

## ALKULAUSE

Tämä opinnäytetyö on tehty Veljekset Paakki Oy:lle. Työn tilaajan edustajana toimi Jaakko Paakki, ohjaavana opettajana lehtori Leo Ilkko, kieltenohjaajina lehtori Kaija Heinula ja tuntiopettaja Kaija Posio. Kiitokset työn aiheen tarjoamisesta kuuluvat Veljekset Paakki Oy:lle, laboratorioinsinööri Janne Räihälle ja työn valmistumisesta kaikille yllä mainituille henkilöille.

Oulussa 3.4.2012

Antti Tapaus

## TIIVISTELMÄ

Oulun seudun ammattikorkeakoulu, Raahen tekniikan ja talouden kampus  
Tietotekniikka

---

Tekijä: Antti Tapaus

Opinnäytetyön nimi: MySQL ja PHP liiketoiminnan kehittämisen työkaluina

Työn ohjaaja: Leo Ilkko

Työn valmistumislukukausi ja -vuosi: Kevät 2012

Sivumäärä: 31

---

Opinnäytetyön tarkoituksena oli rakentaa työn tilaajan tarpeisiin soveltuva sopimusten hallintajärjestelmä web-käyttöliittymällä. Vaatimuksiin kuuluivat muun muassa sopimusten luonti, muokkaus, tulostus, haku useilla eri kriteereillä, sopimuskohtaisten liitetiedostojen hallinta, eri tyyppiset käyttäjäryhmät ja käyttäjätilien hallinta.

Työn ensimmäisessä vaiheessa toteutettiin järjestelmän runko, joka sisälsi MySQL-tietokannan, tietokantayhteyksien hallintaluokan, web-käyttöliittymän rungon, mahdollisuuden sisään- ja uloskirjautumiseen sekä käyttäjän oikeuksien tunnistamisen. Toinen vaihe sisälsi vaatimusmäärittelyn mukaisten toimintojen rakentamisen ja liittämisen järjestelmään. Jokainen uusi toiminto oli lähes itsenäinen web-sovellus ja riippuvainen vain järjestelmän rungosta, joten näiden rakentaminen ja integroiminen järjestelmään pystyttiin toteuttamaan jokaisen kohdalla lähes täysin itsenäisesti. Järjestelmän kehittäminen noudatti pääpiirteittäin evo-mallia, joka muodostuu sarjasta toistuvia päivityksiä, joista jokaisen tuloksena on uusilla ominaisuuksilla kasvatettu järjestelmä.

Työprosessin aikana muodostui selkeä näkemys web- ja tietokantaohjelmoinnista sekä niiden tarjoamista mahdollisuuksista ja rajoituksista. Projektin lopputuloksena syntyi käyttövalmis vaatimusmäärittelyn toteuttava järjestelmä. Lopuksi järjestelmä toimitettiin työn tilaajalle asennettuna ja käyttövalmiina.

---

Asiasanat: PHP, MySQL, HTML, web, ohjelmistokehitys

## ABSTRACT

Oulu University of Applied Sciences  
Degree Programme in Information Technology

---

Author: Antti Tapaus

Title of thesis: MySQL and PHP as Business Development Tools

Supervisor: Leo Ilkko

Term and year when the thesis was submitted: Spring 2012

Number of pages: 31

---

The purpose of this Bachelor's thesis was to build a contract management system with a web-based interface suitable for the needs of the client. Among the demands were functions for creating contracts, editing, printing, creating a search engine with wide-ranging criteria, managing contract-related files, different types of user groups and user accounts.

The first phase of this project implemented the system framework containing a MySQL database, a database connections management class, a web-interface frame, a means to log in and out, and a user rights identification. The second phase included the implementation of the functions required by the specification. Each new feature was an almost independent web application and only dependent on the system framework, so in each case the development and integration into the system could be done almost independently. The development of the system followed the outline of the evolutionary delivery model, consisting of a series of repeated updates, each producing new features to the system.

A clear vision about web and database programming, as well as their potential and limitation, was formed during the working process. The final project result was a user-ready system which met the specification. Finally, the system was delivered to the client fully installed and ready for use.

---

Keywords: PHP, MySQL, HTML, web, software, development

## SISÄLLYS

ALKULAUSE.....	1
TIIVISTELMÄ.....	2
ABSTRACT.....	3
SISÄLLYS.....	4
1 JOHDANTO.....	7
1.1 MySQL.....	7
1.2 PHP.....	9
1.3 HTML.....	10
2 MÄÄRITELMÄ.....	12
2.1 Tietokanta.....	12
2.2 Sisään- ja uloskirjautuminen.....	12
2.3 Käyttäjätilit ja käyttäjien hallinta.....	12
2.4 Käyttöliittymät.....	13
2.4.1 Asennuksen käyttöliittymä.....	13
2.4.2 Myynnin käyttöliittymä.....	13
2.4.3 Työnjohdon käyttöliittymä.....	14
2.5 Sopimus pohjat ja sopimusten hallinta.....	14
2.6 Sopimusten tulostus.....	14
2.7 Sopimusten hakutoiminto.....	15
4 TOTEUTUS.....	16
4.1 Evo-malli.....	16
4.2 GET- ja POST-metodit.....	16
4.3 Järjestelmän runko.....	17
4.3.1 Tietokanta.....	17
4.3.2 Tietokantayhteyksien hallintaluokka.....	19

4.3.3 Sisään- ja uloskirjautuminen.....	21
4.3.4 Web-käyttöliittymän runko.....	22
4.4 Käyttäjätilit ja käyttäjien hallinta.....	23
4.5 Sopimus pohjat ja sopimusten hallinta.....	24
4.6 Sopimusten haku.....	24
5 TESTAUS.....	25
5.1 Tietoturva.....	25
5.2 Sopimustietojen paikkaansapitävyys.....	25
5.3 Sopimusten hakutoiminto.....	25
5.4 Käyttöliittymä ja navigointi.....	26
6 JATKOKEHITYSMAHDOLLISUUDET.....	27
7 YHTEENVETO.....	28
LÄHDELUETTELO.....	29
LIITTEET.....	30

