

Lauri Rantanen

SELAINPOHJAINEN VARASTONHALLINTAOHJELMISTO

Tietotekniikan koulutusohjelma
Ohjelmoinnin suuntautumisvaihtoehto
2012

SELAINPOHJAINEN VARASTONHALLINTAOHJELMISTO

Rantanen, Lauri
Satakunnan ammattikorkeakoulu
Tietotekniikan koulutusohjelma
Joulukuu 2012
Ohjaaja: Kivi, Karri
Sivumäärä: 24
Liitteitä: 1

Asiasanat: HTML, PHP, Javascript, MySQL

Tämän opinnäytetyön aiheena oli suunnitella ja toteuttaa selainpohjainen varastohallintaohjelmisto Jouko Sjöblom Oy:lle. Yritys on ottanut käyttöön uuden liimapuu-tehtaan, jonka ylläpitämisen helpottamiseksi yritys halusi kyseisen ohjelmiston.

Ohjelmisto ja erityisesti tietokanta suunniteltiin yritykseltä saatujen tietojen pohjalta. Tietokantaohjelmistona käytettiin MySQL-tietokanta, ohjelmiston ulkoinen rakenne toteutettiin HTML:llä ja toiminnallisuus PHP:llä. Ulkoasun muotoiluun käytettiin CSS-tyylimäärittelyä.

WEB-BASED INVENTORY MANAGEMEN SOFTWARE

Rantanen, Lauri

Satakunnan ammattikorkeakoulu, Satakunta University of Applied Sciences

Degree Programme in Information Technologies

December 2012

Supervisor: Kivi, Karri

Number of pages: 24

Appendices: 1

Keywords: HTML, PHP, Javascript, MySQL

The purpose of this thesis was to design and implement a web-based inventory management software to Jouko Sjöblom Oy. The company wanted this kind of a software in order to facilitate the maintenance of their new laminated timber factory.

The software and specifically the database were designed based on the information given by the company. MySQL was used as database software, the external structure was created using HTML and the functionality with PHP. CSS was used as style description.

SISÄLLYS

Sisällys

LYHENTEET JA TERMIT	5
1 JOHDANTO.....	6
2 HTML.....	7
3 CSS.....	7
4 JAVASCRIPT	8
5 PHP.....	9
5.1 Yleistä	9
5.2 Muuttujat.....	10
5.3 Luokat ja oliot.....	10
5.4 Istunnot	11
6 MYSQL.....	12
6.1 Yleistä	12
6.2 SQL-kieli.....	13
6.2.1 Taulun luonti	13
6.2.2 Tauluun kirjoittaminen	13
6.2.3 Muokkaaminen	14
6.2.4 Poistaminen.....	14
6.2.5 Hakeminen.....	15
7 MVC.....	15
8 VARASTOKIRJANPITO-OHJELMISTO	17
8.1 Tietokannan suunnittelu.....	17
8.2 Käyttöliittymä	19
8.2.1 FrontController	19
8.2.2 Sisään kirjautuminen	20
8.2.3 Materiaalit ja tuotteet.....	21
8.2.4 PDF.....	22
9 YHTEENVETO	23
LÄHTEET.....	24

LYHENTEET JA TERMIT

WWW	World Wide Web
HTTP	Hypertext Transfer Protocol
HTML	Hypertext Markup Language
PHP	Hypertext Preprocessor
CSS	Cascading Style Sheets
JS	JavaScript
SQL	Structured Query Language
MD-5	Message-digest-algoritmi
URL	Uniform Resource Locator
SESSION	Istunto PHP:ssä
MVC	model-view-controller (malli-näkymä-käsittelijä), ohjelmistokehitysmalli
GPL	GNU General Public License, vapaiden ohjelmistojen lisenssi

1 JOHDANTO

Jouko Sjöblom Oy on Jurvassa toimiva yritys, jonka päätoimena on metsä- ja maatalouskonekauppa. Yrityksellä on myös uusi liimapuutehdas, jonka varaston ylläpidon helpottamiseksi haluttiin kehittää varastohallintaohjelmisto.

Tämän opinnäytetyön tavoitteena oli suunnitella, toteuttaa ja ottaa käyttöön selainpohjainen varastohallintaohjelmisto edellä mainitulle yritykselle. Ohjelmiston kehityksessä käytettävät tekniikat ja toteutustapa olivat minun päätettävissä. Työssä tutustutaan MySQL-tietokannan suunnitteluun ja toteutukseen, PHP-ohjelmointiin sekä HTML-, CSS- ja JavaScript-kielten käyttöön. Päätin käyttää ohjelmointikielenä PHP:tä sen helppouden ja tehokkuuden vuoksi ja tietokantana MySQL:ää maksuttomuuden, helppokäyttöisyyden ja luotettavuuden vuoksi.

Varastohallintajärjestelmän tarkoituksena on helpottaa ja nopeuttaa yrityksen materiaaleihin ja tuotteisiin liittyvien prosessien, kuten valmistuksen, pakkaamisen ja lähetysten seuraamista ja dokumentointia. Järjestelmällä voidaan muun muassa tulostaa pakettiseteleitä ja rahtikirjoja, seurata materiaalien ja tuotteiden varastotilannetta sekä pitää yllä asiakasrekisteriä.

2 HTML

HTML on (lyhenne sanoista Hypertext Markup Language, suomennettuna hypertekstin merkintäkieli) kuvauskieli, jolla kuvataan web-sivustojen rakennetta. HTML:n ensimmäinen versio julkaistiin vuonna 1991. Nykyisin käytettävä versio HTML5 on W3C:n standardoima ja ylläpitämä.

HTML-dokumentti aloitetaan ja lopetetaan <HTML>-tagilla. <HTML>-tagin sisällä dokumentti jaetaan kahteen lohkoon: <HEAD> ja <BODY>. <HEAD> lohkon sisällä määritetään muun muassa dokumentin otsikko, tyylitiedostot ja scriptit. <BODY> lohkon sisään kirjoitetaan kaikki, mitä halutaan käyttäjälle näkyväksi.

