

Matias Määttä

TOIMINNANOHJAUSJÄRJESTELMÄN SUUNNITTELU JA
TOTEUTUS

Tietotekniikan koulutusohjelma
Ohjelmistotekniikan suuntautumisvaihtoehto
2015

TOIMINNANOHJAUSJÄRJESTELMÄN SUUNNITTELU JA TOTEUTUS

Määttä, Matias
Satakunnan ammattikorkeakoulu
Tietotekniikan koulutusohjelma
Syyskyy 2015
Ohjaaja: Hentunen, Ilmari
Sivumäärä: 36

Asiasanat: toiminnanohjausjärjestelmä, web-ohjelmointi, tietokanta

Tämän opinnäytetyön tavoitteena oli suunnitella ja toteuttaa Kartanokoti Oy:lle toiminnanohjausjärjestelmä, joka helpottaa yrityksen rakennusprojekteja ja parantaa niiden hallittavuutta. Tavoitteena oli päästä eroon paperiasioista ja saada tiedot kerättyä talteen yhteen ja samaan paikkaan.

Toiminnanohjausjärjestelmä tehtiin internetselaimella toimivaksi käyttäen nykyaikaisia Web-tekniikoita. Järjestelmän tekemiseen käytettiin muun muassa HTML 5 -, PHP-, CSS-, MySQL-kieliä. Toiminnanohjausjärjestelmään tehtiin kaikki keskeiset ominaisuudet, kuten projektien, projekti aikataulun ja asiakastietojen hallinta. Asiakkaalle tehtiin mahdollisuus kirjautua järjestelmään omilla tunnuksilla, ja näin saatiin toiminnanohjausjärjestelmästä kaikki hyöty irti.

ENTERPRISE RESOURCE PLANNING DESIGN AND IMPLEMENTATION

Määttä, Matias

Satakunta University of Applied Sciences

Degree Programme in Information Technology

September 2015

Supervisor: Hentunen, Ilmari

Number of pages: 36

Keywords: enterprise resource planning, ERP, web programming, database

The purpose of this thesis was to plan and carry out enterprise resource planning. Kartanokoti Oy company ordered system. Purpose of the ERP is to help the company's construction projects and to improve their controllability. The aim was minimize paperwork, save and collect information in same place.

ERP have made to work by web browser using modern web technology. The system was implemented by using HTML 5, PHP, CSS and MySQL programming languages. To the System was made all the essential features like control of project-, project timetable and customer information. It was made opportunity for customer to log in to system with own password and system become more useful.

SISÄLLYS

LYHENTEET	5
1 JOHDANTO.....	6
2 YRITYSESITTELY	6
3 TOIMINNANOHJAUSJÄRJESTELMÄ	7
4 KÄTETYT TEKNIIKAT JA OHJELMAT	8
4.1 Web-palvelin.....	8
4.2 HTML 5	8
4.2.1 Määrittely	9
4.2.2 Alueiden määrittely	10
4.2.3 Lomake	10
4.3 PHP	12
4.4 MySQL	12
4.4.1 MySQL ja PHP.....	13
4.5 Notepad++	14
5 SUUNNITTELU	15
6 TOTEUTUS	15
6.1 Kirjautuminen	16
6.2 Pääkäyttäjän näkymä	17
6.2.1 Yrityksen tehtävät.....	17
6.2.2 Projektit	17
6.2.3 Hankintasopimukset	18
6.3 Projektinäkymä	20
6.3.1 Kuvat	21
6.3.2 Projekti aikataulu.....	22
6.3.3 Muut osiot	24
6.4 Asiakkaan näkymä	27
7 TIETOTURVA.....	30
7.1 Merkkijonojen tarkistaminen	30
7.2 Salasanojen suojaaminen	31
7.3 Tiedostojen tallentaminen.....	31
8 TIETOKANNAN TAULUT JA RAKENNE.....	32
9 TESTAUS JA YHTEENVETO	35
LÄHTEET.....	36

LYHENTEET JA TERMIT

HTML	Hypertext Markup Language. Merkintäkieli.
PHP	Hypertext Preprocessor. Suosittu lisenssivapaa (open source) skriptikieli, jota käytetään Web-ympäristössä. PHP mahdollistaa dynaamisten sivujen luonnin.
MySQL	Relaatiotietokantaohjelmisto. Käytetään usein Web-palveluissa tiedon pysyvämpään tallentamiseen.
CSS	Cascading Style Sheets. Voidaan määritellä Web-sivuille tyyliteltyjä ulkoasuja.
JavaScript	Dynaaminen komentosarjakieli, jota käytetään Web-ympäristössä. Mahdollistaa dynaamisen toiminnan tekemisen.
ERP	Enterprise Resource Planning. Toiminnanohjausjärjestelmä.

1 JOHDANTO

Opinnäytetyöksi tein toiminnanohjausjärjestelmän. Idea tästä tuli Kartanokoti Oy:n toimitusjohtaja. Hän oli jo pitkään miettinyt, miten yritystä ja yrityksen toimintatapoja voisi kehittää. Toimitusjohtaja kertoi suunnitelmistaan ja kiinnostuin asiasta.

Tarkoituksena oli tehdä sellainen sovellus, joka helpottaisi ja selkeyttäisi myynnin ja taloprojektien hallintaa. Taloprojektin kaikki tiedot, liitteet, aikataulut, tiedostot ja kuvat haluttiin samaan paikkaan. Asiakkaille haluttiin myös mahdollisuus päästä seuraamaan taloprojektin etenemistä internetistä. Asiakkaalle haluttiin tehdä kaupanteon jälkeisistä toimista selkeä ja helppo tehtävälistä, jota seuraamalla ja noudattamalla taloprojekti valmistuisi ajallaan.

Toteutin toiminnanohjausjärjestelmän web-pohjaisena internetselaimella toimivaksi. Itsellä oli kokemusta PHP-skriptikielestä ja MySQL-tietokannoista. Järjestelmästä tuli näin laitteistoriippumaton. Etuna on, että järjestelmään pääsee kirjautumaan missä tahansa ja talokaupat voi suoraan kaupantekohetkellä lisätä järjestelmään uutena taloprojektina.