# 1 JOHDANTO

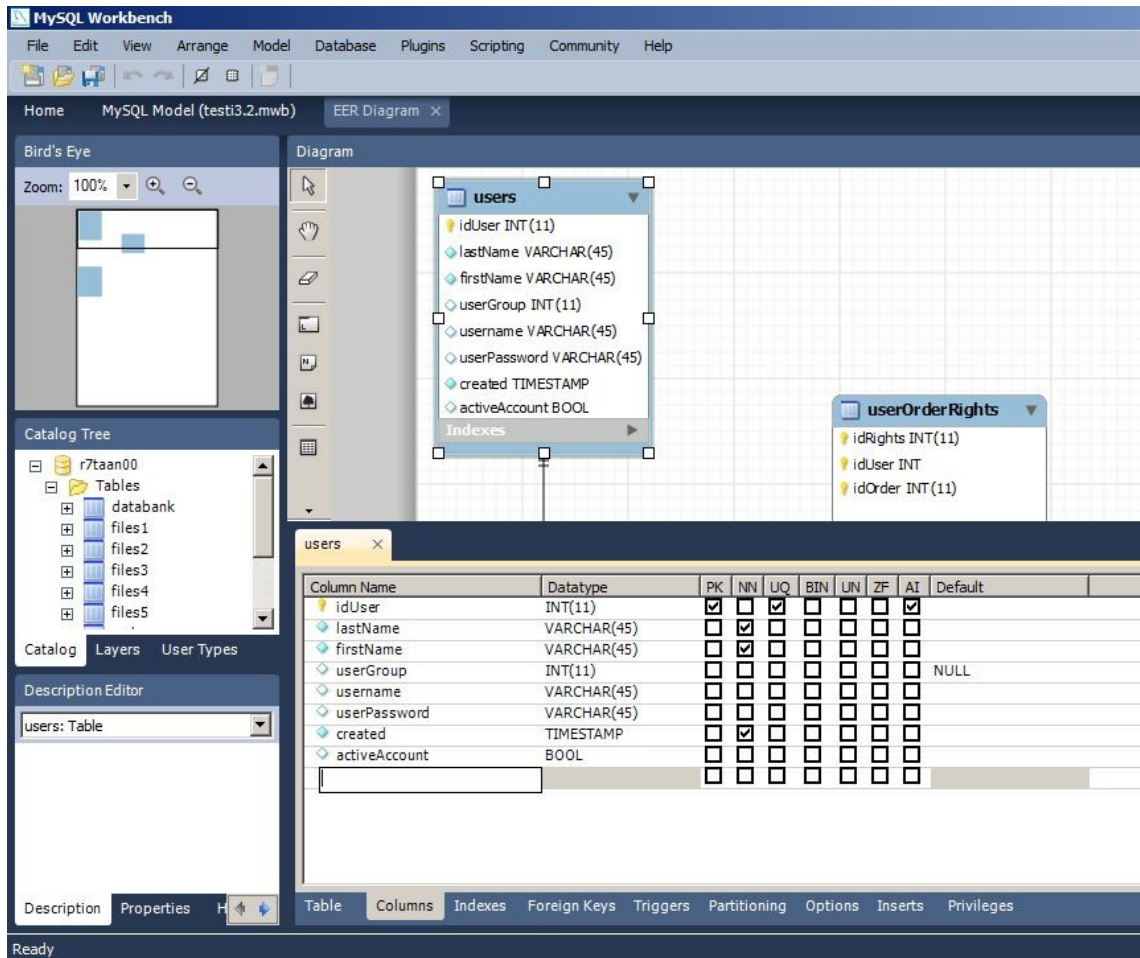
Aihe tähän opinnäytetyöhön syntyi Veljekset Paakki Oy:n tarpeesta päästä osittain eroon paperisista sopimuksista ja siirtymään käyttämään tietoverkkoon luotavia sähköisiä sopimus pohjia ja hallita näiden pohjalta tehtyjä sopimuksia verkkoselaimella web-käyttöliittymän kautta.

Opinnäytetyön aloitusta edeltävä kokemukseni relaatiotietokantaohjelmistoista ja PHP-ohjelmointikielestä oli lähes olematon muutamaa pientä poikkeusta lukuun ottamatta. Kokemukseni HTML-kielestä käsitti vain perusteet. Aiempi kokemukseni ohjelmoinnista koostui suurimmaksi osaksi C++-, Java ME/SE- ja BASIC-ohjelmointikielistä.

## 1.1 MySQL

Avoimen lähdekoodin MySQL-relaatiotietokanta valittiin tähän projektiin, koska se on tarjolla lähes kaikilla palvelintilaa tarjoavilla tahoilla ja käyttäminen on ilmaista GPL (GNU General Public License)-lisenssin mukaisesti. GPL-lisenssi muun muassa takaa veloituksetta oikeuden kenelle tahansa kopioida, muuttaa, käyttää ja jakaa eteenpäin ohjelmaa ja sen lähdekoodia. Tämä koskee myös kaupallisissa tarkoituksissa toimivia, kunhan lisenssin asettamia ehtoja noudatetaan. Muun muassa näistä syistä MySQL on erittäin suosittu valinta tietokannaksi useissa web-sovelluksissa ja palveluissa. Laajasti tunnettuina esimerkkeinä tästä voidaan mainita YouTube, Wikipedia, Google, Facebook, Yahoo!, Nokia ja Twitter. (Wikipedia. 2012, MySQL; Wikipedia. 2012, GNU General Public License; mysql.com. 2012, linkki About MySQL.)

MySQL-tietokannan hallinta tapahtuu joko komentorivillä tai graafisella käyttöliittymällä. Tämän opinnäytetyön tietokantatyökaluiksi valittiin MySQL Workbench 5.2 CE ja phpMyAdmin 3.4. MySQL Workbench CE on graafisen käyttöliittymän omaava työkalu tietokantojen suunnitteluun ja hallintaan.



KUVA 1. MySQL Workbench

PhpMyAdmin on web-käyttöliittymän kautta käytettävä MySQL-tietokannan hallintatyökalu.

The screenshot shows the phpMyAdmin interface for a MySQL database named 'mydemo'. The main table 'customers' is displayed with the following structure:

#	Nimi	Tyyppi	Aakkosjärjestys	Attribuutit	Tyhjä	Oletusarvo	Lisätiedot	Toiminnot
<input type="checkbox"/>	1 id	int(11)			Ei	None		Muokkaa Tuhoa
<input type="checkbox"/>	2 name	varchar(60)	utf8_unicode_ci		Ei	None		Muokkaa Tuhoa
<input type="checkbox"/>	3 address	varchar(60)	utf8_unicode_ci		Ei	None		Muokkaa Tuhoa
<input type="checkbox"/>	4 city	varchar(30)	utf8_unicode_ci		Ei	None		Muokkaa Tuhoa
<input type="checkbox"/>	5 state	varchar(10)	utf8_unicode_ci		Ei	None		Muokkaa Tuhoa
<input type="checkbox"/>	6 zip	int(11)			Ei	None		Muokkaa Tuhoa

Below the table structure, there is a summary section titled '+ Indeksit' with a sub-section 'Tiedot' containing two tables:

Levytilan käyttö		Rivitilastot	
Tyyppi	Käyttö	Tieto	Arvo
Tietoa	5.2 kt	Muoto	dynaaminen
Indeksi	2 kt	Aakkosjärjestys	utf8_unicode_ci
Yhteensä	7.2 kt	Kpl rivejä	96
		Rivin pituus ø	55
		Rivin koko ø	77 tavua
		Luotu	04.04.2012 klo 17:45
		Viimeksi päivitetty	04.04.2012 klo 17:45

KUVA 2. PhpMyAdmin-demo

Molemmat ovat avoimen lähdekoodin ohjelmistoja ja käyttö on ilmaista GPL-lisenssin mukaisesti (Mysql.com. 2012, linkit Downloads (GA) -> MySQL Workbench; Phpmyadmin.net. 2012, linkki License ).