HTML-dokumentti muodostuu elementeistä. Elementit voivat sisältää aloitustagin, lopetustagin, attribuutteja, attribuuttien arvoja sekä tekstiä. Esimerkkinä elementti sisältää aloitustagin , attribuutit scr, height ja width sekä kyseisten attribuuttien arvot. /11/

3 CSS

HTML-kieltä ei suunniteltu määrittämään dokumentin ulkoasua. Ulkoasun määrittämistä varten on kehitetty CSS (Cascading Style Sheets). CSS on tyylikieli, jolla määritetään kuinka HTML-elementit esitetään. CSS on W3C:n ylläpitämä ja standardoima ja sen ensimmäinen versio CSS1 julkaistiin vuoden 1996 lopussa. Nykyisin käytössä oleva versio CSS3 julkaistiin vuonna 2011.

CSS voidaan liittää sivuun ulkoisena tiedostona, kirjoittaa tyylimäärittelyt HTML-dokumentin <HEAD>-tagien sisään tai kirjoittaa tyylimäärittely suoraan HTML-tagin sisään. Viimeisimpänä mainittua tapaa ei ole suositeltu käytettäväksi, sillä se on CSS:n todellisten tarkoituksen vastaista. /11/

```

<html>
  <head>
    <link rel="stylesheet" type="text/css" href="tyyli.css">
    <style>
      h2{
        text-align:center;
        font-size:25;
      }
    </style>
  </head>
  <body>
    <h1>Pääotsikko</h1>

    <h2>Väliotsikko</h2>

    <p style="color:green; font-family:arial">Tekstiä</p>
  </body>
</html>

```

Kuva 1. Kolme eri tapaa liittää CSS-tyylimuotoilu HTML-dokumenttiin

4 JAVASCRIPT

”JavaScript on Netscapen kehittämä oliopohjainen ohjelmointikieli, jolla saadaan lisättyä dynaamisia toimintoja WWW-sivuille. JavaScript ei nimestään huolimatta ole Javaa, vaikka kielten syntakseilla onkin paljon yhteistä. JavaScript tulkataan suoraan selaimessa, joten siitä ei tarvitse kääntää erillistä suoritettavaa tiedostoa”. /5/

”Viimeisin kielen määrittely on JavaScript 1.8.5, joka pohjautuu EcmaScript-standardi ECMA-262 Edition 3. Standardoitua JavaScriptiä kutsutaan nimellä ECMAScript. JavaScriptiä kehittää standardin mukaiseksi Netscape yhdessä ECMA-standardointiorganisaation kanssa.

Useat selaimet tukevat standardin lisäksi lisätoiminnallisuuksia, kuten Mozilla-selainten E4X, joka on XML:n käsittelyyn erikoistunut kielen laajennus (standardi ECMA-357). JavaScriptin kehitti alun perin Netscapelle Brendan Eich nimellä *Mocha*. Nimi muutettiin *LiveScriptiksi* ja viimein markkinointisyistä *JavaScriptiksi*”. /6/

JavaScript voidaan kirjoittaa suoraan HTML-koodin sekaan tai lukea erillisestä tiedostosta. Kummassakin tapauksessa se sijoitetaan `<script type="text/javascript">` ja `</script>` -tagien väliin.

```
<script type="text/javascript">
  // Tulostetaan rivi tekstiä selaimen:
  document.write("Hello World!");
</script>
```

Kuva 2. JavaScriptin kirjoittaminen suoraan HTML-koodin sekaan

```
<script type="text/javascript" src="esimerkki.js">
</script>
```

Kuva 3. JavaScriptin lukeminen erillisestä tiedostosta ”esimerkki.js”

5 PHP

5.1 Yleistä

”PHP (PHP: Hypertext Preprocessor) on työväline dynaamisten Web-dokumenttien luomiseen. Laajasti ymmärrettynä PHP viittaa sekä varsinaiseen PHP-kielen että niihin teknisiin ratkaisuihin, joilla PHP-kielisten ohjelmien suorittaminen on mahdollista. Kokonaisuutena voidaan puhua PHP-ympäristöstä. PHP on HTML-dokumenttien sisään upotettavaa Web-palvelimella tulkittava ohjelmointikieli. Tulkittavia ohjelmointikieliä kutsutaan yleisesti scriptikieliksi. PHP:n syntaksi on lainattu suurimmaksi osaksi C-kielestä. Lisäksi mukana on piirteitä Java-, Perl- ja C++-kielistä muutamine täysin omine PHP-lisäyksineen”. /2 s.12/

PHP sai alkunsa vuonna 1994 Rasmus Lerdorf:n kehitettyä sen ensimmäisen version ”Personal Home Page Tools”. Seuraava versio julkaistiin vuonna 1995 nimellä PHP/FI (Personal Home Page / Forms Interpreter). Version PHP/FI 2.0 jälkeen alettiin käyttää nimeä PHP. Vuonna 2004 julkaistiin PHP 5, joka on tällä hetkellä käytössä oleva versio. /2, 3/

5.2 Muuttujat

PHP:ssä ei tarvitse erikseen määrittää muuttujia, vaan muuttujan tietotyyppi määräytyy automaattisesti sen perusteella, minkä tyyppistä dataa siihen sijoitetaan. Muuttujat erotetaan dollarimerkillä, esimerkiksi *\$muuttuja*. Muuttujan nimi voi sisältää isoja ja pieniä kirjaimia, numeroita sekä alaviivoja, mutta se ei voi alkaa numerolla. PHP:ssä iso ja pieni kirjain merkitsevät eri asiaa, joten *\$muuttuja* ei ole sama kun *\$MUUTTUA*. /2/