2 YRITYSESITTELY

Opinnäytetyö tehtiin Kartanokoti Oy:lle. Kartanokoti on talotehdastoimittaja, joka myy hirsiiä ja hirsitalopaketteja avaimet käteen -periaatteella. Kartanokoti Oy on perustettu vuonna 2012. Yritys sijaitsee Turussa ja sen liikevaihto vuonna 2014 oli 838 000 euroa.

3 TOIMINNANOHJAUSJÄRJESTELMÄ

Toiminnanohjausjärjestelmästä käytetään yleisesti lyhennettä ERP. Lyhenne tulee englannin kielen sanasta Enterprise Resource Planning. Vapaasti käännettynä enterprise tarkoittaa yritystä, resource tarkoittaa voimavaroja tai resursseja ja planning tarkoittaa suunnittelua.

Toiminnanohjausjärjestelmä on tietojärjestelmä, joka pyrkii ohjaamaan yrityksen toimintaa ja yhdistämään sen eri toimintoja helposti hallittavaksi kokonaisuudeksi. Toimintoja ovat esimerkiksi tuotanto, laskutus ja kirjanpito. ERP-järjestelmät pitävät sisällään yleensä erilaisia moduuleja, joita saa tarvittaessa otettua käyttöön. Moduuleja ovat muun muassa tuotannonohjaus, dokumenttien-, asiakas-, valmistuksen-, toimituksen- ja projektienhallinta. Pienillä yrityksillä ovat erilaiset tarpeet, kuin isoilla yrityksillä.

Valmiita maksullisia ja ilmaisia vapaan lähdekoodin ERP-järjestelmiä on tarjolla, mutta järjestelmän pystyy myös suunnittelemaan ja tekemään itse suoraan oman yrityksen tarpeisiin. ERP-järjestelmää hankittaessa kannattaa miettiä tarkasti mihin käyttöön järjestelmä tulee ja kannattaako sitä alkaa suunnittelemaan ja tekemään itse alusta lähtien vai löytyykö hyviä maksullisia tai ilmaisia vaihtoehtoja. ERP-järjestelmä on yritykselle yleensä kallis hankinta, mutta ajan kuluessa se maksaa itsensä takaisin. Hankintahintaan kannattaa sisällyttää työntekijöiden kouluttamiskulut.

ERP-järjestelmästä saadaan hyötyä vasta sitten kun sitä oikeasti käytetään. Yrityksen kaikkien työntekijöiden pitäisi sitoutua käyttämään järjestelmää. ERP-järjestelmän tekijän pitäisi osata ottaa huomioon sen käyttäjät, jotta järjestelmää olisi helppoa ja vaivatonta käyttää. Erityisesti tietotekniikkaa vähemmän käyttäneille järjestelmän käyttöönotto ja käyttäminen voi olla vaikeakin asia.

ERP-järjestelmän tarkoituksena on tehostaa ja automatisoida yrityksen toimintoja ja toimintatapoja. Järjestelmä auttaa yritystä hallitsemaan ja seuraamaan liiketoimintaa. Järjestelmään määritellään strategiat ja tavoitteet, jotka ohjaavat sen käyttäjää tehok-

kaaseen ja yksinkertaiseen toimintaan. ERP-järjestelmää käyttämällä pyritään vähentämään paperisia dokumentteja ja manuaalisia toimintoja. Yrityksen prosesseja ja toimintoja on helppo hallita yhdestä ja samasta järjestelmästä. ERP-järjestelmän kytkeminen internetiin mahdollistaa lisäksi toimittajan moduuleiden käyttämisen. (Toiminnanohjausjärjestelmä (ERP) 2015.)

4 KÄTETYT TEKNIIKAT JA OHJELMAT

Työn tekemiseen käytettiin HTML-, PHP-, MySQL-, JavaScript- ja CSS-kieliä. Ulkoasu sivustolle saatiin käyttämällä HTML:ää, CSS:ää ja kuvia. PHP-skriptikieltä käytettiin sivuston toimintojen tekemiseen. PHP:n avulla tallennetaan esimerkiksi liitetiedostoja palvelimelle ja tietoja MySQL-tietokantaan. JavaScriptiä käytettiin kun haluttiin sivustolle dynaamista toimintaa, johon HTML ei pystynyt. Työn tekemiseen käytettiin Notepad++ -tekstieditoria.

4.1 Web-palvelin

Käytimme web-palvelimena jo ennestään Kartanokodilla käytössä ollutta palvelintilaa. Palvelimelle sai luotua rajoittamattomasti tietokantoja, ja tiedontallennustilaa oli riittävästi. Palvelin tukee yleisempiä teknologioita: PHP 5.4 ja MySQL 5. Palvelimelta löytyi phpmyadmin-ohjelma, jolla pystyi kätevästi luomaan ja hallitsemaan tietokantoja.

4.2 HTML 5

HTML 5 on merkintäkielen uusin versio, ja se julkaistiin lokakuussa 2014. Sivustojen luomisessa käytettiin HTML 5 -kieltä, jonka uudet ominaisuudet olivat vahvasti teke- misessä mukana.

HTML 5 toi mukanaan uusia ominaisuuksia, jotka helpottavat web-kehittäjää ja käyttäjiä. Monet HTML 5:den ominaisuudet on suunniteltu erityisesti kännykällä toimiviksi. Esimerkiksi käyttämällä tekstikentän tyyppinä number-arvoa, kännykkä osaa avata tekstiä syöttäessä automaattisesti numeronäppäimistön. Tässä osiossa kerrotaan mitä HTML 5:den ominaisuuksia työn tekemisessä käytettiin.

4.2.1 Määrittely

Sivun määrittelyn voi tehdä kätevämmän. Ei tarvita pitkää doctype-osoitetta. Tyylitiedostot eivät tarvitse pitkiä määritelmiä, ja JavaScriptin voi sivulle liittää lyhyemmin kirjoitettuna. Kuvasta 1 huomaa hyvin uudistuneen määrittelyn. (W3schools HTML(5) tutorial 2015.)