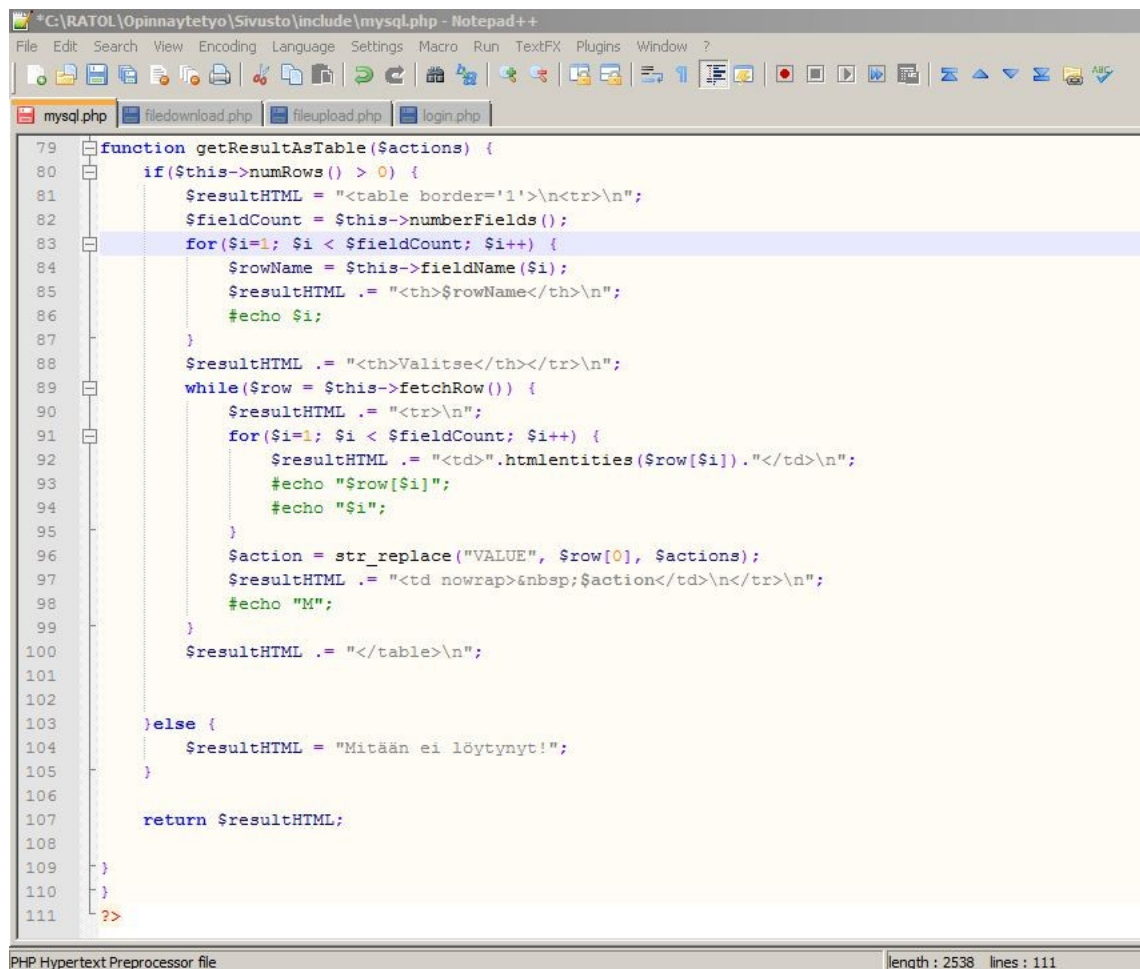
## 1.2 PHP

PHP (engl. Hypertext Preprocessor) valittiin tämän opinnäytetyön dynaamisten osien toteuttavaksi ohjelmointikieleksi. PHP on avoin ohjelmointikieli, joka on laajasti käytössä web-ympäristössä dynaamisten Internet-sivujen luontiin. PHP on tämän hetken suosituin palvelinpuolen ohjelmointikieli, noin 77.5% osuudella (W3techs.com. 2012). Tunnetuimpina käyttäjinä voidaan mainita esimerkiksi MediaWiki, Facebook, IRC-Galleria, phpBB, WordPress ja Travian (Wikipedia. 2012, PHP).

PHP:llä luotujen Internet-sivujen lukemiseen ja käyttämiseen riittää pelkkä Internet-yhteydellä ja verkkoselaimella varustettu tietokone tai mobiililaitte. Kaikki sivujen toiminnallisuus tapahtuu palvelimella PHP-tulkin toimesta ja tulokset näkyy verkkoselaimelle normaalissa HTML-muodossa. PHP tarjoaa myös kattavan ohjelmointirajapinnan MySQL-tietokannan käyttämiseen.



PHP:n kehitystyökaluksi valittiin Notepad++, joka on avoimeen lähdekoodin (GNU General Public License) perustuva tekstieditori Windowsille. Notepad++ tukee yli 50 ohjelmointikieltä tarjoten näihin syntaksin korostuksen, koodilohkojen tunnistamisen ja pienentämisen. (Notepad-plus-plus.org. 2012.)



```
*C:\RATOL\Opinnaytetyo\Sivusto\include\mysql.php - Notepad++
File Edit Search View Encoding Language Settings Macro Run TextFX Plugins Window ?
mysql.php filedownload.php fileupload.php login.php
79 function getResultAsTable($actions) {
80     if($this->numRows() > 0) {
81         $resultHTML = "<table border='1'>\n<tr>\n";
82         $fieldCount = $this->numberFields();
83         for($i=1; $i < $fieldCount; $i++) {
84             $rowName = $this->fieldName($i);
85             $resultHTML .= "<th>$rowName</th>\n";
86             #echo $i;
87         }
88         $resultHTML .= "<th>Valitse</th></tr>\n";
89         while($row = $this->fetchRow()) {
90             $resultHTML .= "<tr>\n";
91             for($i=1; $i < $fieldCount; $i++) {
92                 $resultHTML .= "<td>".htmlentities($row[$i])."</td>\n";
93                 #echo "$row[$i]";
94                 #echo "$i";
95             }
96             $action = str_replace("VALUE", $row[0], $actions);
97             $resultHTML .= "<td nowrap>&nbsp;$action</td>\n</tr>\n";
98             #echo "M";
99         }
100         $resultHTML .= "</table>\n";
101
102     }else {
103         $resultHTML = "Mitään ei löytynyt!";
104     }
105 }
106
107 return $resultHTML;
108
109 }
110 }
111 ?>
```

PHP Hypertext Preprocessor file | length : 2538 lines : 111

KUVA 3. Notepad++

### 1.3 HTML

HTML (engl. Hypertext Markup Language) on pääasiallinen kuvauskieli Internet-sivuille. Kuvauskieltä käytetään merkitsemään ja kuvaamaan Internet-sivun sisältöä niin, että verkkoselain osaa näyttää sen oikein.

HTML-editoriksi valittiin ilmainen (GNU General Public License) Nvu 1.0. Ohjelmalla voi luoda ja muokata verkkosivuja suoraan editoriin kirjoittamalla HTML-koodia tai graafisesti tekstinkäsittelyohjelmien tavoin.(Wikipedia. 2012, Nvu.)

tilaus [file:///.../printA.html] - Nvu

File Edit View Insert Format Table Tools Help

New Open Save Publish Browse Anchor Link Image Table Form Spell Print

Body Text

(no class) Variable Width

```

325.         <option value="1">UPK</option>
326.         <option value="2">SPK</option>
327.     </select>
328.     <br>
329. </td>
330.     <td style="white-space: nowrap;">
331.     <select name="a46">
332.         <option selected="selected"
333. value="0">&lt;Valitse&gt;</option>
334.         <option value="1">puolipy&ouml;r&auml;</option>
335.         <option value="2">kantikas</option>
336.     </select>
337.     <br>
338. </td>
339. <td style="white-space: nowrap;">
340.     <select name="a47">
341.         <option value="0">&lt;Valitse&gt;</option>
342.         <option value="1">RR22</option>
343.         <option value="2">RR23</option>
344.         <option value="3">RR32</option>
345.         <option value="4">RR33</option>
346.         <option value="5">RR750</option>
347.         <option value="6">RR29</option>
348.         <option value="7">RR37</option>
349.         <option value="8">RR11</option>
350.         <option value="9">muu</option>
351.     </select>
352.     <br>
353. </td>
354. </tr>
355. <tr>
356.     <td style="white-space: nowrap;"><small>SY&Ouml;KSYTORVI</small></td>
357.     <td style="white-space: nowrap;">
358. <hr>

```

Normal HTML Tags Source Preview

KUVA 4. Nvu

## **2 MÄÄRITELMÄ**

Tämän opinnäytetyön tavoitteena on luoda sähköisten sopimusten hallintajärjestelmä, jossa helpokäyttöinen käyttöliittymä. Järjestelmä tulee asentaa käyttövalmiina tilaajan osoittamalle julkiselle palvelimelle. Työhön sisältyy tietokantataulukoiden, käyttöliittymän ja vaatimusmäärittelyn mukaisten toiminnallisuuksien suunnittelu sekä toteutus. Myös järjestelmän käyttöönotto ja opastus kuuluvat työhön.