5.3 Luokat ja oliot

”PHP on myös oliopohjainen ohjelmistokieli eli sen avulla voidaan määritellä luokkia, niiden jäsenfunktioita ja ominaisuuksia. Luokkien periminen on myös mahdollista”. /1, s.57/

Luokan määrittely aloittaa sana *class*, jonka jälkeen tulee luokan nimi. Luokka muodostuu muuttujista ja funktioista, jotka määritellään luokan sisällä. Funktioille ja muuttujille voidaan määrittää näkyvyysalueeksi *private*, *public* tai *protected*. Jos näkyvyysalue on *private*, muuttujaa tai funktiota voi käyttää vain luokan sisäpuolelta. Jos näkyvyysalue on *public*, käyttämien onnistuu luokan ulkopuolelta. *Protected* näkyvyys mahdollistaa muuttujan tai funktion käyttämisen periytyvissä luokissa. Muuttuja tai funktio saa automaattisesti näkyvyysalueeksi *public*, jos sitä ei ole määritetty.

Funktio `__construct` on luokan konstruktori, jota kutsutaan aina, kun luokasta muodostetaan olio. Komento *new* luo luokkaa vastaavan olion. Luokan nimen jälkeen annetaan parametrit luokan konstruktoria mukaisesti. /8/

```

<?php
class Osasto {

    private $ID;
    private $nimi;
    private $paikkakunta;

    function __construct($ID, $nimi, $paikkakunta){
        $this->ID = $ID;
        $this->nimi = $nimi;
        $this->paikkakunta = $paikkakunta;
    }

    function tulosta(){
        echo "$this->ID $this->nimi $this->paikkakunta";
    }
}

$osasto = new Osasto(10, "Laskutus", "Pori");
$osasto->tulosta();

?>

```

Kuva 4. Esimerkki PHP luokasta.

Kuvassa 4 on luotu luokka `Osasto`, jolla on kolme muuttujaa sekä funktiot `__construct` ja `tulosta`. Kun luokasta luodaan olio `new`-komennolla, konstruktori ottaa vastaan annetut parametrit ja tallettaa ne muuttujiin. Tämän jälkeen kutsutaan funktiota `tulosta`, joka tulostaa muuttujiin tallennetut tiedot.

5.4 Istunnot

Istuntojen toiminta perustuu siihen, että palvelun käyttäjälle muodostetaan yksilöllinen id-tunnus, joka joko tallennetaan evästeeseen tai kulkee URL:n mukana. Palvelimelle luodaan PHP:n asetuksissa määriteltyyn hakemistoon id-tunnusta vastaava tiedosto, johon istunnossa käytetyt muuttujat tallentuvat. /9/

Istunnot käynnistetään kutsumalla funktiota `session_start()`. Kyseinen funktio kutsutaan heti PHP -sivun alussa ennen HTML:ää. Kun istunto on käynnistetty, voidaan istuntomuuttujiin tallettaa tietoa. Istuntomuuttuja luodaan istuntomuuttujatauluun `$_SESSION` antamalla muuttujalle nimi kuvan 5 esittämällä tavalla. Yksittäisen is-

tuntomuuttujan poistaminen tapahtuu *unset()*-komennolla. Koko istunnon voi poistaa käyttämällä komentoa *session_destroy()*.

```
<?php
session_start();
$_SESSION['nimi'] = 'Matti';
$_SESSION['osasto'] = 'Laskutus';

unset($_SESSION['nimi']);
session_destroy();
```

Kuva 5. Esimerkki istunnon käyttämisestä

6 MYSQL

6.1 Yleistä

”MySQL on monipuolinen, joustava ja suorituskykyinen relaatiotietokanta, jota käytetään niin suurten kuin pientenkin WWW-palvelujen taustalla. Se on ruotsalaisen MySQL Ab:n käsialaa ja oli alun perin tarkoitettu kyseisen konsultointiyrityksen sisäiseen käyttöön. MySQL noudattaa asiakas-palvelin-arkkitehtuuria, jossa sovellukset eivät koskaan käsittele tietokantaa suoraan, vaan käsittely tapahtuu aina palvelinohjaimen kautta. MySQL on helppo asentaa ja ylläpitää. Se ei vaadi täyspäiväistä huolenpitoa siihen tapaan kuin kalliimmat kaupalliset tietokantaohjelmistot. MySQL soveltuu näin pienempien WWW-palveluiden taustatietokannaksi, sillä sen ylläpitokustannukset ovat todella pienet”. /1, s.34/

MySQL on saatavana sekä ilmaisena että maksullisena versiona. Ilmainen versio on GPL –lisensoitu, eli jokaisella on oikeus käyttää, muokata ja jakaa sitä. Maksullisissa versioissa on enemmän ominaisuuksia sekä niille on laajempi tuki. /7/

6.2 SQL-kieli

”SQL (Structured Query Language) on pitkälle standardoitu relaatiotietokantojen kysely ja määrittelykieli. Sen rakenteet muistuttavat etäisesti englanninkieltä; komentosanoina ovat muun muassa select, update, delete, where ja order by. SQL on täysin merkkipohjainen. SQL-kielen komennot jaotellaankin määrittely- ja käsittelykomentoihin sen perusteella, luodaanko uusia tietorakenteita, kuten tauluja vai käsitelläänkö tauluihin tallennettuja tietoja”. /1, s.67/

6.2.1 Taulun luonti

Uuden taulun luominen tapahtuu CREATE TABLE -komennolla. Taululle annetaan nimi, määritetään sen kentät, kenttien tietotyypit sekä perusavain (PRIMARY KEY).

```
mysql> CREATE TABLE osasto(  
-> ID int PRIMARY KEY,  
-> nimi varchar(20),  
-> paikkakunta varchar(20)  
-> );  
Query OK, 0 rows affected (0.03 sec)
```

Kuva 6. SQL CREATE TABLE-komento

Tämä komento luo taulun nimeltä osasto, jolla on kolme kenttää: ID, nimi ja paikkakunta. Kentän nimen perässä kerrotaan sen tietotyyppi. Kenttä ID on tietotyypiltään kokonaisluku, ja kyseinen kenttä on määritetty myös taulun perusavaimeksi. Kentät nimi ja paikkakunta ovat merkkijonoja, joiden enimmäispituus on 20 merkkiä.