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4      <meta charset="UTF-8">
5      <title>Otsikko</title>
6  </head>
7  <body>
8      Sisältö
9  </body>
10 </html>
```

```
1  <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
2      "http://www.w3.org/TR/html4/strict.dtd">
3  <html>
4  <head>
5      <meta http-equiv="content-type" content="text/html; charset=UTF-8">
6      <title>Otsikko</title>
7
8      <link rel="stylesheet" href="tyyli.css" type="text/css" />
9      <script type="text/javascript" src="jscripti.js"></script>
10 </head>
11 <body>
12
13 </body>
14 </html>
```

Kuva 1. Uusi ja vanha määrittely.

4.2.2 Alueiden määrittely

Enää ei tarvitse alueita määrittellä `<div id="header"></div>` -tyylillä. Html 5 toi mukanaan helpomman ja nopeamman tavan. Alueiden määrittely onnistuu nyt suoraan ilman divejä (kuva 2). Sivun alkuun kannattaa lisätä

`<script src="http://html5shim.googlecode.com/svn/trunk/html5.js">`, jotta sivusto näkyisi kaikilla selaimilla oikein. Sivuston .css-tyylitiedostoon pitää laittaa seuraavat tyylimäärittelyt: `header, nav, section, article, footer {display:block;}`.

```
1 <nav>Navigaatio</nav>
2 <section>Sisältö</section>
3 <footer>Sivuston alaosa</footer>
```

Kuva 2. Alueiden Määrittely.

4.2.3 Lomake

Lomakkeeseen pystyy tekemään käteviä pieniä lisiä Html 5:den avulla. Kalenterista saa poimittua oikean päivämäärän tai viikon lomakekenttään, ja kenttien tarkistamista on helpotettu. Kuva 3 havainnollistaa kalenterin toimintaa. Käytin sivustolla viikon lisäystä, sähköpostin tarkistamista (kuva 4), tiedoston latausominaisuuksia ja placeholder-ominaisuutta (kuva 5).

Vapaasti käännettynä placeholder tarkoittaa paikkamerkkiä. Placeholder on kätevä ominaisuus, jos haluaa säästää lomaketilaa tekstiltä. Se tekee lomakkeen tekstikenttään himmennetyn tekstin, jotta käyttäjä tietää, mitä tietoa siihen pitää kirjoittaa. Uudet tiedostonlatausominaisuudet ovat `multiple` ja `accept`. `Multiple`-valintaa käyttämällä onnistuu kerralla useamman tiedoston lisäys lomakkeeseen ja siitä palvelimelle. `Accept`-valinnalla saa rajoitettua, mitä tiedostomuotoja hyväksytään lomakekenttään.

```

1 <form>
2   Toimitusaika:
3   <input type="week" name="toimitusaika">
4 </form>

```

Toimitusaika: Viikko 25, 2015

kesäkuu 2015

Viikko	ma	ti	ke	to	pe	la	su
23	1	2	3	4	5	6	7
24	8	9	10	11	12	13	14
25	15	16	17	18	19	20	21
26	22	23	24	25	26	27	28
27	29	30	1	2	3	4	5

Kuva 3. Viikon numero lomakkeeseen käyttäen kalenteri-ominaisuutta.

```

1 <form>
2   Sähköposti:
3   <input type="email" name="sposti">
4   <input type="submit">
5 </form>

```

Sähköposti:

! Sähköpostiosoitteeseen kuuluu @-osa. Osoitteesta eetu.esimerkki gmail.com puuttuu @.

Kuva 4. Sähköpostin tarkistaminen.

```

1 <form>
2   <input type="text" name="enimi" placeholder="Etunimi"><br>
3   <input type="text" name="snimi" placeholder="Sukunimi">
4 </form>

```

Kuva 5. Placeholder-ominaisuus lomakkeessa.

4.3 PHP

PHP on palvelin pohjainen ohjelmointikieli, jota käytetään web-ympäristössä. Lyhenne PHP tulee sanoista Hypertext Preprocessor. Ensimmäinen versio julkaistiin vuonna 1994 ja uusin versio on julkaistu vuonna 2015. Se on helppokäyttöinen ja hyvin monipuolinen ohjelmointikieli. PHP:llä pystyy helposti ottamaan yhteyden tietokantaan ja käsittelemään palvelimen tiedostoja. PHP-koodi suoritetaan palvelimella, ennen kuin se lähetetään käyttäjälle, joten PHP ei vaadi tukea selaimelta. (W3schools PHP tutorial 2015.)

PHP sisältää valmiina suuren määrän erilaisia funktioita, esimerkiksi salasanan suojaamisen, tietokantayhteyden luomiseen ja sähköpostin lähettämisen. Myös käyttäjät ovat tehneet valmiita funktioita, joita löytyy internetistä helposti. PHP-koodi upotetaan yleensä html-sivulle (kuva 6), ja palvelin osaa poimia ja suorittaa koodin siitä. PHP-koodi ei näy käyttäjälle samalla tavalla kuin HTML. Sitä ei näe sivun lähdekoodista.

```

1  <!DOCTYPE html>
2  <html>
3  <head>
4      <meta charset="UTF-8">
5      <title>Otsikko</title>
6  </head>
7  <body>
8      <?php echo "Tulostetaan tekstiä HTML-sivulle!"; ?>
9  </body>
10 </html>