### **2.1 Tietokanta**

Tietokanta tulee rakentaa tavalla, joka mahdollistaa uusien ominaisuuksien tuomisen järjestelmään ilman suuria rakenteellisia muutoksia. Yksittäisen tiedon lisääminen, muuttaminen ja poistaminen tulee pystyä tekemään yhdestä paikasta. Sama tieto ei saa löytyä kuin yhdestä taulukosta.

### **2.2 Sisään- ja uloskirjautuminen**

Järjestelmän toiminnot ja tiedot on suojattava niin, että ulkopuolisilla ei ole niihin pääsyä. Päästäkseen sisään järjestelmään käyttäjän on syötettävä hänelle etukäteen jaettu henkilökohtainen käyttäjätunnus ja salasana sisäänkirjautumis-lomakkeeseen. Uloskirjautumisen jälkeen järjestelmään uudelleen sisään pääseminen ilman oikeaa käyttäjätunnusta ja salasanaa on estettävä.

### **2.3 Käyttäjätilit ja käyttäjien hallinta**

Käyttäjätiliin kuuluviin tietoihin tulee sisältyä käyttäjän käyttäjätunnus, salasana, koko nimi käyttäjäryhmä sekä tieto siitä, onko tili lukittu. Käyttäjäryhmiä tulee olla kolme erilaista omilla käyttöliittymillä ja erilaajuisilla käyttöoikeuksilla. Jokaisen käyttäjätilin on kuuluttava yhteen ja vain yhteen kolmesta käyttäjäryhmästä. Oma-aloitteinen järjestelmään rekisteröityminen ja käyttäjätilin luominen ei tule olla sallittua, vaan kaikki käyttäjätilien hallintaan kuuluvat toiminnot tulee sijoittaa pääkäyttäjän käyttöliittymään.

Lukuun ottamatta käyttäjätunnusta, kaikkia muita käyttäjätilin tietoja tulee olla mahdollista muuttaa käyttäjien hallinnasta. Käyttäjätilien poistaminen ei tule olla mahdollista web-

käyttöliittymän kautta, mutta ne on kuitenkin voitava lukita niin, että tunnuksilla ei pääse kirjautumaan sisään järjestelmään.

## **2.4 Käyttöliittymät**

Käyttöliittymän on oltava selkeä ja yksinkertainen kokonaisuus, ja sen opettelemisen ja käyttämisen tulee olla vaivatonta. Osalla tulevista loppukäyttäjistä kokemus tämän kaltaisista järjestelmistä on vaatimatonta tai voi puuttua kokonaan. Kaikki järjestelmän sisällä tapahtuva navigointi tulee pystyä suorittamaan hyperlinkkien ja painikkeiden avulla. Käyttäjän on aina heti pystyttävä näkemään, millä sivulla hän kulloinkin on.

Navigoinnin rungon tulee koostua sivun yläosaan sijoitetusta palkista, josta ilmenee käyttäjän koko nimi sekä käyttäjäryhmä mihin hän kuuluu. Palkin alaosaan on sijoitettava hyperlinkeistä koostuva navigointipuu, joka vie toimintoihin joihin käyttäjällä on oikeus. Käyttöliittymän graafisen ulkoasun suunnittelu ja toteutus jätetään tämän työn ulkopuolelle.

### **2.4.1 Asennuksen käyttöliittymä**

Käyttöoikeuksiltaan rajoittunein käyttöliittymä. Opinnäytetyön tilaajan asiakkaiden rakennustyömailla työskenteleville asentajille on luotava käyttöliittymä sopimusten lukemiseen ja tulostamiseen. Sopimusten kauppahintoja ja liitetiedostoja lukuun ottamatta kaikkien muiden tietojen tulee olla näkyvissä. Oikeudet sopimusten lukemiseen tulee voida asettaa sopimuskohtaisesti työnjohdon käyttöliittymän kautta. Oikeus asentajatilien luomiseen ja hallintaan tulee olla vain työnjohdon käyttäjätilin omaavilla henkilöillä.

### **2.4.2 Myynnin käyttöliittymä**

Rajoituksia sisältävä käyttöliittymä. Ei käyttäjien hallintaa eikä sopimuskohtaisten lukuoikeuksien hallintaa. Käyttäjällä on oltava oikeus luoda, hakea ja muokata kaikkia sopimuksia. Oikeuksiin tulee myös sisältyä vapaa sopimuskohtaisten liitetiedostojen ja urakan edistymistä kuvaavien muuttujien muokkaus. Oikeus myyntitilien luomiseen ja hallintaan tulee olla vain työnjohdon käyttäjätilin omaavilla henkilöillä.

### **2.4.3 Työnjohdon käyttöliittymä**

Pääkäyttäjä. Kaikki edellä mainitut toiminnot on oltava käytettävissä. Pääkäyttäjän oikeuksiin kuuluttava käyttäjätilien luonti, muokkaus ja lukitseminen sekä sopimusten poisto järjestelmän käyttäjien näkyvistä. Sopimuksien tulee kuitenkin haluttaessa aina olla palautettavissa eikä niitä voida poistaa muuten kuin suoraan tietokantaa muokkaamalla.

### **2.5 Sopimusohjelmat ja sopimusten hallinta**

Kaikista kolmesta paperisesta sopimusohjelmasta on luotava sähköiset vastineet järjestelmään. Sähköiset sopimusohjelmat on rakennettava käyttäen HTML-elementtejä ja niiden on vastattava ulkomuodoltaan ja ryhmittelyltään paperisia versioita, jotta niiden käyttö olisi sujuvaa kaikille käyttäjille.

Uuden sopimuksen tallentamisen jälkeen käyttäjä on ohjattava sopimuksen esikatselu sivulle, joka esittelee juuri luodusta sopimuksesta tärkeimpiä tietoja, seitsemän sopimusta koskevaa urakan tilaa ja näiden muokkaus mahdollisuuden sekä mahdollisuuden lisätä liitetiedostoja sopimukseen. Ainoastaan pääkäyttäjälle tarjotaan mahdollisuus sopimuksen poistamiseen järjestelmän käyttäjien näkyvistä. Vahingossa tapahtuvan sopimuksen poistamisen välttämiseksi on käyttäjältä aina kysyttävä tai saatava vahvistus jossain muodossa ennen kuin sopimus voidaan poistaa.

Tarvittaessa sopimukset on voitava suojata jälkepäin tapahtuville tahallisilta ja tahattomilta muokkauksilta. Uutta sopimusta luotaessa tai vanhaa muokattaessa on oltava mahdollisuus merkitä sopimus vahvistetuksi, mikä käytännössä tarkoittaa sitä, että asiakas on hyväksynyt kyseisen sopimuksen sisällön. Tällöin sopimuksen tiedot lukitaan eikä niitä lisätieto-kenttää lukuun ottamatta enään voida muokata. Sopimusten lukitusta ei voida poistaa muuten kuin suoraan tietokantaa muokkaamalla.