6.2.2 Tauluun kirjoittaminen

Olemassa olevaan tauluun voidaan kirjoittaa INSERT INTO -komennolla. Komennolla kerrotaan minkä taulun eri kenttiin mitään tietoa tallennetaan.

```
mysql> INSERT INTO osasto (ID, nimi, paikkakunta)
values (1, 'laskutus', 'Pori');
Query OK, 1 row affected (0.00 sec)
```

Kuva 7. SQL INSERT INTO-komento

Jos tallennettavan tiedon tietotyyppi vastaa kentälle määritettyä tietotyyppiä, antaa ohjelmisto virheilmoituksen eikä tietojen tallennus tietokantaan onnistu.

6.2.3 Muokkaaminen

Tietojen muokkaaminen onnistuu UPDATE-komennolla.

```
mysql> UPDATE osasto SET nimi = 'Ulvila' WHERE ID = 1;
Query OK, 1 row affected (0.00 sec)
Rows matched: 1 Changed: 1 Warnings: 0
```

Kuva 8. SQL UPDATE-komento

Komento muuttaa nimi-kentän arvoksi 'Ulvila' niillä riveillä, joilla ehto 'ID on 1' käy toteen.

6.2.4 Poistaminen

DELETE-komennolla voidaan poistaa tietokannastakannasta rivejä.

```
mysql> DELETE FROM osasto WHERE ID = 1;
Query OK, 1 row affected (0.00 sec)
```

Kuva 9. SQL DELETE-komento

Osasto-taulun rivit, joiden ID on 1, poistetaan kannasta.

6.2.5 Hakeminen

SELECT –komennon avulla kannasta voidaan hakea tietoa. SELECT –komennossa määritetään mistä taulusta, mistä kentistä ja millä hakuehdoilla tietoa haetaan sekä missä järjestyksessä tiedot esitetään. Esimerkki SELECT –komennosta:

```
SELECT nimi, paikkakunta FROM osasto WHERE ID = 1 or ID = 2;
```

Komento palauttaa niiden osastojen nimet ja paikkakunnat joiden ID on yksi tai kaksi. /7/

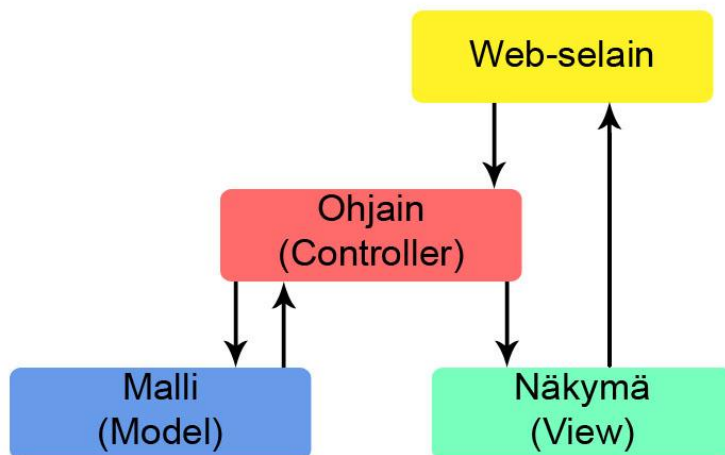
7 MVC

”Malli-Näkymä-Ohjain (Model-View-Controller) -malli eli MVC-malli on 1970-luvulla Xeroxin tutkimuslaboratoriossa kehitetty arkkitehtuurimalli graafisiin käyttöliittymiin perustuvien sovellusten toteuttamiseen. MVC-mallin mukaiset toteutukset perustuvat luokkiin ja olioihin Javan ohella myös PHP-kielessä.

Malli (Model) vastaa MVC-mallin mukaisessa toteutuksessa sekä pysyvän (persistent) tietosisällön varastoinnista että tietosisällön muokkaamiseen liittyvien sääntöjen toteuttamisesta. Usein sanotaan, että malli toteuttaa sovelluksen liiketoimintalogiikan (business logic). Malli on ainoa osa sovellusta, joka on yhteydessä tietovarastoon.

Näkymä (View) esittää sovelluksen tietosisällön käyttäjälle. Verkkopalvelussa yleisin tiedon esitysmuoto on luonnollisesti HTML.

Ohjain (Controller) tai kontrolleri toimii MVC-mallin mukaisen sovelluksen sydämenä ja aivoina. Ohjain ottaa vastaan sovellukselle esitetyt HTTP-protokollan mukaiset pyynnöt. Pyyntöön perusteella ohjain muuttaa tarvittaessa sovelluksen tilaa mallin avulla ja valitsee asiakkaalle välitettävän näkymän”. /12 s.110-112/



Kuva 10. MVC-mallin toimintaperiaate

Kuvassa 10 on esitettyä MVC-mallin yleinen toimintaperiaate. Aluksi Web-selain tekee HTTP-pyynnön (HTTP-request) ohjaimelle. Ohjain ottaa pyynnön vastaan ja antaa mallille tehtäväksi esimerkiksi HTTP-pyynnön mukaisen tietokantakyselyn. Malli palauttaa kyselyn tuloksen ohjaimelle. Lopulta ohjain yhdistää kyselyn tuloksen ja näkymän yhdeksi kokonaisuudeksi ja palauttaa sen Web-selaimelle.

