



**SAVONIA**

# **Sähköinen varausjärjestelmä**

**Teemu Tiainen**

Opinnäytetyö

---



Koulutusala Tekniikan ja liikenteen ala	
Koulutusohjelma Tietotekniikan koulutusohjelma	
Työn tekijä(t) Teemu Tiainen	
Työn nimi Sähköinen varausjärjestelmä	
Päiväys 9.2.2012	Sivumäärä/Liitteet 34
Ohjaaja(t) Lehtori Jussi Koistinen, Lehtori Sami Lahti	
Toimeksiantaja/Yhteistyökumppani(t) Savonia-AMK	
Tiivistelmä <p>Tämän opinnäytetyön aiheena oli tehdä sähköinen varausjärjestelmä Savonia-AMK:n Ympäristötekniikan opetus- ja tutkimusyksikön tiloille ja laitteille. Järjestelmästä tuli tehdä Internet-sivusto, jonka avulla varaukset olisivat helposti hallinnoitavissa. Varauksien lisäksi järjestelmässä tulisi olla selkeä tilakalenteri, laiterekisteri, huoltotietojen kirjanpito sekä laitteiden käyttöhistorian raportointi.</p> <p>Työn ensimmäinen vaihe oli tehdä toimeksiantajan tekemät suunnitelmat valmiiksi, jotta sivuston ohjelmointiosuus voisi alkaa. Suunnitelmien muodostuessa rakentui samalla sivuston käyttämä tietokanta, joka toteutettiin MySQL-tietokantana.</p> <p>Työn toinen vaihe oli ohjelmoida ensimmäisen vaiheen suunnitelmien mukainen web-sivusto. Sivujen ohjelmointikielinä käytettiin HTML, PHP sekä JavaScript -kieliä ja sivuston ulkoasu määriteltiin omaan CSS-tiedostoon. Sivujen toimivuutta testattiin jatkuvasti testipalvelimen avulla. JavaScriptin lisäksi varauslomakkeen luonnin apuna käytettiin jQuery-kirjaston valikko- ja kalenteriominaisuuksia. Lopuksi valmis sivusto oli tarkoitus asentaa sille varatulle palvelimelle.</p>	
Avainsanat Internet, varaus, HTML, MySQL	

Field of Study Technology, Communication and Transport			
Degree Programme Degree Programme in Information Technology			
Author(s) Teemu Tiainen			
Title of Thesis Electrical Booking System			
Date	9 February 2012	Pages/Appendices	34
Supervisor(s) Mr Jussi Koistinen, Lecturer, Mr Sami Lahti, Lecturer			
Client Organisation/Partners Savonia University of Applied Sciences			
<p>Abstract</p> <p>The purpose of this final project was to make an electrical booking system for the Department of Environmental Technology at Savonia University of Applied Sciences. The system was to be a website that contains all laboratory rooms and devices and its purpose was to clarify and ease the maintaining of those bookings. Other requirements for the system were a mode-specific calendar, registry for the devices, bookkeeping of maintenance and usage reports of the devices.</p> <p>Firstly, the preliminary plans for the system were finalized in order to start the programming part. At the same time with the finalizing process, the purpose was to plan and build the database for the website where all the data would be stored. The database was made by MySQL.</p> <p>Secondly, the website was programmed using the plans that had been made in the first phase. The functionality of the site was made with the HTML, PHP, JavaScript scripting languages. Some features were made with the jQuery JavaScript library and the appearance details were defined in the CSS file. The completed code was installed on the web-server.</p>			
Keywords Internet, booking, HTML, MySQL			

## SISÄLTÖ

1	JOHDANTO.....	6
2	KÄYTETYT TEKNIIKAT.....	7
2.1	HTML .....	7
2.2	CSS.....	8
2.3	PHP.....	9
2.4	JavaScript.....	10
2.5	jQuery.....	11
2.6	SQL ja MySQL .....	12
3	VARAUSJÄRJESTELMÄ.....	14
3.1	Järjestelmän perusominaisuudet .....	14
3.1.1	Sivurunko.....	14
3.1.2	Käyttäjryhmät ja oikeudet .....	15
3.2	Järjestelmän toiminnot.....	15
3.2.1	Kalenteri.....	15
3.2.2	Laiterekisteri.....	17
3.2.3	Laitekortti.....	18
3.2.4	Huoltotoiminnot .....	19
3.2.5	Varauksen teko .....	20
3.2.6	Varausten hallinta .....	24
3.3	Järjestelmän ylläpitotoiminnot.....	25
3.3.1	Järjestelmän tietojenhallinta.....	25
3.3.2	Raportit.....	26
3.4	Tietojen tallentaminen .....	27
3.5	Hylätyt ominaisuudet .....	27
3.5.1	Automaattinen uloskirjautuminen .....	28
3.5.2	Liittäminen Savonian lukujärjestyksiin .....	29
3.5.3	Raporttien valmiit tulostusversiot.....	29
4	TIETOKANTA.....	30
5	TARKASTELU.....	32
	LÄHTEET .....	34

## 1 JOHDANTO

Internet-teknologiaa sovelletaan yhä enemmän palveluiden tarjonnan välineenä. Lähes jokaisella kaupalla on jo katalogit sähköisessä muodossa ja uutiskirjeitä lähetetään sähköpostina normaalien paperiversioiden lisäksi. Myös ajanvaraukset erilaisiin palveluihin alkavat olla nykypäivää, joten tulevaisuudessa puhelinvaraukset voivat hyvinkin olla häviävä palvelumuoto.

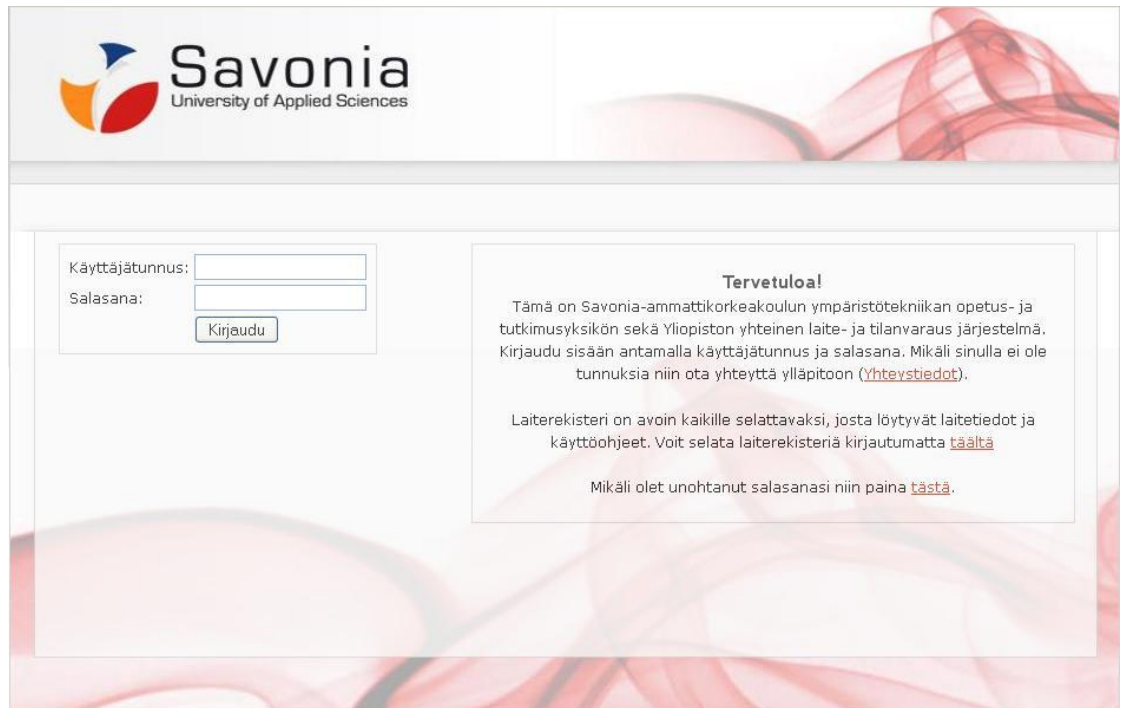
Tämän opinnäytetyön tarkoitus on poistaa epäselvät paperivaraukset laboratorioiden tiloista ja laitteista ja koota ne yhtenäiseksi ja selkeäksi kokonaisuudeksi. Koska varauksien teko oli mahdollistettava myös työtilojen ulkopuolelta, tuli järjestelmästä tehtävä Internet-sivusto.

Työstä oli tehty hyvin karkea alkusuunnitelma, joten ensimmäisenä toimena oli tutkia, täydentää ja viimeistellä tämä suunnitelma valmiiksi. Suunnitelman kehittyessä oli mahdollisuus samalla rakentaa pohjaa sivuston tarvitsemalle tietokannalle.