### **2.6 Sopimusten tulostus**

Kaikkien tulostettujen sähköisten sopimusten on vastattava ulkomuodoiltaan ja ryhmittelyiltään pääpiirteittäin alkuperäisiä paperisia versioita sopimusohjelmista. Tulostettujen sopimusten on aina mahdollista pystysuunnassa yhdelle A4-kokoiselle paperiarkille. Kirjasimen tulee olla selkeä ja helposti luettavissa.

## **2.7 Sopimusten hakutoiminto**

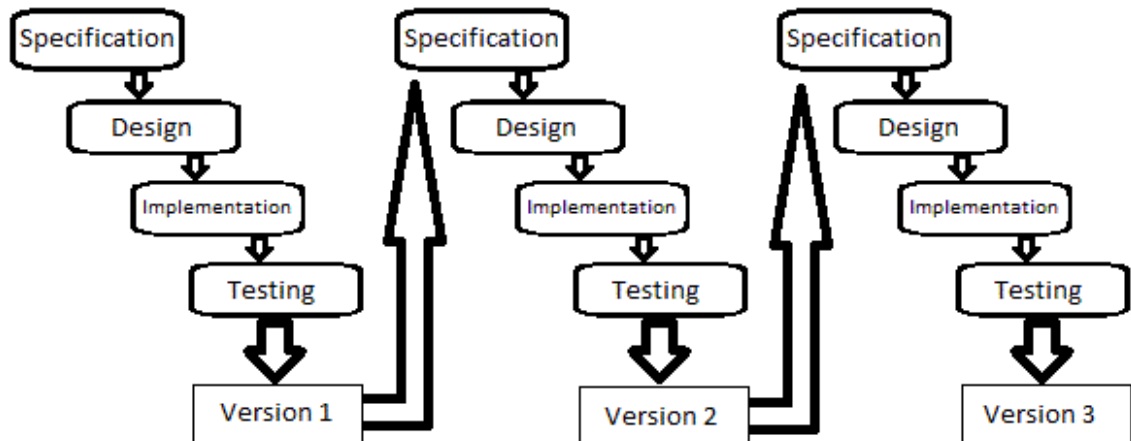
Työnjohdon ja myynnin käyttäjäryhmien jäsenille on rakennettava sopimustenhakutoiminto. Haut tulee voida suorittaa asiakkaan sukunimen ja paikkakunnan, myyjän nimen, sopimuksen luonnin kuukauden ja vuoden perusteella. Haun tulee onnistu myös silloin, kun asiakkaan nimi hakuparametrina käytetään sukunimen alkuosaa.

## 4 TOTEUTUS

Tässä luvussa kuvataan järjestelmän tekninen toteutus käytännössä. Luvussa käydään pintapuolisesti läpi myös projektin elinkaarimalli sekä työskentelymenetelmiä.

### 4.1 Evo-malli

Projektin elinkaarimalliksi valikoituu lähes huomaamatta evo-malli (engl. Evolutionary delivery). Evo-malli koostuu sarjasta toistuvia vesiputouksia, joista jokaisen tuloksena syntyy uusilla ominaisuuksilla kasvatettu järjestelmä. Yksittäinen vesiputous pitää sisällään määrittely-, suunnittelu-, toteutus- ja testausvaiheen joiden tuloksena uusi versio syntyy. (Haikala & Märijärvi 2002, 41.)



KUVA 5. Evo-malli

Evo-mallin ideaa noudattaen järjestelmää rakennettiin ensimmäisenä runko, jonka päälle kaikki vaaditut toiminnot tultiin erikseen rakentamaan, yksi kerrallaan. Jokainen uusi toiminto tuli olemaan lähes itsenäinen web-sovellus ja riippuvainen vain järjestelmän rungosta.

### 4.2 GET- ja POST-metodit

HTTP-protokollan GET- ja POST-metodit ovat kaksi erilaista tapaa siirtää tietoa käyttäjältä (engl. client) palvelimelle (engl. server). Molemmilla metodeilla pystytään saavuttamaan sama lopputulos, mutta väärä valinta näiden kahden väliltä voi johtaa odottamattomiin tai jopa vahingollisiin lopputuloksiin.

GET-metodi lähettää tiedon verkkoselaimelta palvelimelle URL-osoitteen (engl. Uniform resource locator) perään lisätyllä merkkijonolla. Esimerkiksi [www.esimerkki.fi/preview.php?idOrder=8](http://www.esimerkki.fi/preview.php?idOrder=8) lähettää palvelimella sijaitsevalle preview.php skriptille muuttujan idOrder arvolla 8. PHP-skripti voi tämän tiedon perusteella esimerkiksi hakea tietokantaan tallennetun sopimuksen, jonka id-numero on 8 ja tulostaa sen Internet-sivulle. Muuttujien määrää tai pituutta ei rajoita muu kuin URL-osoitteen enimmäispituus.

POST-metodi toimii lähes samalla tavalla kuin GET-metodi mutta lähetettävä tieto ei näy URL-osoitteen perässä eikä sen pituus ole rajoitettu. POST-metodia ei kuitenkaan voi käyttää muutoin kuin form-tagin sisällä olevien HTML-elementtien sisältämien tietojen lähettämiseen. Näitä elementtejä voivat olla esimerkiksi tekstikentät, valintaruudut, valintanapit, alusvetovalikot sekä painikkeet.

Järjestelmässä käytetään GET-metodia vain sellaisissa toiminnoissa joissa tietoa haetaan. POST-metodia taas sellaisissa tilanteissa joissa tietoa muutetaan. Tämä takaa sen, että käyttäjä ei esimerkiksi vahingossa suorita uudestaan PHP-skriptiä, joka muuttaa tai tuhoaa tietokannan taulukon sisältämiä tietoja.

### **4.3 Järjestelmän runko**

Järjestelmän rungon rakentaminen aloitettiin tietokannan suunnittelulla ja toteutuksella. Seuraavaksi runkoon lisättiin toiminnot sisään- ja uloskirjautumiseen sekä web-käyttöliittymän runko.