8 VARASTOKIRJANPITO-OHJELMISTO

Jouko Sjöblom Oy tarvitsi varastohallintaa helpottavan, selainpohjaisen kirjanpito-ohjelmiston. Ohjelmisto haluttiin toteuttaa selainpohjaisena sen helpon ylläpidettävyyden ja päivitettävyyden vuoksi ja jotta sen käyttö onnistuu mistä tahansa. Koska PHP oli minulle ohjelmointikielenä ennestään tuttu, oli sen valinta ohjelmiston toteuttamiseen selvä. Ilmainen ja helppokäyttöinen MySQL-tietokanta sopii erinomaisesti yhteen PHP:n kanssa ja sitä käytettiin myös tämän ohjelmiston toteutukseen.

8.1 Tietokannan suunnittelu

Tietokanta muodostaa nykyaikaisten sovellusten perustan. Vaikka käyttäjälle päin näkyvät lähinnä käyttöliittymä ja raportit, on tärkeä suunnitella ja rakentaa perustana oleva tietokanta hyvin. Jos tietokanta on huonosti suunniteltu, ei sovelluksesta tule koskaan onnistunutta, koska sovellusohjelmilla joudutaan usein paikkaamaan hankalia tietorakenteita. Hyvin suunniteltu tietokanta taas helpottaa ohjelmointia suuresti ja muodostaa sovelluksen kanssa vankan perustan. /10 s.20/

Yritykseltä ei tullut tarkkaa ohjelmistomäärittelyä, vaan luettelo tarvittavista toiminnoista, joita ohjelmistolla pitäisi pystyä suorittamaan (Liite 1).

MATERIAALI -taulu sisältää tiedot varastoon tulevien materiaalien/raaka-aineiden koodit, ulkoiset mitat, puulajin, päiväykset varastoon tuomiselle ja tuotantoon viemiselle sekä toimipisteen, jonka varastoon se on sijoitettu.

TUOTE on lähes sama kuin materiaali. Eroavaisuuksia ovat lisätty kenttä laatu, joka kertoo tuotteen tuotetyypin ja kenttä date_tuotantoon on korvattu kentällä rahtikirjanumero.

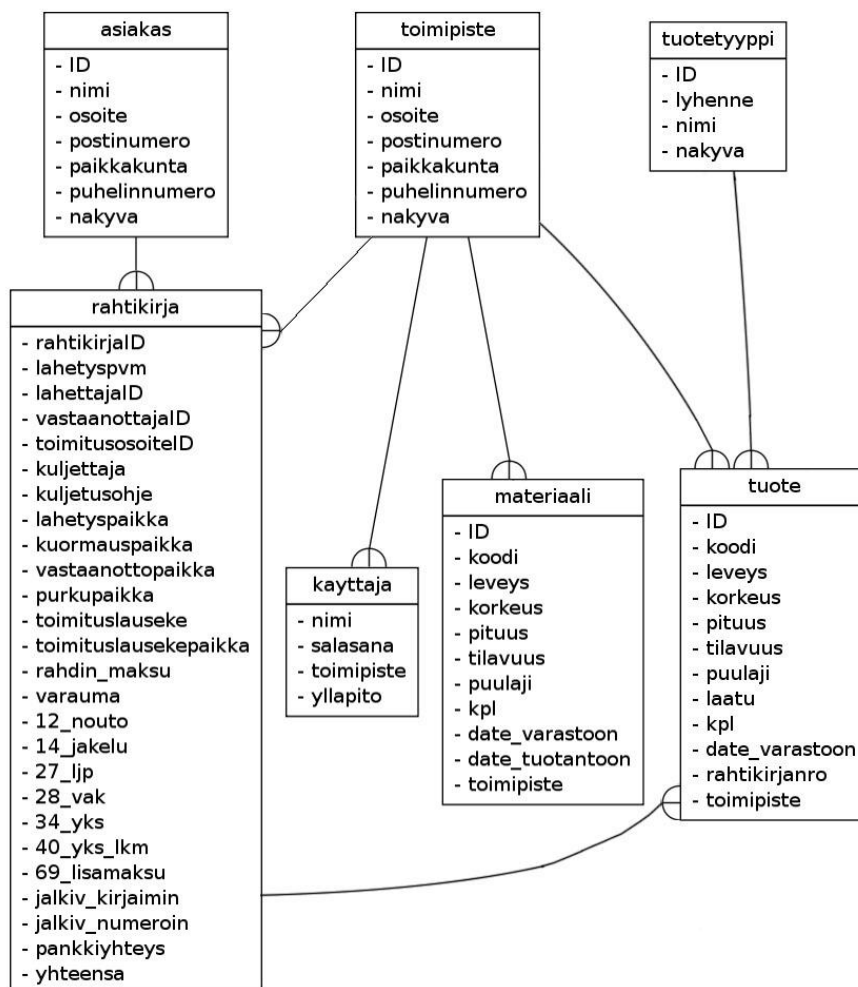
Taulussa TUOTETYYPPI on listattuna tuotetyyppien nimet ja lyhenteet.

KAYTTAJA –taulu on vain käyttäjien tunnistamista varten. Siihen talletetaan käyttäjien nimet, salasanat ja toimipisteet.

ASIAKAS ja TOIMIPISTE ovat rakenteeltaan identtiset. ASIAKAS –taulua käytetään rahtikirjan luomisessa vastaanottajan tietojen tallettamiseen ja vastaavasti TOIMIPISTE -taulua lähettäjän tietojen tallettamiseen. Lisäksi TOIMIPISTE määrittelee materiaalien, tuotteiden ja käyttäjien sijainnit

RAHTIKIRJA –tauluun talletetaan rahtikirjalomakkeen tiedot, kun rahtikirjaa luodaan.

Tietokanta on suunniteltu niin, että käyttöliittymän kautta ei kannasta pysty poistamaan mitään. Sen vuoksi tauluihin ASIAKAS, TOIMIPISTE ja TUOTETYYPPI on laitettu näkyvä-kenttä, jolla käyttäjä voi ”poistaa” esimerkiksi asiakkaan osoitelistaa.