Varausjärjestelmän tärkeimpiä toimintoja varauksien tekemisen lisäksi on lukujärjestystyylinen tilakalenteri, laiterekisteri, laitteiden huoltotoimenpiteiden kirjanpito sekä käyttöhistorian raportointi halutuilla ehdoilla. Näiden toimintojen lisäksi sivuston tulisi näyttää Savonian web-sivujen ulkoasumäärittysten mukaiselta. Työn valmistuttua se asennettaisiin Savonian tietohallinnon hankkimalle palvelimelle.

## 2 KÄYTETYT TEKNIIKAT

Tässä luvussa esitellään varausjärjestelmän rakentamiseen käytetyt tekniikat. Tekniikat on esitelty lyhyesti, tarkoituksena antaa niistä taustatietoa sekä esimerkeillä selvittää, mihin tarkoitukseen niitä on työssä tarvittu käyttöä.



Kuva 1. Varausjärjestelmän etusivu

### 2.1 HTML

HTML on lyhenne sanoista hypertext markup language, joka on avoimesti standardoitu kuvauskieli hypertekstin kuvaamiseen. Ohjelmoinnissa on käytössä joukko erilaisia elementtejä, joilla luodaan haluttuja kokonaisuuksia, kuten taulukoita ja listoja. Pääsääntöisesti HTML-kieltä käytetään web-sivujen ohjelmointiin<sup>4</sup>.

Työssä HTML-kielellä on ohjelmoitu sivupohjat. Suoraa HTML-koodia on käytetty lähinnä sivujen jakamiseen lohkoiksi. Näihin lohkoihin muodostetaan niiden näyttämät toiminnot dynaamisesti PHP:llä. Seuraavalla koodilla on toteutettu Kuva 1 näkyvän varausjärjestelmän etusivun sivupohja.

```

<!DOCTYPE HTML>
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title>Kirjautumissivu</title>
<link rel="stylesheet" type="text/css" href="tyyli.css" />
<script src="scriptit.js" type="text/javascript"></script>
</head>
<body>
<div id="pohja">
  <div id="header">

    </div>
    <div id="kontaineri">
      <div id="sisalto">
        <div id="login_data">
          <?php in-
            clude("main_login.php") ?>

          </div>
        </div>
      </div>
    <div id="footer">
      <p><br /><br /><br /><br />
      />Ympäristötekniikan opetus- ja tutkimusyksikkö PL 6 (Microkatu 1C)
      70201 KUOPIO, PUH (017) 255 6000, <a
      href="http://portal.savonia.fi/amk/tktoiminta/teknymp/ymparistotekniika">ympäristötekniikka.fi</a></p>
    </div>
  </div>
</body>
</html>

```

## 2.2 CSS

CSS on lyhenne sanoista cascading style sheets, ja se on erityisesti Internet-sivuja varten kehitetty tyyliohjeiden laji. Yleensä sivustoilla käytetään yhtä CSS-tiedostoa, johon on kerätty kaikki tarvittavat tyylimäärittelyt. Määrittelyjä on mahdollista sisältää suoraan HTML-koodiin mutta tätä tapaa hyvin harvoin käytetään, koska tyylien muokkaaminen on tällöin työläämpää ja usein hankalampaa kuin CSS-tiedoston muokkaaminen. Varausjärjestelmässä kaikki tyylimäärittelyt on tehty yhteen CSS-tiedostoon. Seuraavalla CSS-tyylimäärittelyillä on toteutettu Kuva 1 näkyvän kirjautumissivun yläpalkki.

```

#header
{
    width:100%;
    float:left;
    height:190px;
    background-image:url(images/header_eng.jpg);
    border-right:1px solid #D5D5D5;
    border-left:1px solid #D5D5D5;
    border-top:1px solid #D5D5D5;
}

```



```
#header img
{
    width:100%;
    float:left;
}
```

### 2.3 PHP

PHP on lyhenne sanoista hypertext preprocessor, ja se on ohjelmointikieli, jota käytetään dynaamisten web-sivujen luontiin. Ohjelmointikielen lisäksi PHP-ympäristössä on laaja luokkakirjasto. Tunnettuja PHP:llä ohjelmoituja web-sivustoja ovat muun muassa Facebook (sosiaalisen median sivusto), Moodle (avoimeen lähdekoodiin perustuva opetussivusto) ja MediaWiki (esimerkiksi wikipedia)<sup>3</sup>.

Varausjärjestelmässä PHP:tä on käytetty sivujen dynaamisuuden tuottamiseen sekä yhteyden luomiseen järjestelmän käyttämään tietokantaan. Lisäksi PHP:n tarjoamat valmiit funktiot ovat olleet käytännölliset aikamääritysten tekemiseen sekä tietojen lähettämiseen sivulta toiselle. Järjestelmän PHP-toiminnot on ohjelmoitu omiin PHP-tiedostoihin, jotka liitetään sivuihin seuraavanlaisesti.

```
<?php include("main_login.php") ?>
```

Seuraavalla koodilla on toteutettu Kuva 1 näkyvät käyttäjätunnuksen ja salasanan syöttämiseksi tarkoitetut kentät sekä niiden vieressä oleva tervehdysteksti.

```
<?php
echo'<div id="login">
    <form action="checklogin.php" method="post">
        <table>
            <tr>
                <td>
                    Käyttäjätunnus:
                </td>
                <td>
                    <input
type="text" name="kt"></input>
                </td>
            </tr>
            <tr>
                <td>
                    Salasana:
                </td>
                <td>
                    <input
type="password" name="salasana"></input>
                </td>
            </tr>
            <tr>
                <td>
                </td>
            </tr>
        </table>
    </form>
</div>
```

```

                                <td>
                                <input
type="submit" name="ok" value="kirjaudu"></input>
                                </td>
                                </tr>
                                </table>
                                </form>';
                                if(isset($_GET["virhe"]))
                                {
                                    echo'<p>virheellinen kirjautuminen!!!</p>';
                                }
                                if(isset($_GET["aika"]))
                                {
                                    echo'<script>
                                        alert("Varausaika umpeutui");
                                        window.location = "login.php";
                                    </script>';
                                }
                                echo'</div>
                                <div id="login_teksti">
                                    <p><h3>Tervetuloa!</h3>Tämä on Savonia-
                                ammattikorkeakoulun ympäristötekniikan opetus- ja tutkimusyksikön
                                sekä Yliopiston yhteinen laite- ja tilanvaraus järjestelmä. Kirjaudu
                                sisään antamalla käyttäjätunnus ja salasana. Mikäli sinulla ei ole
                                tunnuksia niin ota yhteyttä ylläpitoon (<a
                                href="yhteystiedot.php">Yhteystiedot</a>).<br /><br />Laiterekisteri
                                on avoin kaikille selattavaksi, josta löytyvät laitetiedot ja käyttö-
                                ohjeet. Voit selata laiterekisteriä kirjautumatta <a
                                href="laite_lista.php">täältä</a><br /><br />Mikäli olet unohtanut
                                salasanasasi niin paina <a href="reset.php">tästä</a>.</p>
                                </div>';
                                ?>

```

## 2.4 JavaScript

JavaScript on pääasiassa web-ympäristössä käytettävä komentosarjakieli. JavaScript, PHP:n ohella, mahdollistaa web-sivujen dynaamisuuden luontia. JavaScriptiä voi myös hyödyntää erilaisissa tarkistustilanteissa ilman web-sivun uudelleen lataamista, joka olisi PHP:llä tehtynä pakollista. Kuten CSS:ssäkin, JavaScriptit on yleensä kerätty yhteen JavaScript-tiedostoon, josta ohjelmoituja toimintoja on helppo kutsua. Tarvittaessa JavaScript-koodia on mahdollista lisätä suoraan HTML-koodin yhteyteen.

Varausjärjestelmässä JavaScriptillä on luotu erilaisia ilmoitusikkunoita, tarkistettu muun muassa kellonaikojen pitävyyttä ja luotu pieniä ikkunoita. Seuraavalla koodilla avataan JavaScript-funttiolla uusi ikkuna, joka lataa varaus\_tiedot.php-tiedostoon ohjelmoitun sivun. Ikkuna on tehty tietynkokoiseksi, sen kokoa ei voi muuttaa ja siitä puuttuvat kaikki työkalu- ja valikkopalkit, jotka selaimessa on käytössä.

```
function naytaLaitteet(id)
{
    window.open("varaus_tiedot.php?id="+id+"", "Varaustiedot",
    "status=1,width=550,height=350,resizable=0,toolbars=0,menubar=0,
    scrollbars=1");
}
```

## 2.5 jQuery

jQuery on kaikille avoimen lähdekoodin JavaScript-kirjasto, joka on tarkoitettu kaikille Internet-selaimille. jQuery on noussut suosituksi web-sivujen ohjelmoinnin apuvälineeksi sen helposti ymmärrettävän syntaksin ansiosta. Erityisesti toimintojen käsittely, animaatioiden tekeminen, DOM-elementtien valitseminen ja Ajax-sovellusten toteutus on jQueryllä helposti toteutettavissa<sup>1</sup>.