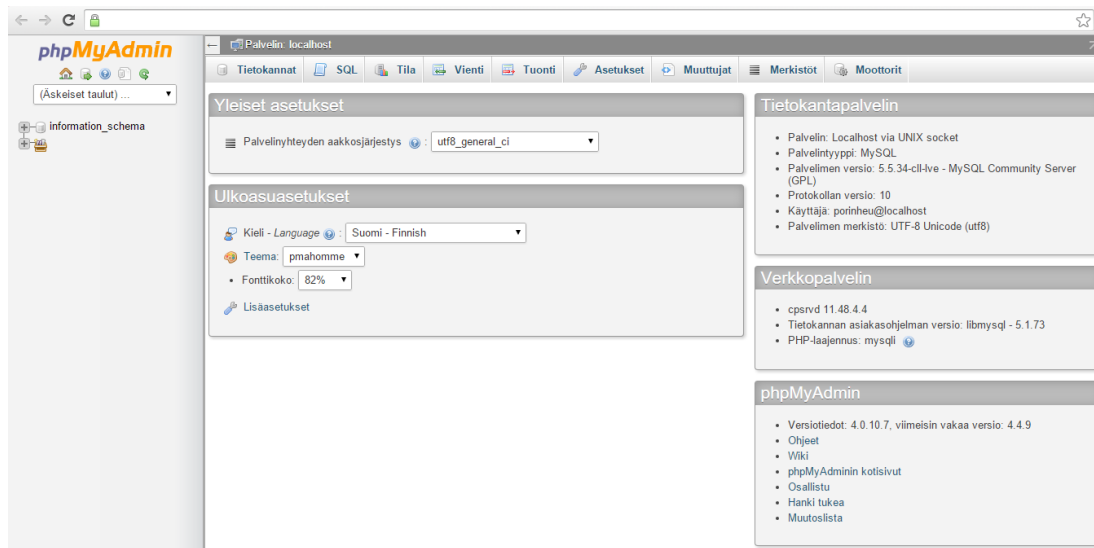
```

Kuva 6. PHP-koodin upottaminen sivulle.

4.4 MySQL

MySQL on relaatiotietokantaohjelmisto. Sitä käytetään yleensä web-palveluiden tietovarastona. Tietokantaan on helppoa ja tehokasta lisätä tietoa. Suurien tietojenkin käsittely onnistuu hyvin. Esimerkiksi tietokantaan voi tallentaa verkkokaupan tuotteita tai käyttäjätietoja. MySQL-tietokantaan luodaan tauluja ja tauluihin luodaan kenttiä ja rivejä. Tietokannasta ja tauluista haetaan tietoja SQL-kyselyillä.

Useissa web-palvelimissa ja webhotelleissa on asennettuna phpMyAdmin (kuva 7), joka on kätevä ja helppo työkalu tietokantojen luomiseen ja hallintaan. Sitä käytetään internetselaimen kautta. Tässä opinnäytetyössä käytin tätä kyseistä sovellusta.



Kuva 7. PhpMyAdmin.

4.4.1 MySQL ja PHP

PHP:lla yhteyden muodostaminen tietokantaan onnistuu helposti. Alkuun määritellään tietokannan nimi, salasana ja käyttäjänimi. Tämän jälkeen PHP hoitaa yhdistämisen. Tietojen hakemista varten pitää vielä määrittellä sql-lause. Sql-lausekkeeseen kirjoitetaan taulun nimi. Tietokannasta voidaan hakea tietoa taulun nimen perusteella, ja haku voidaan myös rajata. Alla olevasta kuvasta näkee hyvin, miten yhteyden muodostaminen tapahtuu. (W3schools SQL tutorial 2015.)

```

3 //Tietokanta yhteyttä varten tarvittavat tiedot
4 $servername = "localhost";
5 $username = "käyttäjänimi";
6 $password = "salasana";
7 $dbname = "tietokannan_nimi";
8
9 // Luodaan yhteys
10 $conn = mysqli_connect($servername, $username, $password, $dbname);
11 mysqli_set_charset($conn,"utf8");
12 // Tarkistetaan yhteys
13 if (!$conn) {
14     die("Yhdistäminen epäonnistui " . mysqli_connect_error());
15 }
16 // SQL-kysely lauseke
17 $sql = "SELECT * FROM `yhteystiedot`";
18
19 // Haetaan tiedot tietokannan yhteystiedot taulusta
20 $result = $conn->query($sql);
21 if ($result->num_rows > 0) {
22     while($row = $result->fetch_assoc()) {
23         //Tallennetaan taulun nimi kentän tieto $nimi muuttujaan
24         $nimi = $row["nimi"];
25     }
26 }
27 if (mysqli_query($conn, $sql)) {
28 }
29 else {
30     echo "Virhe haettaessa tietoja" . mysqli_error($conn);
31 }
32 // Suljetaan yhteys
33 mysqli_close($conn);

```

Kuva 8. Yhdistäminen tietokantaan.

4.5 Notepad++

Notepad++ on monipuolinen avoimen lähdekoodin tekstieditori (kuva 9). Notepad++ tukee monia ohjelmointikieliä ja syntaksien väriä, mikä helpottaa ohjelmointia huomattavasti. Tekstieditoriin saa useita välilehtiä auki samanaikaisesti. Ohjelman haku ja tekstin korvausominaisuudet ovat hyviä. Jonkin tietyn tekstin saa korvattua useammasta aukinaisesta välilehdestä napin painalluksella. Käytin Notepad++ -ohjelmaa toiminnanohjausjärjestelmän tekemiseen. Ohjelman saa myös suomenkielisenä.