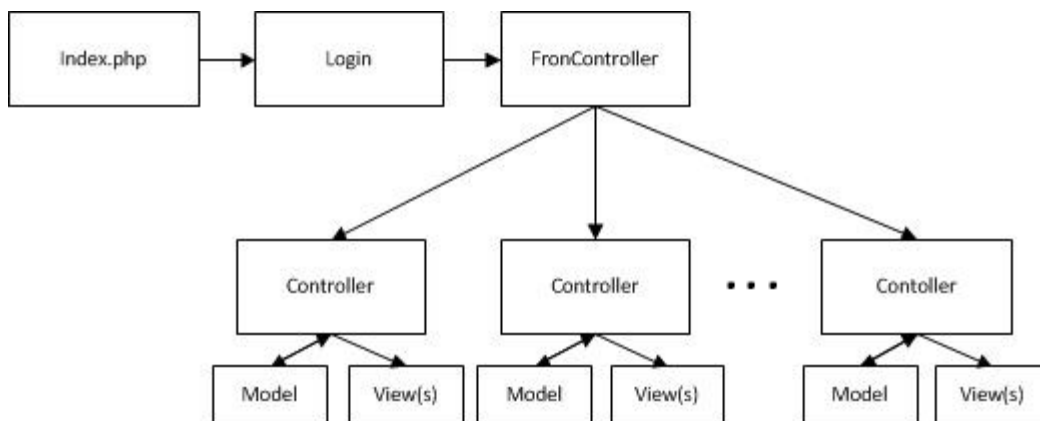
Kuva 11. Taulut ja niiden väliset yhteydet

8.2 Käyttöliittymä

8.2.1 FrontController

Ohjelmisto toteutettiin käyttämällä MVC-mallia. MVC-mallin käyttäminen helpottaa ja nopeuttaa kehitystyötä, vähentää tarvetta kirjoittaa samaa koodia useasti sekä selkeyttää koodin ymmärtämistä. Alun perin tarkoitukseni oli käyttää jotain valmiista MVC-frameworkkiä, mutta lopulta päädyin kehittämään oman, yksinkertaistetun MVC-tyyppisen ratkaisun.

Sivusto on toteutettu jakamalla se lohkoihin. Jokaisella lohkolla on oma ohjain ja malli sekä tarvittavat näkymät. Koko sivustoa ohjaa niin sanottu FrontController (etuohjain), jonka ensisijainen tehtävä on lukea URL osoiteriviltä ja ladata sen perusteella oikea Controller ja ajaa pyydetty funktio. Jos URL on virheellinen tai vastaavaa Controlleria ei löydy, antaa FrontController virhettä vastaavaan virheilmoituksen.



Kuva 12. Sivuston rakenne

Kuvassa 12 on esitettyä sivuston runko. FrontControllerille pääsee vasta, kun kirjautuminen on suoritettu onnistuneesti.

8.2.2 Sisään kirjautuminen

Varastonhallinta-järjestelmä on sijoitettu vuokratulle Web-palvelimelle, jolla on julkinen IP-osoite ja domain. Tämä tarkoittaa, että kuka tahansa voi ottaa selaimellaan yhteyttä palvelimeen ja käyttää ohjelmistoa. Tämä on kuitenkin estetty pakottamalla käyttäjä kirjautumaan sisään järjestelmään ennen kuin sitä voidaan käyttää.

Käyttäjän tullessa sivulle hänelle esitetään kirjautumislomake, jossa on kaksi kenttää: nimi ja salasana. Kun lomake on lähetetty, järjestelmä ottaa vastaan syötetyn nimen ja salasanan ja vertaa niitä tietokannassa oleviin käyttäjätietoihin.

```

2
3 function login($nimi, $salasana){
4
5     $mysqli_init = $this->connect();
6
7     $nimi      = $mysqli_init->real_escape_string($nimi);
8     $salasana  = $mysqli_init->real_escape_string($salasana);
9
10    $salasana  = md5($salasana);
11
12    $sql_lause = "select * from kayttaja where nimi='$nimi' and salasana='$salasana'";
13
14    if (!$result_init = $mysqli_init->query($sql_lause))
15        die($mysqli_init->error);
16
17    $nro_rows  = mysqli_num_rows($result_init);
18
19    if($nro_rows == 1){
20
21        $row = $result_init->fetch_array();
22
23        $_SESSION['nimi']      = $row[0];
24        $_SESSION['toimipiste'] = $row[2];
25        $_SESSION['yllapito']  = $row[3];
26
27        header('location: index.php');
28    }
29    else{
30        session_destroy();
31        header('location: index.php');
32    }
33 }
34

```

Kuva 13. Sisäänkirjautumis tietojen vertaaminen tietokantaan

Kuvassa 13 on esitettyä kirjautumisen yhteydessä ajettava funktio. Rivillä 5 kutsutaan funktiota *connect()*, joka alustaa ja luo tietokantayhteyden ja palauttaa syntyneen objektin *\$mysqli_init* –muuttujaan. Tämän jälkeen tarkastetaan syötteet SQL-injektion varalta. SQL-injektiolla tarkoitetaan muun muassa lomakkeiden kautta an-

nettavia SQL-komentoja joilla pyritään aiheuttamaan ilkeävaltaa tai pääsemään sisälle järjestelmiin, joihin käyttäjällä ei ole oikeutta. Rivillä 10 salasanalle tehdään *md5()* – funktio, joka laskee salasanasta 32-merkkisen MD5-koosteen. Käyttäjätunnuksen luonnin yhteydessä kaikille salasanoille tehdään sama toimenpide, jolloin tietokannassa ei lue käyttäjän salasanaa selväkielisenä tekstinä.