Varausjärjestelmässä jQueryn animaatio- ja kalenteriominaisuuksia on hyödynnetty varauskaavakkeen toimintojen rakentamiseen. jQuery-kirjasto on tallennettava erilliseen JavaScript-tiedostoon, joka linkitetään web-sivuun. Myös monet julkiset palvelimet, kuten Google ja Microsoft, tarjoavat suoraa linkkiä jQuery-kirjastoon, jolloin erillistä tiedostoa ei tarvita. Omaan tiedostoon on kuitenkin jQueryn sivuilla mahdollista muokata esimerkiksi valmis värimaailma vastaamaan sivustoa, jota on ohjelmoimassa. Varausjärjestelmään on muokattu jQueryn värit vastaamaan järjestelmään käytettyä väritystä.

Seuraavilla esimerkeillä esitellään, kuinka varausjärjestelmän varauskaavakkeen käyttämä kalenteri on ohjelmoitu. Ensiksi jQueryn kalenteri alustetaan suomeksi seuraavalla koodilla:

```
jQuery(function($){
    $.datepicker.regional['fi'] = {
        closeText: 'Sulje',
        prevText: '&laquo;Edellinen',
        nextText: 'Seuraava&raquo;',
        currentText: 'T&uuml;n&uuml;n&uuml;n',
        monthNames: ['Tammikuu', 'Helmi-
        kuu', 'Maaliskuu', 'Huhtikuu', 'Toukokuu', 'Kes&uuml;kuu',
        'Hein&uuml;kuu', 'Elokuu', 'Syyskuu', 'Lokakuu', 'Marraskuu',
        'Joulukuu'],
        monthNamesShort: ['Tam-
        mi', 'Helmi', 'Maalis', 'Huhti', 'Touko', 'Kes&uuml;',
        'Hein&uuml;', 'Elo', 'Syys', 'Loka', 'Marras', 'Joulu'],
        dayNamesShort: ['Su', 'Ma', 'Ti', 'Ke', 'To', 'Pe', 'Su'],
        dayNames: ['Sunnun-
        tai', 'Maanantai', 'Tiistai', 'Keskiviikko', 'Torstai', 'Perjantai', 'Lauan-
        tai'],
        dayNamesMin: ['Su', 'Ma', 'Ti', 'Ke', 'To', 'Pe', 'La'],
        weekHeader: 'Vk',
        dateFormat: 'dd.mm.yy',
        firstDay: 1,
```

```

        isRTL: false,
        showMonthAfterYear: false,
        yearSuffix: '';
    $.datepicker.setDefaults($.datepicker.regional['fi']);
});

```

Kalenterin funktio on seuraavanlainen.

```

$( "#kerta_pvm" ).datepicker({
    showWeek: true,
    firstDay: 1,
    minDate: 0,
});

```

Kalenterin saa toimimaan ainoastaan tekstikenttään ja kalenterin funktiossa on määriteltä, että tekstikentän id täytyy olla kerta\_pvm. Kalenteri asetetaan tekstikenttään seuraavalla HTML-koodilla:

```
<input type="text" name="pvm" id="kerta_pvm" readonly="readonly" />
```

## 2.6 SQL ja MySQL

SQL on lyhenne sanoista structured query language. SQL on standardoitu kyselykieli, jolla voi tehdä hakuja, muutoksia ja lisäyksiä relaatiotietokantaan. Tärkeimpinä käytökäskyjä ovat SELECT, UPDATE, INSERT ja DELETE. Tärkeimmät määrittelykäskyt ovat CREATE TABLE ja CREATE VIEW. Myös ALTER-käsky on hyödyllinen, mikäli esimerkiksi taulujen sarakkeiden lukumäärää on tarve jälkeinpäin muokata.

MySQL on relaatiotietokantaohjelmisto, joka on saatavilla joko vapaalla GNU GPL-lisenssillä tai kaupallisella lisenssillä. MySQL:n noudattaa SQL-standardia mutta siinä on myös omia ominaisuuksia, jotka eivät ole yhteensopivia standardin kanssa. MySQL-ohjelmistolla on helppo luoda uusi tietokanta ja hallita tietokannan sisältöä. Ilmaisen ohjelman MySQL tarjoaa hyvän alustan web-sivuston ja tietokannan yhteyden testaamista varten<sup>2</sup>.

Varausjärjestelmän tietokanta on toteutettu MySQL-ohjelmistolla käyttäen SQL-kielen käskyjä. Sekä testipalvelin että virallinen palvelin käyttävät MySQL-ohjelmaa, joten työssä ei tarvinnut ottaa huomioon tietokantaohjelmistojen erilaisuuksia. Varausjärjestelmän SQL-käskyt tapahtuu PHP:n tarjoamilla valmiilla MySQL-funktioilla. Esimerkiksi varausjärjestelmän laiterekisteristä kaikki laitteet haetaan seuraavalla hakulauseella.

```
SELECT laite.id, laite.laite as nimi, laite.malli, sijainti.tila FROM  
laite INNER JOIN sijainti ON laite.sijainti_id = sijainti.id WHERE  
laite.status=2 ORDER BY sijainti.tila, laite.laite ASC;
```

### 3 VARAUSJÄRJESTELMÄ

Varausjärjestelmää suunnitellessa kävi mielessä, että työhön voisi löytyä valmiita pohjia avointen lähdekoodien joukosta. Päädyin kuitenkin ratkaisuun ohjelmoida varausjärjestelmä alusta asti. Merkittävin syy päätökseen oli epäily, että vieraan koodin tutkiminen ja muokkaaminen olisi hitaampaa kuin oman koodin tuottaminen.

Varausjärjestelmä on tarkoitettu toimimaan täydellisesti Internet Explorer -selaimella mutta se on pyritty ohjelmoimaan lähes samaan toiminnallisuuteen myös muilla Internet selaimilla. Toiminnallisuus näissä eroaa lähinnä joidenkin JavaScriptien vaihtelevalla toiminnallisuudella, esimerkiksi Mozilla Firefox selaimella eivät JavaScriptien tietyt toiminnot toimineet ollenkaan. Ulkonäöllisesti suurimmat erot näkyvät lähinnä lomakkeiden komponenttien ulkonäössä.

#### 3.1 Järjestelmän perusominaisuudet

Tässä luvussa selvitetään, kuinka järjestelmän sivurunko ja käyttäjäryhmät toimivat. Perusominaisuuksiin voidaan työssä lukea myös perustoiminnot, kuten sisään kirjautuminen, joita ei tarkemmin tässä raportissa käydä läpi.

##### 3.1.1 Sivurunko

Sivurunko on luotu HTML-kielellä. Jokaisella sivulla on siis samankaltainen sivupohja, johon sitten liitetään sivun tarvittavat PHP-kielellä tehdyt toiminnot. Lisäksi sivurunkoon ladataan valmiiksi tiedot ohjelmiston käyttämistä JavaScript-kirjastoista.

Kaikki sivut koostuvat yläpalkista, keskiosan ominaisuuksista ja toiminnoista sekä alapalkista. Yläpalkissa on sivuston logo, joka näkyy kaikille, sekä sisään kirjautuneille valikko, josta pääsee käyttäjän profiiliin ja kirjautumaan ulos sivustolta. Lisäksi ylläpitäjät pääset valikon kautta hallintapaneeliin. Keskiosan vasemmassa laidassa on valikko sivuston tärkeimpiin toimintoihin, kuten kalenteriin ja varausten tekoon. Valikon viereen jäävä tila on tarkoitettu sivun omille toiminnoille. Nämä toiminnot on ohjelmoitu omiin tiedostoihinsa, jotka liitetään sivuihin niille varattujen div-elementtien sisään. Alapalkista löytyvät tarvittavat yhteystiedot.

### 3.1.2 Käyttäjäryhmät ja oikeudet

Sivustolle on määritelty kolme käyttäjäryhmää: peruskäyttäjät, laitevastaavat ja ylläpitäjät. Näillä ryhmäjoilla jaetaan oikeudet sivuston käyttäjille ja pidetään huoli, etteivät ulkopuoliset pääse näkemään ylläpitäjille tarkoitettuja ominaisuuksia. Ulkopuoliset selaajat on myös otettu sivustossa huomioon.

Ulkopuolisille selaajille on avoimena vain laiterekisteri ja siitäkin heille näkyy vain perustiedot laitteista. Peruskäyttäjillä on käytössään suurin osa sivuston toiminnoista mutta tietokantaan he voivat lisätä tai muuttaa ainoastaan omia varauksiaan. Laitevastaavilla on lähes samat oikeudet kuin ylläpitäjillä. Laitevastaavat eivät pääse hallintapaneeliin, mutta heillä on oikeus muuttaa laitteen statusta sekä ylläpitää huoltokirjaa niistä. Heille tulee myös huoltopyynnöt laitteista, joihin he ovat laitevastaavina. Ylläpitäjillä on oikeudet kaikkiin sivuston ominaisuuksiin ja heidän tehtävänä on lisätä tai muokata laitteita, tiloja tai henkilöitä. Lisäksi ylläpitäjät pääsevät näkemään raportit laitteiden ja tilojen käytöstä.