```

1 <?php
2
3 //Tietokanta yhteyttä varten tarvittavat tiedot
4 $servername = "localhost";
5 $username = "käyttäjänimi";
6 $password = "salasana";
7 $dbname = "tietokannan_nimi";
8
9 // Luodaan yhteys
10 $conn = mysqli_connect($servername, $username, $password, $dbname);
11 mysqli_set_charset($conn,"utf8");
12 // Tarkistetaan yhteys
13 if (!$conn) {
14     die("Yhdistäminen epäonnistui " . mysqli_connect_error());
15 }
16 // SQL-kysely lauseke
17 $sql = "SELECT * FROM 'yhteystiedot'";
18
19 // Haetaan tiedot tietokannan yhteystiedot taulusta
20 $result = $conn->query($sql);
21 if ($result->num_rows > 0) {
22     while($row = $result->fetch_assoc()) {
23         //Tallennetaan taulun nimi kentän tieto $nimi muuttujaan
24         $nimi = $row["nimi"];
25     }
26 }
27 if (mysqli_query($conn, $sql)) {
28 }
29 else {
30     echo "Virhe haettaessa tietoja" . mysqli_error($conn);
31 }
32 // Suljetaan yhteys
33 mysqli_close($conn);
34

```

Kuva 9. Notepad++ -tekstieditori.

5 SUUNNITTELU

Toiminnanohjausjärjestelmää alettiin suunnitella kesänloman alussa, kun työt alkoivat. Jari kertoi, mitä ominaisuuksia haluaisi järjestelmään, ja kirjoitimme asioita ylös paperille. Suunnittelu oli mukavaa. Järjestelmä suunniteltiin tehtäväksi osissa. Alustavien suunnitelmien perusteella tehtiin tietokannan taulut. Järjestelmän tekeminen aloitettiin yrityksen hallintakäyttöliittymästä ja lopuksi suunniteltiin ja tehtiin asiakaspuolen käyttöliittymä. Järjestelmän ulkoasu muokkautui tekemisen aikana, eikä sitä paljoakaan tarvinnut miettiä erikseen. Lopuksi suunniteltiin ja viimeisteltiin asiakaskäyttöliittymän ulkoasu.

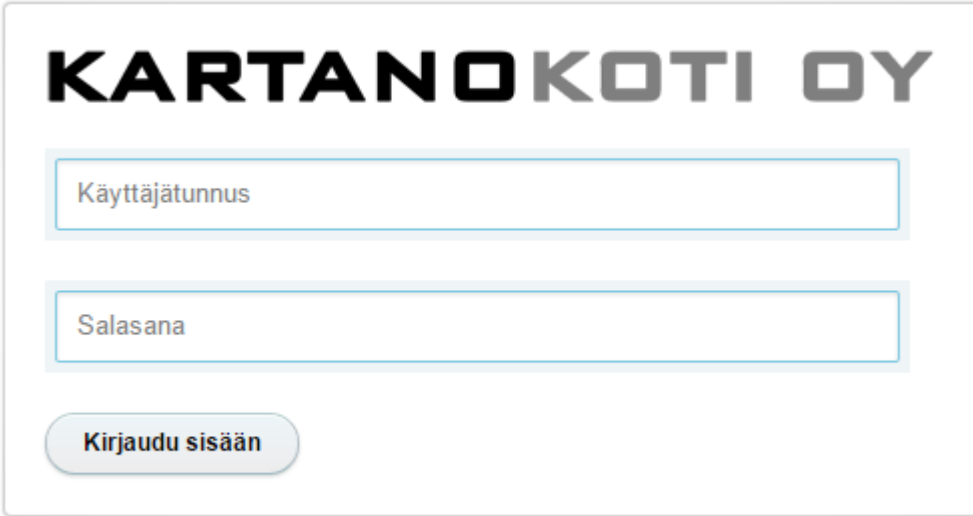
6 TOTEUTUS

Tässä osiossa kerrotaan, mitä ominaisuuksia ERP-järjestelmä pitää sisällään, miten järjestelmän tärkeimmät ominaisuudet toteutettiin ja mitä ohjelmointikieliä niiden tekemiseen käytettiin.

6.1 Kirjautuminen

Kirjautumissivu toteutettiin käyttäen HTML-, PHP- ja JavaScript-kieliä. Sivun tyyli toteutettiin CSS-määrittelyillä. JavaScriptillä tehtiin kirjautumisnappiin animoitu ominaisuus. Kun nappia painetaan, napin tilalle tulee pyörivä talo ja teksti ”kirjaututaan...”. Lomakekentissä käytettiin placeholder-ominaisuutta. Kirjautumissivusta tehtiin index.php -niminen, joten tämä sivu avautuu ensimmäisenä.

Kirjautumisruudun (kuva 10) ulkoreuna tehtiin käyttämällä CSS:n reunanpyöritys- ja laatikonvarjotyylejä. Tekstikenttiin käytettiin samoja tyylejä, mutta lisättiin outline-tyyli ja värit. ”Kirjaudu sisään” -napissa käytettiin samoja tyylejä. Taustaväri saatiin käyttämällä gradient-väritystä. Tyylien tekemisessä piti ottaa huomioon, että eri internet-selaimia varten piti käyttää eri tyylimääritelmiä, mutta lopputulos näytti kuitenkin samalta kaikissa yleisimmissä internet-selaimissa.



KARTANOKOTI OY

Käyttäjätunnus

Salasana

Kirjaudu sisään

© 2014 Kartanokoti Oy, Rauhankatu 6, 20100 Turku | 0400-511 487 | info@kartanokoti.fi

Kuva 10. Kirjautumisruutu.

Kirjautumisruudusta tiedot lähetetään eteenpäin PHP-sivulle, mikä käsittelee ja tarkistaa tietojen oikeinpitävyyden. PHP-sivu ottaa vastaan käyttäjätunnuksen ja salasanan. Salasanasta tehdään md5-tekniikalla tiiviste, jota verrataan tietokantaan. Jos tietokannasta löytyy käyttäjä, jonka käyttäjänimi ja tiiviste ovat samat kuin kirjautumista yrittäneellä, niin järjestelmä päästää tämän eteenpäin. Ennen tietokannasta hakua lomakkeesta tulleista tiedoista poistetaan turvallisuussyistä MySQL- ja HTML-komennot.

Jos oikea käyttäjä on löytynyt, niin kirjautumista yrittänyt ohjataan omalle sivulleen. Käyttäjälle luodaan cookie, joka pitää sisällään kirjautumisvarmenteen. Varmenteesta tiedetään, kuka käyttäjä on kirjautunut sisälle. Järjestelmässä käyttäjän rooli ratkaisee, mihin osioihin hänellä on oikeus. Administrator-tason käyttäjällä on oikeus tieteenkin kaikkeen. Asiakaspuolen käyttäjänoikeudet riittävät vain omalle asiakastilille. Jos tietokannasta ei löytynyt oikeaa käyttäjää, niin PHP-sivu ohjaa tämän takaisin kirjautumissivulle, ja sivulle tulee teksti ”Väärä käyttäjänimi tai salasana!”.

6.2 Pääkäyttäjän näkymä

Kun pääkäyttäjänoikeudet omaava henkilö kirjautuu järjestelmään, niin hänet ohjataan pääkäyttäjän sivulle. Tältä sivulta pääsee näkemään ja hallitsemaan yrityksen tehtäviä, projekteja, hankintasopimuksia ja yhteistyökumppaneita.

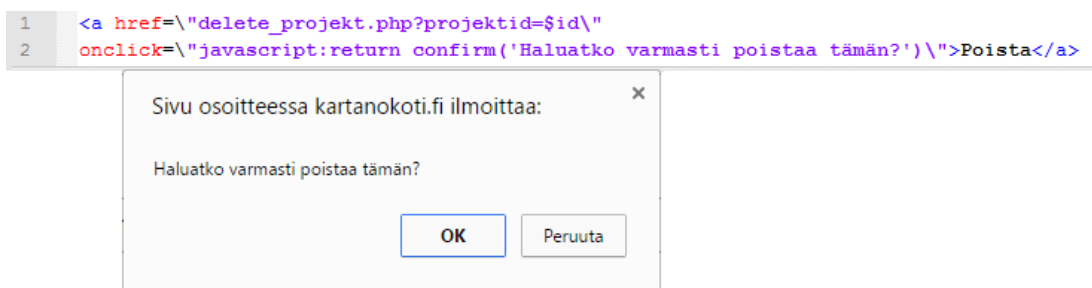
6.2.1 Yrityksen tehtävät