Tämän jälkeen muodostetaan ja suoritetaan SQL-kielinen lause, joka hakee KAYTTAJA –taulusta lomakkeeseen syötettyä nimeä ja salasanaa vastaavaa käyttäjätunnusta. Jos kyselyllä löytyy tulos, talletetaan istuntopaikkoihin käyttäjän nimi, toimipiste ja käyttäjäluokka jonka jälkeen käyttäjä ohjataan takaisin *index.php*-sivulle. Tämän jälkeen tarkistetaan onko nimi-istuntopaikkoihin talletettu arvo. Käyttäjä voi jatkaa ohjelmiston käyttöä jos kirjautuminen onnistuu. Jos istuntopaikkoihin ei ole asetettu arvoa, joutuu käyttäjä yrittämään kirjautumista uudelleen.

8.2.3 Materiaalit ja tuotteet

Kun käyttäjä on onnistuneesti kirjautunut sisään, voidaan ohjelmiston toimintoja alkaa käyttää. Ohjelmiston keskeisimmät toiminnot ovat materiaalien ja tuotteiden lisääminen varastoon, materiaalien vieminen tuotantoon sekä tuotteiden lähettäminen asiakkaalle. Näiden toimintojen lisäksi ohjelmistossa on ylläpitoon liittyviä toimintoja, kuten käyttäjien, tuotetyyppien ja asiakastietojen hallinta.

Materiaalien ja tuotteiden lisääminen varastoon tapahtuu lomakkeella, jonka käyttäjä täyttää. Lomakkeen käsittelyssä tarkistetaan onko jokin lomakkeen kentistä tyhjä sekä onko kenttiin syötetty tieto oikean tyyppistä. Virheellinen syöte tai tyhjä kenttä antavat virheilmoituksen, eikä lomakkeen tietoja talleteta järjestelmään. Jos lomakkeeseen syötetyt tiedot hyväksytään, kirjoitetaan tietokantaan lomakkeen tiedot sekä tallennushetken päiväys.

Koodi	Puulaji	Laatu	W	H	L	Kpl / paketti	
	Mänty - RW	liimapalkki (li)					Lisää

liimapalkki (li)
 sormijatkettu (ja)
 raakki (ra)
 Väliseinätolppa (VS)
 Muottituki (MT)
 Tarvike / terä (TA)
 NÄYTE (NT)
 Log (LG)

Kuva 14. Lomake tuotteiden lisäämiseen.

Varastoon lisätyt materiaalit ja tuotteet listataan omaan taulukoihin kuvan 15 mukaisesti. Taulukon oikeassa reunassa sijaitsevalla valintaruudulla valitaan ne materiaalit tai tuotteet, jotka halutaan siirtää järjestelmässä eteenpäin.

360	ZI04	42	66	6000	108	1.796256	WW	VS	2012-11-16 14:05:03	<input type="checkbox"/>
355	ZI30	90	315	12000	8	2.7216	WW	li	2012-11-15 15:12:39	<input type="checkbox"/>
354	ZI48	115	315	12000	3	1.3041	WW	li	2012-11-15 14:37:27	<input type="checkbox"/>
353	ZI62	140	315	12000	3	1.5876	WW	li	2012-11-15 13:10:25	<input type="checkbox"/>
352	ZI41	115	225	8000	3	0.621	WW	li	2012-11-15 13:08:19	<input type="checkbox"/>
351	ZI66	140	450	12000	5	3.78	WW	li	2012-11-15 09:40:01	<input type="checkbox"/>
350	ZI48	115	315	12000	9	3.9123	WW	li	2012-11-15 09:09:44	<input type="checkbox"/>
349	ZI48	115	315	12000	9	3.9123	WW	li	2012-11-15 09:08:26	<input type="checkbox"/>
348	LG	88	172	11800	24	4.2865152	RW	LG	2012-11-15 07:24:25	<input type="checkbox"/>
347	LG	88	172	11800	36	6.4297728	RW	LG	2012-11-15 07:22:54	<input type="checkbox"/>
346	LG	88	172	11800	36	6.4297728	RW	LG	2012-11-15 07:21:35	<input type="checkbox"/>

Luo lähetyalista

Kuva 15. Tuote-taulukko.

8.2.4 PDF

Kun valmis tuote lisätään varastoon ja kirjataan järjestelmään, ohjelmisto luo automaattisesti kyseistä tuotetta vastaavan pakettisetelin. Pakettiseteliin kirjataan tuotetta vastaavat tiedot, se tulostetaan paperille ja kiinnitetään tuotteeseen. Kun tuotteet lähetetään asiakkaalle, luodaan rahtikirja, johon kirjataan muun muassa asiakkaan osoitetiedot sekä luettelo tuotteista. Nämä dokumentit luodaan PDF-muodossa, jotta ne saadaan helposti tulostettua ja säilytettäväksi.

PHP:ssä ei ole valmiita työkaluja PDF-tiedostojen luontiin. Tämän vuoksi PDF:n luomista varten jouduin käyttämään valmiita FPDF-luokkaa, joka on ilmainen PDF-tiedostojen luomiseen ja muokkaamiseen tarkoitettu työkalu. FPDF-luokan avulla voidaan suoraan luoda pakettiseteli käyttämällä luokan funktioita. Rahtikirjan luon-

nissa käytetään tyhjää rahtikirjapohjaa, jonka päälle tiedot kirjoitetaan. Rahtikirjapohja on PDF-muodossa, eikä FPDF-luokka tarjoa mahdollisuutta avata jo aiemmin luotua PDF-tiedostoa. Tämän vuoksi jouduin käyttämään FPDI-luokkaa, jota voitaisiin kutsua FPDF-luokan lisäosaksi, jolla voidaan avata valmiita PDF-tiedostoja muokattavaksi. Lisäksi käytössä oli toinen apu-luokka, HTMLtoPDF, joka mahdollisti HTML-kielisen koodin kirjoittamisen PDF:ään.