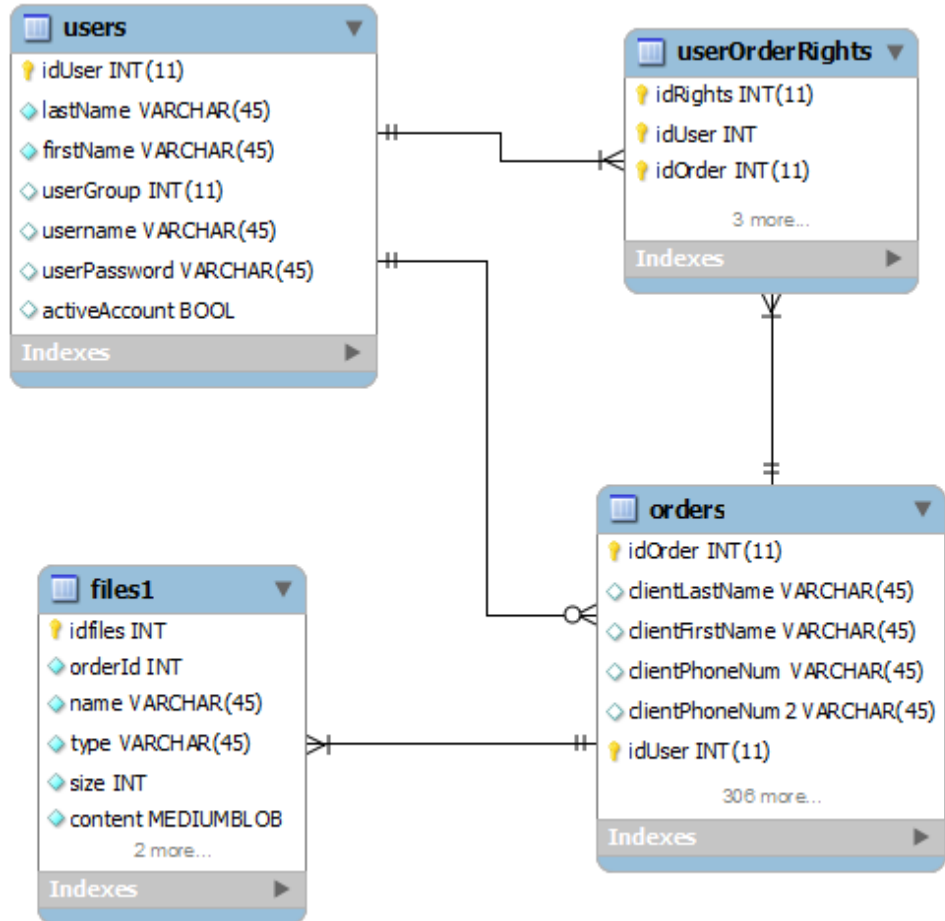
#### **4.3.1 Tietokanta**

Tietokannan suunnitteluvaihe alkoi, kun järjestelmän keskeisimmät toiminnot oli sovittu työn tilaajaosapuolen kanssa. Tietokanta suunniteltiin niin, että uusien toimintojen rakentaminen järjestelmään on mahdollista ilman suuria rakenteellisia muutoksia.

Tietokantaan luotiin users-, userOrderRights-, orders- ja files1-nimiset taulukot. Users-tilukko sisältää järjestelmän käyttäjien id-numeron, sukunimen, etunimen, käyttäjäryhmän, käyttäjätunnuksen, salasanan ja tiedon onko tili lukittu. UserOrdersRights-tilukko sisältää asentajakäyttäjäryhmään kuuluvien tiedot, mitä sopimuksia on oikeus lukea. Orders-tilukko sisältää luotujen sopimusten tiedot. Tietoihin kuuluu asiakkaan nimi- ja yhteystiedot sekä

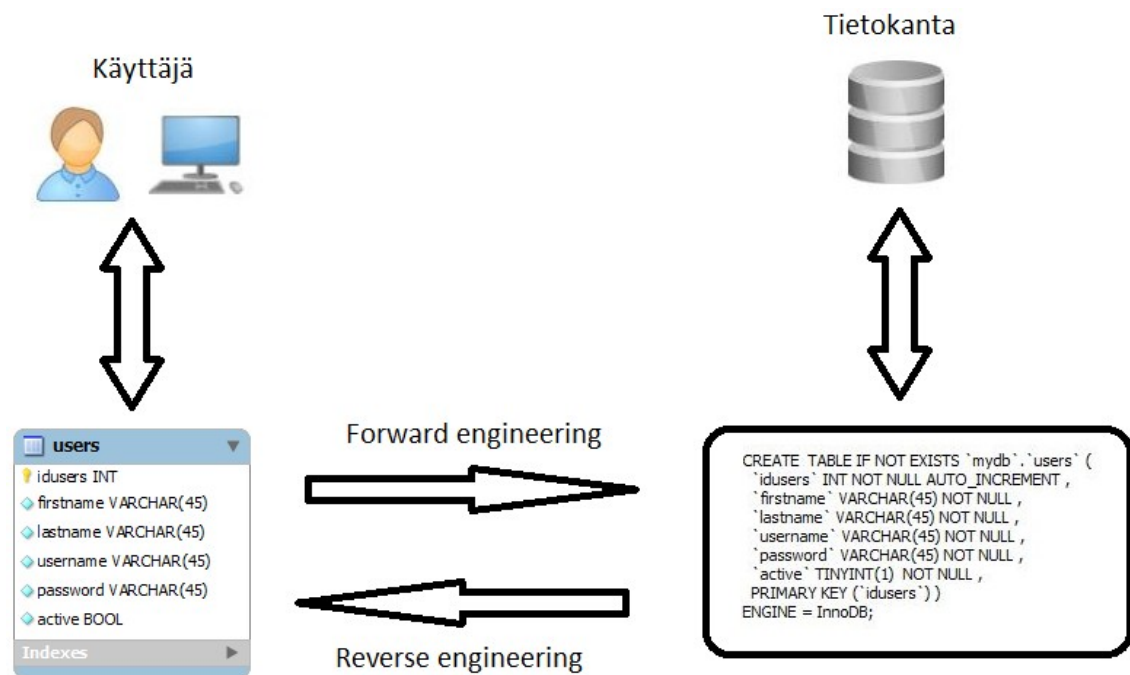


sopimukset yksityiskohdat joita voi olla reilusti yli 100 kappaletta. Taulukoiden välille luotiin myös yhteydet, eli tieto siitä miten taulukot liittyvät yhteen. Kuvassa alla on tietokannan ER-malli, josta taulukoiden väliset yhteydet käyvät selvästi ilmi.



KUVA 6. Tietokannan ER-malli

Tietokannan kaikki taulukot suunniteltiin ja luotiin MySQL Workbench-tietokannan hallintatyökalulla. Valmis tietokanta siirrettiin palvelimelle MySQL Workbench-eteenpäinmallinnustyökalulla (engl. Forward engineering). Työkalu mahdollistaa tietokannan suunnittelun ja toteutuksen täysin graafisesti. Käyttäjän ei periaatteessa tarvitse osata edes SQL-kielen alkeita. Työkalu osaa kääntää graafisesti luodun ER-mallin SQL-kielille ja luoda sen avulla tietokannan palvelimelle. Työkalulla voi myös takaisinmallintaa (Engl. Reverse engineering) jo olemassa olevan tietokannan. Seuraavassa havainnollistava kuva.



KUVA 7. Takaisin- ja eteenpäin-mallinnus

#### 4.3.2 Tietokantayhteyksien hallintaluokka

Kaikki järjestelmän tarvitsemat tietokanta-funktiot koottiin yhteen tiedostoon omaksi luokakseen. Jokaiseen näitä funktioita tarvitsevaan järjestelmän osaan eli PHP-tiedostoon tämä luokka tuodaan "include "<luokan\_nimi>.php"-käskyllä. Tällöin kaikkia sen sisältämiä funktioita voidaan kutsua sieltä, missä niitä kulloinkin tarvitaan.

Tietokantayhteyttä tarvittaessa kutsutaan connect()-funktiota, joka mysql\_connect()-funktiota kutsumalla avaa yhteyden tietokantaan. Onnistuneen yhteyden luonnin jälkeen valitaan tietokanta mysql\_select\_db-funktiolla. Alla yksinkertaistettu esimerkki.

```
function connect()
{
    $linkid=mysql_connect($host, $user, $passwd);
    mysql_select_db($database, $linkid);
}
```

Tietokannan sisältämien tietojen, hakemiseen, tallentamiseen ja päivittämiseen kutsutaan query()-funktia. Funktiokutsun argumentti sisältää halutun SQL-kyselyn, jonka mysql\_query()-funktio toteuttaa. Alla yksinkertaistettu esimerkki.

```
function query($query)
{
    $result=mysql_query($query, $linkid);
}
```

Mysql\_query-funktion tietokannasta noutama tieto on mahdollista muuttaa objektiksi, josta tiedot on helppo poimia käyttöön. Alla yksinkertaistettu esimerkki.

```
function fetchObject()
{
    $row=mysql_fetch_object($result);
    return $row;
}
```

Luokka sisältää myös useita muita tietokannan käyttöön liittyviä funktioita. Kaikki funktiot on esitelty alla olevassa taulukossa.