Pääkäyttäjälle avautuu ensimmäiseksi ”Yrityksen tehtävät” -osio. Tästä pääkäyttäjä näkee yhdellä silmäyksellä taloprojekteihin liittyvät tehtävät, kuten onko projekti myöhässä, mitä tehtäviä on tekemättä ja mitä tehtäviä pitäisi tehdä lähiaikoina. Jos projekti on myöhässä, tehtäväkentän väri muuttuu punaiseksi. Sivun PHP-koodi hakee tietokannasta kaikki projektit ja käy läpi projektien tehtävät. Tarkistamisessa otettiin huomioon viikko, päivä, vuosi ja se, onko tehtävä kuitattu jo tehdyksi. Tarkistamisen jälkeen tehtävät listataan allekkain sivulle. Listasta näkee asiakkaan nimen, tehtävän ja tehtävän suoritusviikon. Asiakkaan nimeä klikkaamalla siirrytään kyseisen asiakkaan projektiaikatauluun tämän tehtävän kohdalla.

6.2.2 Projektit

Projektit-osiossa pääkäyttäjä pääsee näkemään jo luotuja projekteja. Projektit haetaan tietokannasta PHP-koodin avulla ja listataan näkyville järjestyksessä uusimmasta van-

himpaan. Listasta näkee asiakkaan nimen, rakennuksentyypin ja projekti-id:n. Asiakkaan nimeä klikkaamalla siirrytään kyseisen asiakkaan projektiin. Projekti on mahdollista poistaa ja arkistoida. Poistamisen vahvistaminen toteutettiin JavaScript-ilmoituksella (kuva 11). OK-nappia painamalla siirrytään delete_projekt.php -sivulle, jossa projekti poistetaan tietokannasta. Projektin poistaminen poistaa samalla kaiken muunkin, mikä liittyy samaan projektiin, esimerkiksi kuvat. Peruuta-nappia painamalla pyydytään samalla sivulla. Delete-projekt -sivulta ohjataan automaattisesti takaisin projektit-sivulle. Käyttäjä ei huomaa, että on käyty toisella sivulla poistamassa projekti.



Kuva 11. JavaScript-ilmoitus.

6.2.3 Hankintasopimukset

Hankintasopimukset-osiossa pääkäyttäjä pääsee näkemään ja hallitsemaan jo tehtyjä talokauppojen hankintasopimuksia. Hankintasopimus luodaan samalla, kun talokaupat sovitaan. Osiossa näkyy hankintasopimuksen tehneen asiakkaan nimi ja sopimuksen tekoajankohta. Hankintasopimuksen pystyy poistamaan samalla tavalla kuin edellä esitellyn projektin poistaminen. Uuden hankintasopimuksen pystyy myös tekemään tästä osiosta. Hankintasopimusta tehdessä kirjataan lomakkeeseen asiakkaan tiedot, tulevan rakennuksen tietoja, toimitusaika jne. Lomake on valmiiksi virallisen hankintasopimuksen näköinen.

Kun hankintasopimuslomake on täytetty, se tallennetaan. Hankintasopimuksen voi myös tulostaa tulosta-nappia painamalla. Tulosta-napin toiminto tehtiin JavaScriptiä hyödyntämällä. Napin painaminen kutsuu printPage() -funktiota. Funktio piilottaa hankintasopimuksesta kaikki napit ja kutsuu internetiselaimen window.print() -funktiota, mikä avaa alla olevan kuvan mukaisen tulostusikkunan.

The screenshot shows a web application interface for printing a document. On the left, there is a sidebar with the following elements:

- Tulosta** (Print): A section with a summary 'Yhteensä: 1 paperiarkki' and buttons for 'Tulosta' and 'Peruuta'.
- Kohde** (Destination): A dropdown menu set to 'PDFCreator' with a 'Vaihda...' button.
- Sivut** (Pages): Radio buttons for 'Kaikki' (selected) and 'esim. 1-5, 8, 11-13'.
- Kopiot** (Copies): A numeric input field set to '1' with '+' and '-' buttons.
- Ulkoasu** (Appearance): A dropdown menu set to 'Pystysuunta'.
- Väri** (Color): A dropdown menu set to 'Väri'.
- A link: '+ Lisää asetuksia'.
- Footer text: 'Tulosta järjestelmän tulostusikonalla... (Ctrl+Shift+P)'.

The main area displays a document titled 'KARTANOKOTI OY' and 'HANKINTASOPIMUS'. The document contains a table with columns for 'Tilaa 1', 'Tilaa 2', and 'Tilaa 3', and a section for 'Kappalemäärä (ml, ALV:0%)'. The table has rows for 'Tilaa 1', 'Tilaa 2', and 'Tilaa 3'. The 'Kappalemäärä' section includes a list of items with their quantities and prices.

Kuva 12. Hankintasopimuksen tulostusnäkymä.

Kun hankintasopimus tallennetaan, lisätään samalla uusi talokauppa projektitietokantaan ja projektille luodaan myös muut tarvittavat tietokannan taulut. Projektille arvotaan numerosarja projektitunnukseksi. Tarkistetaan myös, ettei se ole jo käytössä. Asiakas saa omat tunnukset järjestelmään.

Tallennuksen yhteydessä PHP-funktio arpoo satunnaisen salasanan ja käyttäjätunnukseksi toimii asiakkaan sähköposti. Arpominen tehdään käyttäen substr-, md5- ja rand-funktioita (kuva 13). Md5 tekee kryptatun merkkijonon rand-funktion arvosta ja substr lyhentää merkkijonon sopivan mittaiseksi. Merkkijonosta saadaan numero- ja kirjainyhdistelmä. Salasana tulostetaan väliaikaisesti selväkielisenä näytölle, ja tästä se voidaan tulostaa asiakkaalle. Salasanan voi vaihtaa myöhemmin.