Käyttäjäryhmien tunnistaminen tapahtuu sisään kirjautumisen yhteydessä tehdystä session-muuttujasta, johon tallennetaan järjestelmään kirjautuneen käyttäjän tietoja. Näillä tiedoilla haetaan tietokannasta tarvittut tiedot käyttäjän oikeuksista kuhunkin sivuston toimintoon.

## 3.2 Järjestelmän toiminnot

Tässä luvussa esitellään varausjärjestelmän tärkeimmät toiminnot.

### 3.2.1 Kalenteri

Kalenteri on yksi keskeisimpiä toimintoja varausjärjestelmässä ja se toimii myös kirjautumisen jälkeen ohjelmiston pääsivuna. Kalenterisivu koostuu kolmesta osasta, jotka ovat päivämäärän- ja tilanvalintapaneeli, minikalenteri ja lukujärjestys.

Kalenteri

Varaus Kalenteri

Tila: C1009

Päiväys: 27 10 2011

Päivitä

Viikko: 43 Tila: C 1009 (Vesilaboratorio)

Klo	Maanantai 24.10.2011	Tiistai 25.10.2011	Keskiviikko 26.10.2011	Torstai 27.10.2011	Perjantai 28.10.2011	Lauantai 29.10.2011	Sunnuntai 30.10.2011
07:00	Lisää varaus	Lisää varaus	Lisää varaus	Lisää varaus	Lisää varaus	Lisää varaus	Lisää varaus
07:30							
08:00							
08:30							
09:00							
09:30							
10:00							
10:30							
11:00							
11:30							
12:00							
12:30							
13:00							
13:30							
14:00							
14:30							
15:00							

KUVA 2. Kalenterisivu

Minikalenterin toteutukseen oli monia ideoita, kuten Internetistä löytyvät valmiit kalenterit ja jQuery:n tarjoama JavaScript. Paras ratkaisu oli kuitenkin ohjelmoida PHP:llä oma koodi, koska sitä olisi helppo tarvittaessa muuttaa ja ominaisuudet saisi toimimaan halutulla tavalla.

Kalenteri näyttää aina kuukauden kerrallaan, joten vain valitun kuukauden päivät ovat valittavissa. Kalenteria voidaan selata vaihtamalla kuukautta joko seuraavaan tai edelliseen kuukauteen. Vuosiluvut vaihdetaan samalla tavalla.

Kalenterin muodostaminen alkaa päivämäärämuuttujien alustamisella. Jos päivämäärää ei ole valittu, käytetään sen hetkistä päivämäärää. Kun alustukset ovat valmiina, aletaan muuttujilla ja PHP:n tarjoamilla funktioilla, esim. `date()`-funktio, tutkia kuukauden päivien lukumäärää. Lisäksi otetaan selvää että mikä viikonpäivä on kyseessä, kuinka monta päivää esitetään edelliseltä ja tulevalta kuukaudelta sekä sen kuukauden viikkojen numerot. Viikonpäivät ja viikkojen numerot alustetaan myös aputauluihin. Lopuksi kalenteri muodostetaan for-silmukassa HTML-tauluksi ja muokataan sitä JavaScriptillä, kuten KUVA 2. Kalenterisivu nähdään.

Valintapaneelilla on tarkoitus valita haluttu tila. Tilat haetaan tietokannasta pudotuslaatikkoon, josta valitsemalla ja päivitä-painiketta painamalla valitaan tila. Paneeliin



lisättiin myös päivämäärän asettamiselle kenttä, joka nopeuttaa tietyn päivämäärän hakemista ilman minikalenterin selaamista.

Lukujärjestykseen haetaan valitun päivämäärän ja tilan perusteella varaukset, mikäli niitä on tehtynä. Minikalenterin tavoin ensiksi alustetaan tarvittavat päivämäärämuutujat, minkä jälkeen tutkitaan, mistä viikosta on kyse. Tämän jälkeen tietokannasta haetaan halutulle viikolle tehdyt varaukset kyseiseen tilaan ja alustetaan ne aputauluihin. Aputaulujen avulla tutkitaan, että tulostetaanko tietylle ajalle mitään. Lopuksi lukujärjestys muodostetaan for-silmukassa HTML-tauluksi.

### 3.2.2 Laiterekisteri

Laiterekisteri muodostuu tietokannan sijainti- ja laitetaulukoista. Sivustolla on mahdollista hakea laitteita joko sijainnin tai nimen mukaan. On myös mahdollista listata kaikki laitteet sekä käytöstä poistetut laitteet.

The screenshot shows a web interface for searching equipment. On the left, there is a navigation menu with links: Kalenteri, Laiterekisteri, Tee Varaus, Omat Varaukset, and Yhteyshiedot. The main content area has a search form titled 'Anna hakuohdot:'. The form contains two input fields: 'Laite:' with the value 'mittari' and 'Tila:' with a dropdown menu showing 'C1009'. Below the fields are three buttons: 'Hae', 'Hae Kaikki', and 'Poistetut Laitteet'. Underneath the search form, the results are listed under the heading 'Haun tulokset:'. The results are organized into two columns. The left column contains: 'EC-käsimittari / Kitti' (Malli: HQ14d), 'pH-käsimittari / Kitti (nro2)' (Malli: pH-kitti HQ11d), and 'Redox-mittari' (Malli: Intellical™ ORP REDOX/MTC101). The right column contains: 'pH-käsimittari / Kitti (nro1)' (Malli: pH-kitti HQ11d), 'pH/EC-yhdistelmäkäsimittari' (Malli: ), and 'Sameusmittari' (Malli: 2100N IS Turbimeter).

KUVA 3. Laiterekisterisivun laitehaku. Tapauksessa annetut hakusanat ja tulostus

Laitehaku tehdään hakupaneelilla, joka on toteutettu HTML-formilla. Käyttäjän valitessa halutut hakumäärittelyt lähetetään nämä määrittelyt get-metodia käyttäen. Tällä tavalla hakusanat tulevat selaimen osoitepalkkiin, josta ne on helppo ottaa tutkittavaksi. Näillä hakusanoilla muodostetaan sitten MySQL hakulauseke, jolla saadaan haluttu hakutulos. Hakulauseke ottaa myös jokerimerkit huomioon, joten hakusanana ei tarvitse olla laitteen oikea nimi, kuten 17 tapauksessa näkyy. Hakutulok-

set tulostetaan HTML-tauluksi hakupaneelin alle. Laitteet ovat linkkejä, joilla pääsee jokaisen laitteen omalle laitekorttisivulle.

### 3.2.3 Laitekortti

Laitekorttisivu generoidaan laitehaun mukaan. Jokaisella laitteella on oma id numeronsa tietokannassa ja tämä numero lähetetään laitehaun yhteydessä laitekorttisivulle. Laitteen tiedot haetaan tämän id-numeron perusteella, ja ne tulostetaan omiin kenttiinsä, kuten 18 on tehty.

LAITE	EC-käsimittari / Kitti	<div style="border: 1px solid black; padding: 20px; width: fit-content; margin: auto;"> <p>No Photo Available</p> </div>
MALLI	HQ14d	
VALMISTAJA	HACH	
OMISTAJA	AMK	
LAITEVASTAAVA		
SIJAINTI	Vesilaboratorio / C1009	
STATUS	<input type="button" value="käytössä"/> <input type="button" value="Muuta"/>	
HANKINTA		
PÄIVÄMÄÄRÄ		
HANKKIJÄ		
POISTO		
PÄIVÄMÄÄRÄ		
POISTAJA		

[Etusivulle](#)

[Laiterekisteriin](#)
[Siirry varaussivulle](#)
[Lähetä huoltopyyntö](#)
[Kirjaa huoltotieto](#)

HUOLLOT	HUOLTO/TOIMENPIDE
PVM	

KUVA 4. Laitekortti laitteesta EC-käsimittari / Kitti

Laitteilla tietoja ovat nimi, malli, valmistaja ja omistaja. Näiden tietojen lisäksi laitteella on status, jolla määritetään onko laite normaalikäytössä tai jostain syystä poissa käytöstä. Jokaiselle laitteelle on määritetty laitevastaava, jolla on oikeus muuttaa laitteen statusta tarvittaessa. Myös sivuston ylläpitäjillä on tähän oikeus. Laitevastaavan tehtäviin kuuluu myös laitteen huoltotoiminnot, jotka ovat huoltopyyntöjen vastaanotto ja huoltomerkintöjen teko.