9 YHTEENVETO

Ohjelmisto on otettu yrityksessä hyvin vastaan ja se on päivittäisessä käytössä. Ohjelmistoon toteutettiin alkuperäisen suunnitelman mukaiset toiminnot sekä muutama muu ominaisuus, joiden puute huomattiin testauksen yhteydessä. Siihen lisättiin yrityksen toiven lisäksi hallintapaneeli käyttäjille, toimipisteille ja tuotetyypeille sekä pienimuotoinen asiakasrekisterijärjestelmä. Noin neljän kuukauden testi- ja käyttöajan aikana ohjelmiston kautta on kulkenut lähes 400 yksittäistä tuotetta ja yli sata rahtikirjaa.

Alkuperäinen aikataulu ohjelmiston käyttöönottoon venyi usealla kuukaudella pitkään kestäneen palvelintilan hankintaprosessin sekä yrityksen kommunikaation hitauden vuoksi. Ohjelmisto saatiin kuitenkin lopulta toimimaan yrityksen hankkimalle palvelimelle. Tulevaisuudessa ohjelmistoa on tarkoitus laajentaa muun muassa kehittämällä lujustestijärjestelmä.

Ohjelmiston tekeminen yritykselle omatoimisesti ja vastuullisesti oli haastava kokemus. Suunnitelu- ja ohjelmointityö etenivät suhteellisen hitaasti. Oma ajankäyttöä olisi voinut suunnitella paremmin ja yhteydenpitoa yritykseen tehostaa. Ohjelmistoon voin kuitenkin olla tyytyväinen, sillä se vastaa yrityksen esittämiä toivomuksia ja jatkossa tehtävät laajennukset on helppo toteuttaa.

LÄHTEET

1. Heinisuo, R 2004. PHP ja MySQL: Tietokantapohjaiset verkkopalvelut. 3uud. p. Jyväskylä. Gummerus Kirjapaino Oy.
2. Rantala, A 2002. PHP Web ohjelmoinnin peruskirja. Porvoo. WS Bookwell.
3. Wikipedia: PHP. 2012. Viitattu 3.11.2012. <http://fi.wikipedia.org/wiki/PHP>
5. Ohjelmistoputka: JavaScript-perusopas: Osa 1 – Perusteet. 2007. Viitattu 7.11.2012. http://www.ohjelmointiputka.net/oppaat/opas.php?tunnus=js_01
6. Wikipedia. 2012. Viitattu 7.11.2012. <http://fi.wikipedia.org/wiki/JavaScript>
7. MySQL kotisivut. 2012. viitattu 7.11.2012. <http://www.mysql.com>
8. Ohjelmistoputka: PHP-ohjelmointi: Osa 14 - Olio-ohjelmointi. 2011. Viitattu 5.11.2012. http://www.ohjelmointiputka.net/oppaat/opas.php?tunnus=php_14
9. Ohjelmistoputka: Käytännön PHP-opas: Osa 7 - Evästeet ja istunnot. 2003. Viitattu 5.11.2012. <http://www.ohjelmointiputka.net/oppaat/opas.php?tunnus=phpj7>
10. Hovi, A., Huotari, J., Lahdenmäki, T. 2003. Tietokantojen suunnittelu & indeksointi. Porvoo. WS Bookwell
11. W3C-school. 2012. Viitattu 11.11.2012. <http://www.w3schools.com/>
12. TUT: Edistynyt PHP-ohjelmointi. 2005. Viitattu 14.11.2012. <http://matwww.ee.tut.fi/hmopetus/hm-ohj/2005/pruju/hmohj05-108-119.pdf>

Liitteet

1. Jouko Sjöblom Oy, varastohallinta

Liite 1

Jouko Sjöblom Oy

Jurva

Varastokirjanpito-ohjelmisto

Toteutettava ohjelmisto tulee uuden, parhaillaan rakennettavan liimapalkkitehtaan varastonhallinnan ja taloushallinnon työvälineeksi.

ALKUVAIHEESSA TARVITTAVAT TOIMINNOT

- sisään tulevien raaka-ainepakettien kirjaus pakettinumeroiden perusteella, numero on 5-8 numeroinen sarja. Pakettinumeron takaa löytyy puulaji, dimensio paksuus x leveys millimetreinä, pituus millimetreinä, kappaleiden määrä paketissa sekä kuutiomäärä. Lisäksi kirjautuu päivämäärä, jolloin paketti kirjataan varastoon.
- varastossa olevat paketit tulee voida poistaa tuotantoon, tällöin kirjautuu pakettitietojen lisäksi päivämäärä, jolloin poisto varastoon tapahtuu.
- uuden paketin luonti varastoon, pakettitiedot kuten edellä, mutta lisäksi kirjataan laatu, onko kyseessä liimapalkki, sormijatkettu sahatavara, raakki yms. Tämä erottelu tapahtuu kirjainlyhenteillä.
- raaka-aineet ja valmiit tuotteet kirjautuvat eri varastoihin, josta niitä voidaan lajitella
- varastossa olevista paketeista luodaan kuormakirja / lähetyslista, jossa näkyy paketit numeroittain, laatu ja kuutiomäärä.
- kun kuormakirja on luotu, paketit poistuvat varastosta, ja kyseinen kuormakirja jää tietoineen numerjärjestyksessä talteen. Luonnin yhteydessä voidaan tulostaa paperiversioita esim 2-4 kpl.
- paketteja voidaan lähettää myös toisiin varastoihin alihankkijoille em tavalla, ja alihankkijat pystyvät tekemään samat asiat tuleville sekä valmistuville paketeille. Alihankkijat eivät pääse muihin kuin omassa varastossa oleviin tietoihin.

Järjestelmä pyörii nettitunnuksien takana jollain serverillä.