```

3  $password = substr(md5(rand()),0,8);
4  print("Esimerkki: $password");

```

Esimerkki: 3d1abd97

Kuva 13. Salasanan arpominen.

6.3 Projektinäköymä

Kun asiakkaan nimeä on klikattu pääkäyttäjän näkymän projektit-osiossa, siirrytään kyseisen asiakkaan projektiin (kuva 14). Asiakkaan projektin etusivulta näkee suoraan asiakkaan yhteystiedot, kaupan ehdot, suunnitelmat, kuvat ja projekti aikataulun. Asiakkaan yhteystietojen päivittäminen onnistuu suoraan tästä näkymästä. Ruksaamalla ”Päivitä yhteystiedot tallennettaessa” -valintaruudun, PHP-skripti hakee yhteystiedot sivulta ja päivittää ne tietokantaan.

Kaupanehdot haetaan hankintasopimuksesta, eikä niitä tietoja pysty muokkaamaan. Suunnitelmat-osioon tehtiin liikennevalot. Suunnitelmia pystyy tallentamaan tietokantaan +-merkkiä klikkaamalla. Merkin klikkaaminen tuo näytille tiedoston latausmahdollisuuden. Toiminnollisuus tehtiin JavaScriptillä. Valot syttyvät ja vaihtuvat, kun PHP-koodiin tehty tarkistus-skripti huomaa, että oikea liitetiedosto on tallennettu tietokantaan tai tehtävä on merkitty tehdyksi projekti aikataulussa. Tarkistaminen tehdään aina kun sivu ladataan tai päivitetään. Vihreä valo tarkoittaa, että suunnitelmat ovat valmiit. Keltainen valo tarkoittaa, että suunnitelmat ovat kesken.

Projektin muita osioita ovat hankintasopimus, muutossopimukset, muut liitetiedostot, energiasuunnitelma ja tilausvahvistukset.

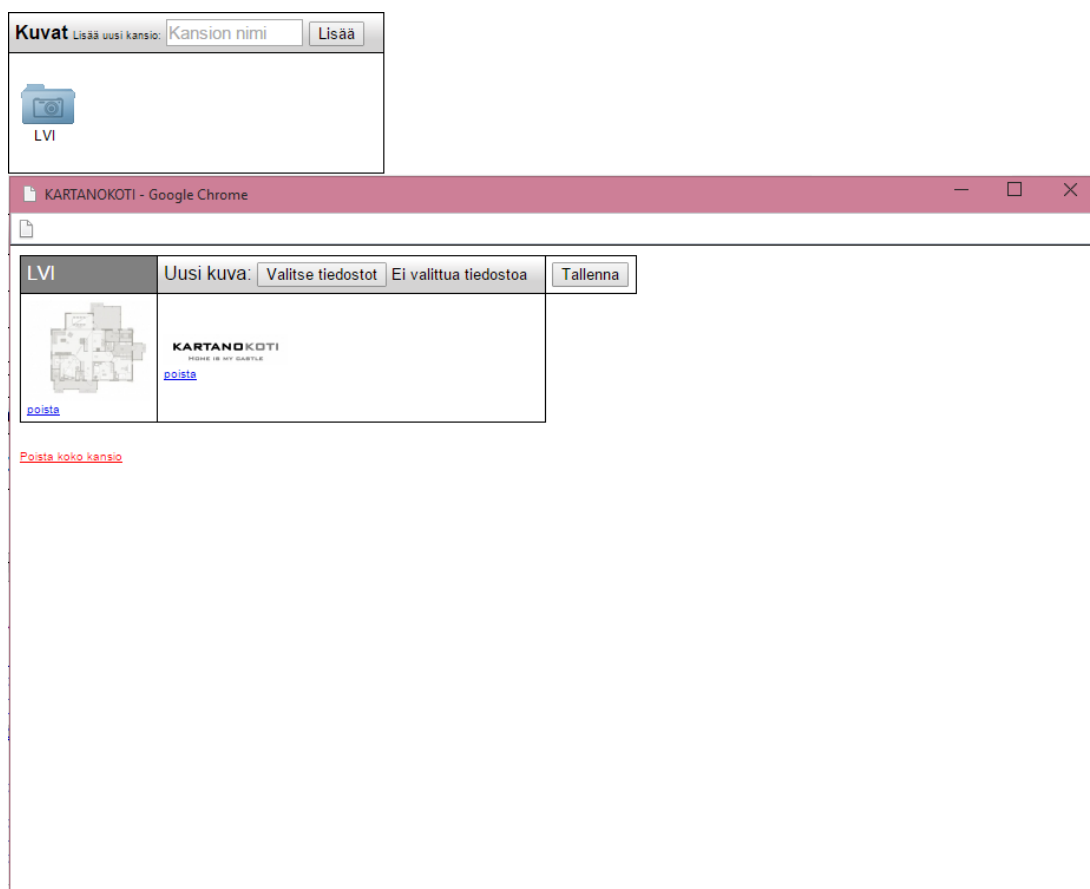
KARTANOKOTI OY

Projekti	Hankintasopimus	Muutossopimukset	Muut Liitetiedostot	Energiasuunnitelma	Tilausvahvistukset			
Yhteystiedot <input type="checkbox"/> Päivitä yhteystiedot tallennettaessa						Suunnitelmat <input type="checkbox"/> Päivitä tallennettaessa		
Asiakas	Eetu Esimerkki			Rakennuslupakuvat	<input type="radio"/>	<input checked="" type="radio"/> +		
Rakennuspaikan osoite:	Esimerkintie 7, 28100 Pori				<input type="radio"/>	+		
Asiakkaan osoite:	Eetuntie 22, 28100 Pori				<input type="radio"/>	+		
Puhelin:	0441232011			(rak. ulkopuoli)	<input type="radio"/>	+		
Sähköposti:	esimerkki.eetu@gmail.com				<input type="radio"/>	+		
Vastaavan työn johtaja:	Matias Määttä				<input type="radio"/>	+		
Puhelin:	0123456789 <input type="button" value="Tallenna"/>				<input type="radio"/>	+		
Sähköposti:	example@example.com				<input type="radio"/>	+		
Kaupanehdot						Kuvat <input type="button" value="Lisää uusi kansio"/> <input type="text" value="Kansion nimi"/> <input type="button" value="Lisää"/>		
Rakennuslupa:	<input type="checkbox"/> On <input checked="" type="checkbox"/> Ei							
Rahoitusehto:	<input checked="" type="checkbox"/> On <input type="checkbox"/> Ei							
Muu ehto:	-							
	Kartanokoti Oy:n tehtävät			Päivä	Viikko <input type="checkbox"/>	Vuosi	Tilaaan/yhteistyökumppanin tehtävät	Päivä
1	Hankintasopimus kirjoitettu			23.06.2015	26	2015		23.06.2015

Kuva 14. Projektinäköymä.

6.3.1 Kuvat

Jokaisen asiakkaan projektiin on mahdollista lisätä kuvia. Kuvia pystyy lisäämään projektinäkömään etusivulta. Ensiksi lisätään kansio ja nimetään se kuvia varten. Kansiota klikkaamalla avautuu uusi pieni ikkuna, josta kuvia voi lisätä (kuva 15). Uuden ikkunan avautuminen tehtiin JavaScriptillä käyttäen `makeExternalPopup` -funktioita. Linkissä kerrotaan PHP-sivulle, minkä projektin kuvakansio avataan. Ikkunasta voi lisätä ja poistaa kuvia. Koko kansion poistaminen onnistuu myös. Kuvia voi lisätä useampia yhdellä kertaa, kuitenkin maksimissaan viisi kuvaa kerralla.



Kuva 15. Kuvat-ikkuna.

Kuvien tallennuksen yhteydessä kuvat pienennetään automaattisesti PHP:n avulla pikselikokoon 1024 x 678. Kuvista tehdään myös pienennys esikatselukuvaa varten. Kuvan kooksi tulee 105 x 80 px. Jpeg- ja png-kuvamuodot ovat sallittuja. Projektiin lisätyt kuvat tulevat automaattisesti näkymään myös asiakkaan näkymässä. Asiakkaan näkymässä on kuvia varten oma galleria. Alkuperäisen kuvan leveys ja korkeus haetaan

käyttäen `getimagesize`-funktiota. Tämän jälkeen lasketaan uusi korkeus hyödyntäen saatuja tietoja. Kuvan tiedostomuoto tutkitaan ja ladataan `source`-nimiseen muuttujaan käyttäen joko `imagecreatefromjpeg`- tai `imagecreatefrompng`-funktiota. Lopuksi kuvan koko muutetaan `imagecopyresampled`-funktiolla ja tallennetaan palvelimelle. Alla olevasta kuvasta näkee miten kuvan pienentäminen tapahtuu PHP:llä.