### 3.2.4 Huoltotoiminnot

Mikäli laitteessa tuntuu olevan vikaa, voi käyttäjä lähettää siitä huoltopyynnön. Huoltopyyntökaavake generoituu valmiiksi laitekortin tee huoltopyyntö -linkin kautta, ainoastaan pakollinen vapaamuotoinen vikailmoitus on käyttäjän kirjoitettava viestiin mukaan. Pyynnot lähetetään sähköpostilla, johon PHP:llä on oma mail()-funktio. Huoltopyynnön lähettäminen tapahtuu seuraavalla koodilla:

```

if($m_ilmoitus != "")
{
    mail($m_vast_posti,$subject,$m_viesti,$headers);

    $result_adminit = mysql_query("SELECT * FROM henkilo WHERE
    tyyppi=1");
    while($adminit = mysql_fetch_array($result_adminit))
    {
        $tmp_posti = $adminit["s_posti"];
        if($tmp_posti != $m_vast_posti)
        {

            mail($tmp_posti,$subject,$m_viesti,$headers);
        }
    }
    $result = mysql_query("SELECT * FROM laite WHERE
    laite='$m_laite'");
    $m_id = mysql_fetch_array($result);
    $id = $m_id[0];
    header("Location:laite.php?id=$id&lahetetty=true");
}
else
{
    $result = mysql_query("SELECT * FROM laite WHERE
    laite='$m_laite'");
    $m_id = mysql_fetch_array($result);
    $id = $m_id[0];
    header("Location:laite.php?id=$id&lahetetty=false");
}

```

Huoltopyynnön lähettäminen ei muuta laitteen statusta, koska vika ei välttämättä rajoita laitteen käyttöä. Mikäli laite osoittautuu epäkuntoiseksi, on laitevastaavan vastuulla lähettää laite huoltoon.

Laitteen palattua huollosta on siitä tarkoitus tehdä huoltomerkintä laitekortille. Huoltomerkinnälle generoidaan huoltopyynnön tavoin merkintäkaavake, johon kirjoitetaan ainoastaan selostus huoltotoimista. Nämä merkinnät tallennetaan tietokannan *huolto*-tauluun seuraavalla koodilla:

```

$m_henk_id = $_POST["henk_id"];
$m_pvm = $_POST["pvm"];
$m_laite_id = $_POST["laite_id"];

$stoimenpide = validate($_POST["tieto"]);

```

```

if($toimenpide == "")
{
    header("Location:laite.php?id=$m_laite_id&virhe=true");
}
else
{
    mysql_query("INSERT INTO huolto (id, laite_id, PVM,
toimenpide, henkilo_id) VALUES (NULL, '$m_laite_id', '$m_pvm',
'$toimenpide', '$m_henk_id')");
    closeDB($conn);
    header("Location:laite.php?id=$m_laite_id");
}

```

### 3.2.5 Varauksen teko

Varauksen tekeminen on siis sivuston keskeisin kokonaisuus laiterekisterin lisäksi. Varaukselle haluttiin paljon mahdollisuuksia, joten varauskaavakkeen täytyi olla helpokäyttöinen ja ennen kaikkea dynaaminen.

Varauskaavakkeen täyttö alkaa käyttäjätietojen syöttämisellä 20 näkyviin kenttiin. Sivusto hakee automaattisesti sisäänkirjautumisen aikana luodusta sessiosta käyttäjän perustiedot, joten määriteltäväksi jää lähinnä käyttäjäryhmän valinta, projektikoodin syöttö sekä lisätiedot varauksesta. Käyttäjäryhmällä eritellään AMK:n ja UEF:n varaukset toisistaan. Projektikoodilla tarkoitetaan projektille annettua erillistä koodia, mikäli sellainen on käytössä, tai sitten kenttä jätetään tyhjäksi. Lisätietueella annetaan käyttäjälle lähinnä mahdollisuus kommentoida tai antaa lisäselvitystä omasta varauksestaan.

Käyttäjätiedot

Käyttäjätunnus:

Nimi:

Sähköposti:

Puhelinnumero:

Käyttäjäryhmä: AMK

Projektikoodi  (jätä tyhjäksi, mikäli sellaista ei ole)

Lisätieto  (esim. varauksen nimi, käyttäjät)

KUVA 5. Käyttäjätietojen hallinta varauslomakkeessa

Käyttäjätietojen jälkeen tulee 21 mukainen aika-asetusosio. Päivämäärien valinta tehdään jQueryn kalenteriominaisuudella, joka aukeaa aina päivämääräkenttä valittaessa. Kalenteria pystyy selaamaan nykyhetkestä eteenpäin, mutta menneitä päivämääriä sillä ei pääse valitsemaan. Kellonaikoja valittaessa sivusto tutkii automaattisesti, että kellonaika on järkevä ja vähintään puolituntia pitkä. Aika-asetukset valitaan kolmen erilaisen paketin avulla, joissa sitten valitaan tarkemmat päivämäärät ja kellonajat.

Kertavaraus on tarkoitettu yhden tietyn päivän valitsemiseksi. Päivämäärän valinnan jälkeen valitaan alkamis- ja loppumisaika alavetovalikosta. Pitkäkestoisessa varauksessa määritetään yhden päivän sijaan aikaväli, joka asetetaan alkamis- ja loppumispäivämääräkenttiin. Tämän lisäksi on mahdollista valita vain tietty viikonpäivä annetulta aikaväliltä tai sitten varaus tehdään jokaiselle viikonpäivälle. Kellonajat määritetään samalla tavalla kuin kertavarauksessa ja kellonaika on sama jokaisena päivänä. Kolmantena vaihtoehtona on vapaamuotoinen varaus, jossa käyttäjä tekee niin monta kertavaraustyylistä valintaa kuin haluaa.

Aika-asetukset

Kertavaraus	Pitkäkestoinen varaus	Vapaamuotoinen varaus
<b>Aikaväli</b> <span style="float: right;">?</span>		
Alk.PVM		Lop.PVM
<input type="text" value="22.11.2011"/>		<input type="text" value="25.11.2011"/>
<b>Päivät</b>		
Ma Ti Ke To Pe La Su		
<input type="checkbox"/> <input type="checkbox"/> <input checked="" type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>		
<b>Kellonajat</b>		
Alk.Aika		Lop.Aika
<input type="text" value="7:00"/>		<input type="text" value="13:00"/>
<input type="button" value="päivitä"/>		

KUVA 6. Varauslomakkeen aika-asetukset. Tapauksessa on valittu pitkäkestoisien varauksen paketti.

Kun käyttäjä on asettanut aika-asetukset halutuiksi, hänen on painettava Päivitä-painiketta, jolloin sivusto päivittää tila- ja laitelistat aika-asetusten mukaan. Tämän ominaisuuden on tarkoitus vähentää päällekkäisvarauksen mahdollisuutta ja lisäksi käyttäjä näkee asettamiensa päivämäärien mukaan olevan varaustilanteen menemättä edestakaisin kalenterin ja varauslomakkeen välillä. Tila- ja laitelistat ovat suoraan laiterekisterin mukaiset.

Varauksen viimeisessä vaiheessa valitaan varausasetukset, jotka toimivat aika-asetusten tyyliisesti kolmen paketin osiona 23 mukaisesti. Vaihtoehtoina ovat tilanvaraus, laitevaraus ja kurssivaraus. Näistä kurssivarauksella on mahdollista varata toisten varaustyyppien päälle, koska kurssien päivämäärien ja kellonaikojen on oltava yhdenmukaiset AMK:n ja UEF:n kurssilukujärjestysten kanssa.

Tilanvarauksessa on ensiksi listattu kaikki valittavat tilat, joita valitsemalla päivittyy laitelista tilojen alle. Tiloja on mahdollista valita monta yhtä aikaa ja lisäksi joissakin tiloissa on vielä erikseen valittavissa tietyt lohkot. Mikäli yhtään lohkoa ei ole valittu niin oletetaan että käyttäjä haluaa koko tilan käyttöönsä. Laitevarauksessa käyttäjä voi varata vain tietyn laitteen käyttöön jättämällä laitteen tilan avoimeksi muille. Laitteita voi myös varata useita kerralla ja ne voivat olla eri tiloista valittuna. Kolmantena varausasetuksena on kurssivaraus, jossa jokaiselle kurssille on määritelty valmiiksi paketti tiloista ja laitteista.