*TAULUKKO 1. Tietokanta-luokan sisältämät funktiot ja niiden toiminta*

<b>FUNKTIO</b>	<b>TOIMINTA</b>
connect()	Kutsuu mysql_connect()-funktia, joka avaa tietokanyteyden. Yhteyden luonnin jälkeen kutsuu mysql_select_db-funktia, joka valitsee tietokannan.
query()	Otaa argumentiksi SQL-kyselyn. Kutsuu mysql_query()-funktia SQL-kysely argumenttinä ja palauttaa mahdollisen tuloksen.
fetchObject()	Kutsuu mysql_fetch_object()-funktia, joka palauttaa mysql_query()-funktion antaman tuloksen objektina.

numberFields()	Palauttaa SQL-kyselyn tuloksen rivien lukumäärän
getResultAsTable()	Palauttaa SQL-kyselyn tuloksen HTML-taulukkona.

### 4.3.3 Sisään- ja uloskirjautuminen

Sisäänkirjautumis-lomake rakennettiin HTML-kielellä form-tagien sisällä. Käyttäjätunnus ja salasana välitetään POST-metodilla skriptille, joka tarkistaa niiden oikeellisuuden ja suorittaa sisäänkirjautumisen.

The image shows a login form with the title "Sisään kirjautuminen". It has two text input fields: "Käyttäjätunnus" and "Salasana". Below the fields is a button labeled "Kirjaudu".

KUVA 8. Sisäänkirjautumis-lomake

Skripti kutsuu query()-funktiota, joka suorittaa SQL-kyselyn tietokantaan. Jos käyttäjätunnusta ja salasanaa vastaava aktiivinen käyttäjätili löytyy haetaan istuntomuuttujiin käyttäjän id-numero, käyttäjäryhmä sekä etu- ja sukunimi. Lopuksi käyttäjä ohjataan sisään oman käyttäjäryhmän käyttöliittymään. Uloskirjautuminen tyhjentää kaikki istuntomuuttujat ja ohjaa käyttäjän takaisin sisäänkirjautumis-sivulle. Alla esimerkki sisäänkirjautumis-skriptistä.

```

$username=$_POST['username'];
$password=$_POST['password'];
$mysqldb->query("SELECT * FROM users WHERE username='$username' AND
password='$password' AND activeAccount='1'");
if($mysqldb->numRows()==0)
{
    echo "Käyttäjätunnus tai salasana väärä!";
}
else

```

```

{
    #käyttäjätunnusta ja salasanaa vastaavaa tili löytyi
    $_SESSION['kirjautunut']='1';
    $row = $mysqlldb->fetchObject();
    $_SESSION['kayttajaid']=$row->idUser;
    $_SESSION['sukunimi']=$row->lastName;
    $_SESSION['etunimi']=$row->firstName;
    $_SESSION['ryhma']=$row->userGroup;
    header("location:main.php");
}

```

#### 4.3.4 Web-käyttöliittymän runko

Jokaiselle käyttäjäryhmälle rakennettiin omat valikkopalkit, jotka sisältävät hyperlinkit toiminnallisuuksiin joihin käyttäjällä oikeus. Palkki tuodaan jokaiseen sivuun 'include "<valikon\_nimi>.php"-käskyllä. Alla esimerkki tästä.

```

if($_SESSION['ryhma'] == 1)
{
    include("johtomenu.php");
}elseif($_SESSION['ryhma'] == 2)
{
    include("myyjamenu.php");
}elseif($_SESSION['ryhma'] == 3)
{
    include("asentajamenu.php");
}

```

Samantyyllisellä if-rakenteella tarkistetaan myös jokaisen sivun alussa, onko käyttäjä kirjautunut sisään ja onko hänellä oikeus kyseiseen resurssiin. Jos yksikään ehdoista ei täyty, tästä ilmoitetaan ja tarvittaessa käyttäjä ohjataan sisäänkirjautumis-sivulle.

# TYÖNJOHTO

Antti Tapaus

---

[Uusi sopimus](#) | [Hae sopimuksia](#) | [Käyttäjien hallinta](#) | [Kirjaudu ulos!](#) |

---

[Käyttäjien hallinta->](#) | [Luo uusi käyttäjä](#) | [Näytä kaikki käyttäjät](#) |

---

KUVA 9. Työnjohdon käyttöliittymän valikkopalkki

## 4.4 Käyttäjätilit ja käyttäjien hallinta

Käyttäjätilien luontiin ja hallintaan rakennettiin yksinkertainen lomake. Lomake lähettää POST-metodilla uudet tai päivitettyt tiedot PHP-skriptille, joka SQL-kyselyllä luo uuden käyttäjätilin tai päivittää vanhan. Uutta käyttäjätiliä luodessaan skripti varmistaa, että käyttäjätunnus ei ole varattu. Lukuunottamatta käyttäjätunnusta ja id-numeroa, käyttäjätilin kaikkia tietoja voi muuttaa ja tilin voi myös lukita. Lukitseminen estää kyseisen käyttäjän sisäänkirjautumisen järjestelmään, mutta ei tuhoa tietoja. Käyttäjätilejä ei voi poistaa kuin tietokantaa muokkaamalla. Tämä siitä syystä, että kaikille luoduille sopimuksille on löydettävä tekijä, jonka perusteella sopimus voidaan hakea ja tunnistaa.

Luo uusi käyttäjätili	
Etunimi	<input type="text"/>
Sukunimi	<input type="text"/>
Käyttäjätunnus	<input type="text"/>
Salasana	<input type="password"/>
Salasana uudelleen	<input type="password"/>
Käyttäjärühmä	Työnjohto ▾
Käyttäjätili aktiivinen	Kyllä ▾
<input type="button" value="Luo tili"/>	

KUVA 10. Käyttäjätilin luontilomake

#### 4.5 Sopimuspohjat ja sopimusten hallinta

Kaikki kolme sopimuspohjaa luotiin Nvu:n graafisella HTML-editorilla. Pohjien runko rakennettiin päällekkäisistä HTML-taulukoista ja sisältö tekstikentistä, valintaruuduista, valintanapeista ja alasvetovalikoista. Pienin muutoksin samoja pohjia käytettiin myös sopimusten muokkaamiseen ja tulostamiseen tarkoitetuissa osissa sivustoa. Sopimuspohjat löytyvät liitteistä 1-3

Uuden tai päivitetyn sopimuksen tiedot välitetään POST-metodilla PHP-skriptille, joka siirtää ne SQL-kyselyllä tietokantaan. Vanhaa sopimus tietokannasta haettaessa PHP-skriptille ilmoitetaan GET-metodilla sopimuksen id-numero, jonka perusteella SQL-kysely noutaa sopimuksen. Noudetut tiedot muutetaan objektiksi `fetchObject()`-funktioilla ja sijoitetaan `echo`-käskeyillä oikeisiin kohtiin sopimusta. Sopimusten esikatselu-näkymä toimintoinen on myös toteutettu yllä mainitulla tavalla.

