$valmistaja = mysql_result($valmhaku, 0, "valmistaja");

echo "<tr onMouseOver=\"this.bgColor='#cccccc';\"
onMouseOut=\"this.bgColor='#f1f0ff';\"><td>$tyyppi</td><td>$valmistaja</td><td>$malli</td>
<td>$nimi</td><td>$kayttaja</td><td>$kiinteisto</td><td>$sn</td><td>$muut</td><td>$ip
&nbsp;</td><td><a href=\"muokkaa.php?id=\" . $laite_id . \">Muokkaa</a></td><td><a
href=\"poista.php?id=\" . $laite_id . \">Poista</a></td></tr>";

}
echo "</table>";
```

Tarkistus, onko laiterekisterissä jo olemassa laite samalla nimellä tai sarjanumerolla, jota ollaan lisäämässä:

```
$nimicheck_kysely = "SELECT * FROM laitteet WHERE nimi = '$nimi' OR sn = '$sn'";
```

```
$nimicheck_haku = mysql_query($nimicheck_kysely, $yhteys) or die("Virhe kyselyssä!");
```

Uuden laitteen lopullinen lisäys tietokantaan:

```
$lisaa = "INSERT INTO rekisteri.laitteet (laite_id, tyyppi_id, malli_id, kiint_id, nimi, kayttaja,
sn, ip, muut) VALUES (NULL, $tyyppi, $malli, $kiinteisto, '$nimi', '$kayttaja', '$sn', '$ip',
'$muut')";
```

Laitteen tietojen muokkaus tietokannassa:

```
$lisaa = "UPDATE rekisteri.laitteet SET tyyppi_id = $tyyppi, malli_id = $malli, kiint_id =
$kiinteisto, nimi = '$nimi', kayttaja = '$kayttaja', sn = '$sn', ip = '$ip', muut = '$muut' WHERE
laite_id = $laiteid";
```

Laitteen poisto tietokannasta:

```
$poistosql = "DELETE FROM laitteet WHERE laite_id = $_POST[poistoid]";
```