Varausasetukset

Tilavaraus	Laitevaraus	Kurssivaraus	
Tila: <span style="float: right;">?</span>			
<input type="checkbox"/>	<a href="#">C2002 (Energiatekniikka)</a>		
<input checked="" type="checkbox"/>	<a href="#">C1004 (Jätevesilaboratorio)</a>		
	<input checked="" type="checkbox"/> <a href="#">1. Lohko</a>		
	<input type="checkbox"/> <a href="#">2. Lohko</a>		
<input type="checkbox"/>	<a href="#">C1008 (Koehalli)</a>		
<input type="checkbox"/>	<a href="#">C1002 (Mikroskopointi)</a>		
<input type="checkbox"/>	<a href="#">C2013 (Tuotekehityslaboratorio)</a>		
<input type="checkbox"/>	<a href="#">C2004 (Valvomo)</a>		
<input type="checkbox"/>	<a href="#">C1009 (Vesilaboratorio)</a>		
Laitteet			
<b>Tilan C1004 laitteet:</b>			
<input type="checkbox"/>	Annostelupumppu (nro1) Malli: DME 8-10 PP,V,F	<input type="checkbox"/>	Annostelupumppu (nro2) Malli: DME 8-10 PP,V,F
<input type="checkbox"/>	Annostelupumppu (nro3) Malli: DME 8-10 PP,V,F	<input type="checkbox"/>	Annostelupumppu (nro4) Malli: DME 8-10 PP,V,F
<input checked="" type="checkbox"/>	Annostelupumppu (nro5) Malli: DME 8-10 PP,V,F	<input type="checkbox"/>	Autoklaavi Malli: CV-EL 18L
<input checked="" type="checkbox"/>	BOD-laitteisto Malli: Oxitop	<input checked="" type="checkbox"/>	COD-Reaktori Malli: COD-reactor
<input type="checkbox"/>	Happianturi Malli: MODEL 58	<input type="checkbox"/>	Johtokyky mittari Malli: LF330
<input type="checkbox"/>	Käsimitari	<input type="checkbox"/>	Kasvatuskaappi

KUVA 7. Varauslomakkeen varausasetukset. Tapauksessa on valittu tilanvaraus paketti.

Kun kaikki tarvittavat valinnat on tehty, lähetä-painikkeella lähetetään varauslomakkeelta tiedot post-metodilla tee\_varaus.php:lle. Vaikka varauslomake tutkii käyttäjän antamat virheelliset arvot hyvinkin tarkasti, tutkitaan päällekkäisvarauksen mahdollisuudet vielä uudestaan ja muutetaan varauslomakkeen lähettämät tiedot oikeanlaiseksi tietokantaan kirjoittamista varten. Mikäli kaikki on kunnossa, kirjoitetaan varaus tietokantaan ja annetaan käyttäjälle tieto varauksen onnistumisesta. Jos varauksessa kuitenkin on jotain virheellistä, mitään ei kirjoiteta tietokantaan ja käyttäjälle annetaan virheilmoitus varauksen epäonnistumisesta. Kurssivarausta tehdessä tutkitaan myös,

tehdäänkö varaus aiemmin tehdyn varauksen yli. Jos kurssivaraus varataan aikaisemmin tehdyn varauksen yli, lähetetään tästä tieto aikaisemman varauksen tehneelle henkilölle ja aikaisempi varaus poistetaan tietokannasta.

### 3.2.6 Varausten hallinta

Tehtyjä varauksia voi vielä tarvittaessa jälkikäteen muokata omat varaukset sivulla. Varaukset listautuvat yhden kuukauden ajalta. Kun käyttäjä valitsee varauksen, niin tulostetaan listattujen varausten viereen tarkemmat tiedot valitusta varauksesta 24 mukaisesti. Myös menneet varaukset näkyvät mutta niitä ei ole mahdollista enää muokata, koska muutokset sekoittaisivat raporttien muodostamista.

Tulevissa varauksissa on mahdollista muokata laitteiden määrää tai poistaa varaus kokonaan. Laitetta lisättäessä sivusto listaa vapaina olevat laitteet varauksen päivämäärän mukaisesti. Laitteiden poistaminen käy samoin: käyttäjä valitsee listalta tarpeettomat laitteet ja poistaa ne varauksesta.

The screenshot shows a web interface for managing reservations. At the top, there are dropdown menus for the month (Marraskuu) and year (2011), and a 'Päivitä' button. Below this is a table of reservations with columns for 'PVM' (date) and 'Aikaväli' (time interval). The selected reservation is C1009 (Vesilaboratorio) on 24.11.2011 from 7:30 to 11:30. To the right, a detailed view of this reservation is shown, including the room (C1009), type (tilavaraus), and a list of equipment with options to 'Lisää laite' or 'Poista laite'. A 'Poista Varaus' button is also visible.

PVM	Aikaväli
<a href="#">C1008 (Koehalli)</a>	23.11.2011 7:00 - 10:00
<a href="#">C2004 (Valvomo)</a>	23.11.2011 7:00 - 10:00
<a href="#">C1004 (Jätevesilaboratorio)</a>	24.11.2011 7:30 - 11:30
<a href="#">C1009 (Vesilaboratorio)</a>	24.11.2011 7:30 - 11:30

<b>Tila</b>	C1009 (Vesilaboratorio) : 1. Lohko,2. Lohko
<b>Tyyppi</b>	tilavaraus
<b>Laitteet</b>	<a href="#">Lisää laite</a> <a href="#">Poista laite</a>
	Digestiolaitteisto (nro1) -
	Digitaalipyretti (50 ml (nro1)) -
	EC-käsimittari / Kittit - HQ14d
	Fermentori (nro2) - Bioflo 3000
<b>Poista Varaus</b>	
<b>Ryhmäleima</b>	AMK

KUVA 8. Varausten hallinta. Tapauksessa valittu marraskuu 2011 ja näkyvissä tilan C1004 varaus.



### 3.3 Järjestelmän ylläpitotoiminnot

Tässä luvussa on selostettu, kuinka ylläpitäjät hallinnoivat järjestelmää sekä kuinka laitteiden käyttöraportit muodostuvat.

#### 3.3.1 Järjestelmän tietojenhallinta

Järjestelmän tietojenhallinta on tarkoitettu ainoastaan sivuston ylläpitäjille. Uuden tiedon syöttäminen tai vanhojen muuttaminen tapahtuu hallintapaneelin kautta. Koska järjestelmän on tarkoitus kerätä dataa erilaisiin toimintoihin, hallintapaneelin kautta ei voi poistaa tietokannasta mitään. Hallintapaneelissa on neljä eri hallintamuotoa: käyttäjähallinta, laitehallinta, tilahallinta sekä kurssipakettihallinta. Myös käyttöasteraportit löytyvät hallintapaneelin kautta.

Käyttäjähallinnassa ylläpitäjät voivat lisätä uusia käyttäjiä, muokata jo tietokantaan tallennettujen käyttäjien tietoja sekä tarvittaessa poistaa käytöstä henkilöitä, joiden ei haluta enää voivan käyttää järjestelmää. Kun uusi käyttäjä lisätään järjestelmään, luodaan hänelle salasana automaattisesti, joka lähetetään ylläpitäjän asettamaan sähköposti osoitteeseen. Järjestelmään ei ole mahdollista rekisteröityä itsenäisesti, vaan rekisteröityminen tapahtuu ylläpitäjien kautta, jonka takia sivustolta löytyy ylläpitäjien yhteystiedot.

Laitehallinnassa on käyttäjähallinnan tavoin mahdollista lisätä uusi laite, muokata vanhojen laitteiden tietoja sekä poistaa laitteita käytöstä. Tätä kautta laitteille asetetaan laitevastaava, joka valitaan laitevastaavien listasta. Lisäksi laitehallinnan kautta voidaan asettaa laitteen sijainti, liittää laitteeseen käyttöoppaita ja ladata laitteen kuva sivustolle.

Tilojenhallintaan lopulta jäi vain toiminto, jolla tilojen lohkojakojen kuvat saadaan ladata sivustolle. Toiminnot tilojen lisäämiseen ja muokkaamiseen olisi laitehallinnan ja käyttäjähallinnan tavoin ollut mahdollista toteuttaa, mutta tilaajat pitivät tätä toimintoa tarpeettomana.

Kurssipakettihallinnassa on mahdollista lisätä uusia kurssipaketteja sekä muokata tai poistaa vanhoja. Uuden kurssipaketin lisääminen tapahtuu tallentamalle ensin uuden kurssipaketin nimi tietokantaan, jonka jälkeen uuteen pakettiin voidaan lisätä tiloja ja laitteita. Tietyn laitteen lisäämiselle on ehtona, että laitteen tila on myös valittuna.

Tiloja ja laitteita on myös mahdollista poistaa kurssipaketeista. Kurssipaketit ovat ainoat tiedot, jotka voidaan kokonaan poistaa tietokannasta, koska niillä ei ole merkitystä varauksien tai raporttien muodostamisessa.

### 3.3.2 Raportit

Raportit on tarkoitettu tilojen ja laitteiden käyttöasteen mittaamiseen. Koska varauksen tekijät ovat sekä AMK:n että UEF:n puolelta, pitää raportteja pystyä tutkimaan joko molempien ryhmien osalta tai sitten erikseen. Käyttöaste saadaan tehdyistä varauksista tietyltä aikaväliltä.