#### 4.6 Sopimusten haku

Sopimusten hakemiseen luotiin lomake, jolla haku voidaan suorittaa asiakkaan nimen, paikkakunnan, myyjän sekä luonti vuoden ja kuukauden perusteella. Hakukriteerit välitetään POST-metodilla SQL-kyselyn suorittavalle PHP-skriptille. Haun tulokset esitetään käyttäjälle `getResultAsTable()`-funktion luomassa taulussa. Funktio luo myös hyperlinkit haettuihin sopimuksiin. Jokainen linkki sisältää URL-osan lisäksi sopimuksen id-numeron (GET-metodi), jonka perusteella sopimuksen tiedot haetaan ja tulostetaan sopimus pohjaan.

Sopimusten haku

Asiakas	Myyjä	Paikkakunta	Kuukausi	Vuosi
<input type="text"/>	<Valitse>	<input type="text"/>	<Valitse>	<input type="text"/>

KUVA 11. Sopimusten hakulomake

## 5 TESTAUS

Tekijä testasi järjestelmää pääosin työn ohessa. Jokainen uusi toiminto testattiin heti valmistumisen jälkeen ja useaan otteeseen projektin edetessä.

### 5.1 Tietoturva

Sisäänkirjautumis lomakkeeseen yritettiin SQL-injektio nimisellä tekniikalla syöttää haitallisia SQL-kyselyjä. Näillä yrityksillä ei ollut mitään vaikutusta järjestelmään eikä sisään murtautuminen onnistunut.

Jokainen yksittäinen sivu käytiin läpi uloskirjautuneena ja pyrittiin pääsemään käsiksi toiminnallisuuksiin ja tietoihin. Apuna käytettiin myös toimiviksi tiedettyjä GET-metodeja. Vaatimusten mukaisesti kaikki sivut antoivat ilmoituksen sisäänkirjautumattomuudesta eikä tietoihin tai toiminnallisuuksiin päästy käsiksi.

### 5.2 Sopimustietojen paikkaansapitävyys

Lomakkeiden testaus toteutettiin järjestelmällisesti käyttäen apunaan verkkoselainta ja konsolilyhteyttä tietokantaan. Web-käyttöliittymän kautta luotiin sopimuksia. Sopimukset käytiin läpi suoraan tietokannasta ja tarkastettiin tietojen oikeellisuus ja sijoittuminen soluihin. Samat sopimukset noudettiin myös käyttöliittymän kautta muokattavaksi ja samat tarkistukset suoritettiin.

Tarkistuksissa löytyi useita virheitä tietojen puuttumisena ja sijoittumisena väärään paikkaan. Kaikki virheet korjattiin ja testit suoritettiin vielä kerran uudelleen.

### 5.3 Sopimusten hakutoiminto

Tietokantaan luotiin konsolilyhteyden kautta testisopimuksia eri tiedoilla, joissa vaihtelivat asiakkaan sukunimi, kotikunta sekä myyjän nimi ja sopimuksen luontipäivämäärä. Haku suoritettiin jokaisen sopimuksen kohdalla erikseen kaikkia haun niitä eri kriteeriyhdistemiä käyttäen, joilla sopimusten oletettiin löytyvän. Myös hakuja, joissa hakukriteerit vastasivat useampia sopimuksia, suoritettiin.



Testitapaukset suoritettiin kaksi kertaa yllä mainitulla tavalla. Missään vaiheessa ei havaittu virheellistä toimintaa ja kaikki testitapaukset antoivat positiivisen tuloksen. Hakutoiminto oli useasti käytössä myös muita järjestelmän osia luotaessa ja testatessa.

#### **5.4 Käyttöliittymä ja navigointi**

Käyttöliittymää ja navigointia testattiin tekemällä epätavallisia liikkeitä ja seurattiin mitä sivuilla ja tietokannassa tapahtuu. Joissain tilanteissa huomattiin, että takaisin-nappia painettaessa edellisellä sivulla suoritettu SQL-kysely toistui. Mitään havaittavaa vahinkoa, häiriötä, tietojen tuhoutumista tai muuttumista tämä ei kuitenkaan aiheuttanut. Käyttäjiä kehoitettiin navigoimaan ainoastaan järjestelmän hyperlinkkien ja painikkeiden kautta, ei verkkoselaimen takaisin- ja eteenpäin-painikkeilla.

## 6 JATKOKEHITYSMÄHDOLLISUUDET

Järjestelmään runkoon voi periaatteessa rajattomasti lisätä toimintoja. Alla on lista mahdollisista uusista ominaisuuksista. Osa on työn tilaajan esittämiä ja osaan idea on syntynyt opinnäytetyön tekijän toimesta

-refaktoroiminen Drupalille tai CSS:lla luotava graafinen ulkoasu.

-sopimusten sähköinen allekirjoitus.

-intranet.

-työmaa- ja työntekijäkohtainen työaikaseuranta.

-varauskalenteri myyjien avuksi.

-pienille mobiililaitteille optimoidut sivut.

## 7 YHTEENVETO

Jokaiselle sopimus pohjalle löytyy nyt omat sähköiset vastineensa. Kaikki vaaditut toiminnallisuudet sopimusten luomiseen, muokkaamiseen ja hakemiseen ovat toimivia ja käytettävissä. Käyttäjäryhmien ja käyttäjätilien hallinta sekä sisään- ja uloskirjautuminen on myös toteutettu vaatimusten mukaisesti ja otettu käyttöön. Projektin lopputuloksena syntyi toiminnallisuuksiltaan vaaditun kaltainen järjestelmä.

Koko projekti alusta loppuun asti jatkuvaa uuden oppimista ja omaksumista. Muiden ohjelmointikielien osaaminen osoittautui alusta lähtien suureksi avuksi uutta ohjelmointikieltä opiskeltaessa. Kaikkien aihealueiden opiskelu tapahtui suurimmaksi osaksi alan kirjallisuuden ja Internet-materiaalin avulla. Myös yksittäisten henkilöiden antamat neuvot ja samankaltaisten valmiiden ohjelmistojen lähdekoodin tutkiminen auttoi muutamissa tilanteissa.

Oleellinen osa projektia oli muuttuvat vaatimukset työn tilaajalta ja keskeneräisen järjestelmän toimivien osien esittely palaverissa. Tämä loi hyvää pohjaa ymmärrykselle siitä, kuinka asiantuntijana on tasapainoteltava asiakkaiden toiveiden ja käytännön toteutusmahdollisuuksien välissä.

## LÄHDELUETTELO

Haikala, I. & Märijärvi, J. 2002. Ohjelmistotuotanto. 8. uudistettu painos. Helsinki: SATKU-Kauppakaari.

MySQL. 2012. MySQL. Hakupäivä 5.4.2012 <http://mysql.com/about>.

MySQL. 2012. MySQL Workbench. Hakupäivä 5.4.2012 <http://www.mysql.com/downloads/> .

Notepad++. 2012. Notepad++. Hakupäivä 6.4.2012 <http://notepad-plus-plus.org> .

Phpmyadmin.	2012.	Lisenssi.	Hakupäivä	5.4.2012
<a href="http://www.phpmyadmin.net/home_page/license.php">http://www.phpmyadmin.net/home_page/license.php</a> .				

W3techs. 2012. PHP. Hakupäivä 10.4.2012 <http://w3techs.com/>.

Wikipedia 2012. MySQL. Hakupäivä 4.4.2012 <http://en.wikipedia.org/wiki/MySQL>.

Wikipedia. 2012. GNU Hakupäivä 4.4.2012 <http://en.wikipedia.org/wiki/GNU>.

Wikipedia. 2012. PHP. Hakupäivä 6.4.2012 <http://en.wikipedia.org/wiki/PHP>.

Wikipedia. 2012. Nvu. Hakupäivä 10.4.2012 <http://en.wikipedia.org/wiki/Nvu>.