Raportteja voidaan tutkia kuukausi kerrallaan tai koko vuoden ajalta. Ensiksi valitaan tila, jonka käyttöaste tahdotaan tietää. Valinnan jälkeen tilan laitteet listataan 26 tavoin ja laitteen perässä näkyy kokonaiskäyttömäärä tunteina. Mikäli yhdestä laitteesta halutaan tarkempia tietoja, niin valitaan laite listasta, jolloin laitteesta aukeaa oma taulukko. Taulukkoon on listattu ensiksi kaikki tunnit yhteensä ja sen jälkeen sekä AMK:n että UEF:n ryhmät erikseen. Mikäli laitteella on projektikoodillisia varauksia, on ne listattu omaan alueeseen taulukossa.

KK/VUOSI	
koko vuosi	2011
<input type="button" value="Päivitä"/>	<input type="button" value="Paluu"/>
<a href="#">Tuotekehityslaboratorio</a> <a href="#">Energiatekniikka</a> <a href="#">Valvomo</a> <a href="#">Jätevesilaboratorio</a> <a href="#">Koehalli</a> <a href="#">Vesilaboratorio</a> <a href="#">Mikroskopointi</a>	
Laitteiden tuntimäärät (AMK/UEF)	
Koehalli	
2011	
<a href="#">Flokkulaattori (6 dek.)</a> - 0 h	<a href="#">Putkistovirtauksen tutkimuslaitos</a> - 0 h
<a href="#">Virtaaman kulku maaperässä -tutkimuslaite (pintavesi, pohjavesi, sadetus, kallistuskulmat)</a> - 20.5 h	<a href="#">Vesihaude (kylmä/kuuma)</a> - 22 h
<a href="#">Lauda</a> - 27 h	<a href="#">Homogenesaattori</a> - 27 h
<a href="#">Automaattinen näytteenotin (16 näytepulloa)</a> - 21 h	<a href="#">Virtaushäviön/nestekitkan tutkimuslaite</a> - 15 h
<a href="#">Laskeutusallas</a> - 3 h	<a href="#">Virtausallas</a> - 16 h
<a href="#">Suodatinkolonnit</a> - 22 h	<a href="#">Täryytin</a> - 11.5 h

KUVA 9. Käyttöasteraportti. Tapauksessa on valittu koko vuosi 2011 sekä molemmat ryhmät. Tapauksessa tilaksi on valittu koehalli, jonka laitteet on listattu raportiksi

### 3.4 Tietojen tallentaminen

Järjestelmässä tietojen tallentaminen aloitetaan syöttämällä ensin HTML-kaavakkeelle tallennettavat tiedot, jonka jotka lähetetään joko post- tai get-metodilla tarkistettavaksi. Jokainen tietorivi tarkistetaan seuraavalla tavalla.

```
function validate($m_Data)
{
    $m_Data = stripslashes($m_Data);
    $m_Data = mysql_real_escape_string($m_Data);

    return $m_Data;
}
```

Tarkistuksessa stripslashes-funktio poistaa tallennettavasta tekstistä kenoviivat, koska niitä ei oleteta kuuluvan tallennettavaan tekstiin. Kenoviivoilla taitavat hakkerit voisivat onnistua sekoittamaan tietokantaa lähettämällä virheellisiä tallennustietoja.

Kenoviivojen poiston jälkeen tallennettava teksti on käyty läpi mysql\_real\_escape\_string-funktiolla. Funktiolla on tarkoitus estää niin sanotun SQL-injection. SQL-injectiolla tarkoitetaan, että syötteen antaja yrittäisi muokata annetulla tekstillä SQL-lauseketta toisenlaiseksi kuin lausekkeen olisi tarkoitus olla.

Nämä toiminnot ovat osa järjestelmän tietoturvaa, jolla pyritään estämään mahdolliset haittatoimet järjestelmän käyttämään tietokantaan. Näiden lisäksi virheellisen tiedon tallentamista pyritään estämään kappaleen 3.1.2 mukaisilla käyttäjäryhmillä.

### 3.5 Hylätyt ominaisuudet

Tässä luvussa esitellään muutamia ominaisuuksia, jotka on jouduttu varausjärjestelmästä jättämään pois.

Varausjärjestelmästä jäi toteuttamatta muutama työnantajan toivoma ominaisuus. Nämä ominaisuudet eivät olleet keskeisiä järjestelmän toimivuuden kannalta, mutta olivat kuitenkin haluttuja lisäominaisuuksia. Keskeisin syy hylkäämiselle oli aika. Kaikki toiminnot olisivat olleet toteutettavissa, mutta toteutus olisi vaatinut enemmän opettelua ja tutkimista, mihin ei aikataulussa ollut tilaa.

### 3.5.1 Automaattinen uloskirjautuminen

Automaattisen uloskirjautumisen toivottiin toimivan siten, että käyttäjän siirryttyä selaimella varausjärjestelmän ulkopuolisille sivuille olisi hänet samalla kirjattu ulos automaattisesti. Toiminto olisi tietoturvasyistä hyödyllinen mutta sen toteuttaminen sellaisenaan olisi haastavaa, joten siitä luovuttiin jo työn alkuvaiheessa.

Ominaisuuden teki haastavaksi erityisesti se, että sellaista ei ole käytössä ainakaan tutuissa sivustoissa, kuten Savonian käyttämässä Moodlessa sekä Outlook-sähköpostiohjelmassa. Savonian käyttämien ohjelmien lisäksi tutkin myös Savonian ulkopuolisten web-sivustojen käyttäytymistä, eikä näilläkään heti automaattisesti kirjauduta ulos järjestelmästä, vaan uloskirjautuminen tapahtuu tietyn ajan kuluessa.

Ominaisuus ei jäänyt lopulta kokonaan toteuttamatta, vaan kirjautumisen yhteydessä muodostettu session-muuttuja ohjelmoitiin tuhoamaan itsensä tietyn ajan kuluttua. Tämä tapahtuu niin, että session-muuttujaan otetaan muistiin kirjautumisen hetkellä oleva aika, jota verrataan sivujen latauksien aikana seuraavan koodin mukaisesti.

```
$m_kesto = 1800;
if(isset($_SESSION['kesto']) )
{
    $session_life = time() - $_SESSION['kesto'];
    if($session_life > $m_kesto)
    {
        session_destroy();
        header("Location: login.php?aika=true");
    }
}
```

`$m_kesto` on muuttuja, johon asetetaan ajan pituus. Tässä tapauksessa 1 800 vastaa sekuntien määrää eli 1 800 sekuntia on puoli tuntia. `$session_life`-muuttujaan alustetaan erotus `time()`-funktion antaman tuloksen sekä session-muuttujan alustusajan väliltä. Tämän jälkeen verrataan, onko `$session_life`-muuttuja suurempi kuin puoli tuntia (`$m_kesto`-muuttuja). Mikäli aikaa on kulunut enemmän kuin puoli tuntia, tuhoetaan session-muuttuja ja ohjataan käyttäjä kirjautumissivulle. Varausjärjestelmä siis kirjaa käyttäjän automaattisesti ulos puolen tunnin kuluessa sisään kirjautumisesta.

### 3.5.2 Liittäminen Savonian lukujärjestyksiin

Savonian lukujärjestyksien liittämisen avulla olisi kurssivaraukset tallennettu järjestelmään automaattisesti. Toimintoa ei kuitenkaan alettu toteuttamaan, koska tutustuminen Savonian lukujärjestyksien käyttämiin ohjelmiin ja tietokantoihin olisi vienyt liikaa aikaa. Lukujärjestyksien liittäminen olisi tuonut nopeasti ja helposti kurssivaraukset järjestelmään, jolloin myös päällekkäisvarauksien mahdollisuus olisi pienentynyt. Nyt kurssivarausten tekeminen jää ensisijaisesti kurssien opettajille.

### 3.5.3 Raporttien valmiit tulostusversiot

Raporttien valmiit tulostusversiot olisivat olleet esimerkiksi Word-dokumentteja, jotka olisi saatu suoraan järjestelmästä. Ominaisuus jäi toteuttamatta, koska sille ei jäänyt aikaa. Järjestelmässä raportit näkyvät omina web-sivuinaan, joihin lasketaan käyttöasteet valmiiksi ja muodostetaan tiedot 26 mukaisesti. Sivuilta tiedot voidaan kopioida tekstinkäsittelyohjelmaan, jolla raportit tulostetaan paperiversioiksi.

## 4 TIETOKANTA

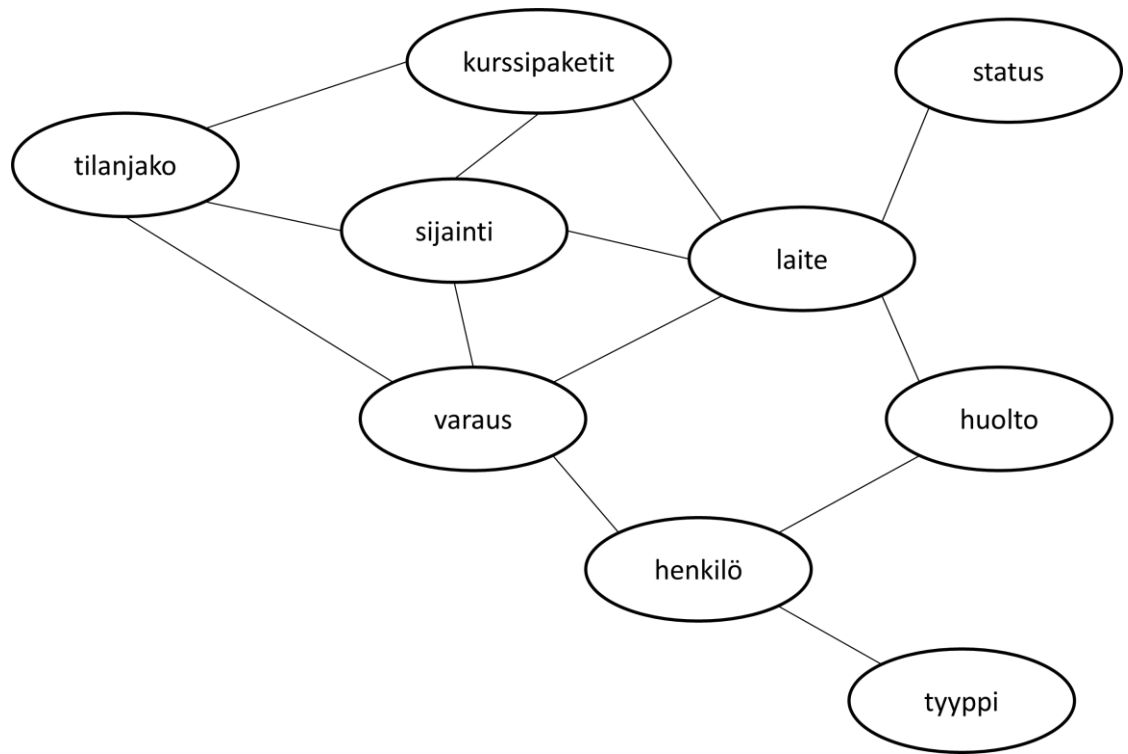
Tässä luvussa esitellään sähköisen varausjärjestelmän pohjana toimiva tietokanta. Tietokanta on toteutettu MySQL-tietokantaan, ja sitä on tarkoitus käyttää varausjärjestelmän datan tallentamiseen sekä sivuston dynaamisuuden avustamiseen. Tietokanta koostuu yhdeksästä taulusta, jotka ovat suurimmaksi osaksi liittyneinä toisiinsa 31 mukaisesti.

Henkilötaulu on tarkoitettu järjestelmän käyttäjien tietojen säilytykseen. Käyttäjien tietojen lisäksi tauluun määritellään myös päivämäärät tietojen lisäämisestä, koska sivuston kautta ei ole tarkoitus kokonaan pystyä poistamaan käyttäjätietoja tietokannasta. Tauluun linkittyy läheisesti tyyppitaulu, joka määrittää kirjautuvan henkilön käyttöoikeudet.

Tietokannan suurin taulu on laitetaulu, johon tallennetaan kaikki sivuston käyttämät laitteet. Perustietojen lisäksi tauluun linkitetään laitteen laitevastaava henkilötaulusta ja laitteen status määritetään statustaulun avulla. Lisäksi mikäli laitetta on tarkoitus huoltaa, tallennetaan huoltomerkinnät huoltotauluun, joka on linkitetty laitetauluun. Laitetaulu toimii sivuston laiterekisterin pohjana.

Laitetauluun linkittyy vahvasti myös sijaintitaulu, johon tallennetaan varattavissa olevat laboratoriotilat. Jokainen laite on merkitty tiettyyn sijaintiin. Tila voi olla jaettuna useaan lohkoon, jotka ovat erikseen varattavissa, joten sijaintitaulu on linkitetty tilanjakotauluun, johon on tarkemmin määritelty kyseiset lohkot.

Keskeisin taulu tietokannassa on varaustaulu. Kun käyttäjä tekee varauksen, kerätään tähän tauluun aikaisemmista tauluista linkitykset varattuihin tiloihin, lohkoihin ja laitteisiin sekä varauksen tekijä. Varaustaulusta saadaan tiedot sivuston varauskalenteriin ja ylläpitäjien raportteihin. Varauksiin on tehty myös valmiit kurssipaketit, jotka on tallennettu kurssipaketit-nimiseen tauluun.



KUVIO 10. Pelkistetty kuva tietokannasta

## 5 TARKASTELU

Työtä tehdessäni sain hyvää kokemusta web-ohjelmiston suunnittelusta ja toteuttamisesta. Työskentelyä helpotti se, että sain itse päättää työntekoon tarvittavat ohjelmat, sekä sain toteuttaa ohjelmoinnin omalla tavallani. Mikäli tilaaja olisi määritellyt minulle ennestään tuntemattomat ohjelmat, aikataulusta olisi tullut hyvin tiukka, koska näihin ohjelmiin olisi täytynyt tutustua.

Työn HTML- ja PHP-ohjelmointi ei ollut uutta, mutta työn edetessä huomasin, kuinka tärkeää oman koodin tunteminen ja kommentointi lopulta on. Monesti tuli vastaan tilanteita, jolloin ohjelmassa tapahtunutta virhettä sai etsiä hyvin paljon kauemmin kuin mitä siihen olisi kommenttien avulla mennyt. Myös suunnitteluvaihe antoi hyvää oppia siitä, kuinka paljon organisoidummin suunnittelu kannattaa toteuttaa, vaikka muita työntekijöitä ei projektissa olisikaan.

Uutena asiana työssä oli varauskaavakkeen kalenteri- ja valikkotoiminnoissa käytetty jQuery. jQuery:n opetteleminen ei kuitenkaan vienyt ylimääräistä aikaa työnteolta, koska jQuery on hyvin samanlaista ohjelmointikieltä kuin perinteinen JavaScript. Huomasin myöhemmin, että jQuery on alkanut yleistyä web-ohjelmointityöpaikkojen vaatimusluetteloissa, joten sen opetteleminen oli hyödyllistä.

Työn ohjelmointiosuuden valmistuttua tuli ongelmia siinä, että oikeassa palvelimessa jotkin ominaisuudet toimivat eritavalla kuin testipalvelimessa. En ollut ottanut palvelimien eroavaisuuksia huomioon ohjelmoinnin alkaessa, vaan oletin palvelimen olevan samanlainen kuin olin aikaisemmin käyttänyt ollessani harjoittelussa Savonian Tietohallinnolla. Tämän huolimattomuuden takia jouduin useat kerrat vielä muokkaamaan valmiita koodeja, jotta ne toimisivat kunnolla myös oikealla palvelimella. Järjestelmä valmistui lopulta toimintakuntoon, kun viimeisetkin koodikorjaukset oli saatu valmiiksi.

Jos aloittaisin työn tekemisen alusta, keskittyisin aluksi perusteellisesti työn suunnitteluun. Uskoisin ohjelmointiosuuden hoituneen paljon nopeammin, mikäli minulla olisi alusta asti ollut selkeästi tiedossa mitä tehdä. Ennen ohjelmoinnin aloittamista olisi tärkeää ottaa selvää palvelimen toiminnoista.

Jatkokehityksenä varausjärjestelmään voisi liittää luvussa 3.5 käydyt hylätyt toiminnot. Myös varausjärjestelmän liittäminen Savonian ja UEF:n tietojärjestelmiin olisi toimiva kehitys ohjelmistolle. Liittämisen avulla varausjärjestelmään tallennettujen



käyttäjätunnuksien ja salasanojen sijaan voitaisiin käyttää Savonian ja UEF:n tunnuksia.

Olen tyytyväinen opinnäytetyön lopputulokseen ja työn rajaukseen. Sain ohjelmoitua toimivan web-sivuston sekä sen käyttämän tietokannan. Tämän perusteella työn tavoite saavutettiin.

## LÄHTEET

<sup>1</sup> jQuery www-sivu [viitattu 19.1.2012]. Saatavissa:

<http://jquery.com/>

<sup>2</sup> MySQL [verkkodokumentti]. [viitattu 19.1.2012]. Saatavissa:

<http://fi.wikipedia.org/wiki/MySQL>

<sup>3</sup> PHP [verkkodokumentti]. [viitattu 19.1.2012]. Saatavissa:

<http://fi.wikipedia.org/wiki/PHP>

<sup>4</sup> W3Schools [verkkodokumentti]. [viitattu 19.1.2012]. Saatavissa:

<http://fi.wikipedia.org/wiki/HTML>