```

4 // Haetaan alkuperäisen kuvan koko
5 list($width, $height) = getimagesize($file_name);
6 $newwidth = 1024;
7 $newheight = 678;
8 $newheight=(($height/$width)*$newwidth);
9
10 if($width > $newwidth) {
11     $thumb = imagecreatetruecolor($newwidth, $newheight);
12     if($filetype == "image/jpeg") {
13         $source = imagecreatefromjpeg($file_name);
14     }
15     else if($filetype == "image/png") {
16         $source = imagecreatefrompng($file_name);
17     }
18
19     // Koon muuttaminen
20     imagecopyresampled($thumb, $source, 0, 0, 0, 0, $newwidth, $newheight, $width, $height);
21     // Tallennetaan pienennetty kuva
22     imagejpeg($thumb, "/$file_name", 100);
23 }

```

Kuvat 16. Kuvan koon pienentäminen.

6.3.2 Projekti aikataulu

Jokaiselle taloprojektille luodaan automaattisesti oma projekti aikataulu (kuva 18). Projekti aikataulussa on jaoteltuna yrityksen eli Kartanokodin omat tehtävät ja tilaajan tai yhteistyökumppanin tehtävät. Projekti aikataulusta pääsee näkemään, minä viikona ja vuotena tehtävä pitäisi tehdä tai on tehty. Tehty tehtävä merkataan hiirellä kaksoisklikkaamalla päivä-kenttää, jolloin se muuttuu vihreäksi ja kenttään ilmestyy nykyinen päivämäärä. Tämä toiminnollisuus tehtiin JavaScriptillä. Kentälle määritettiin `ondblclick`- ja `onchange`-tapahtumat.

Kaksoisklikattaessa kenttää `ondblclick` huomaa tapahtuman ja kutsuu `fchangeclick`-funktiota (kuva 17). Funktiolle lähetetään parametrejä eli tietoja. Tiedot ovat kentän nimi, id ja sijainti. Parametrien avulla funktio tietää, mitä kenttää kaksoisklikattiin. `Fchangeclick`- funktio hakee nykyisen päivämäärän `date`-funktioon ja kirjoittaa sen `document.getElementById`-metodilla klikatun kentän arvoksi. Samalla kentän taustaväri vaihdetaan `document.getElementById.style.backgroundColor`-metodilla vihreäksi ja klikkauksen sijainti tallennetaan `location`-muuttujaan, jotta tallennettaessa sivu osaa

palata vanhaan sijaintiin. Onchange-tapahtuma huomaa kentän arvon muuttumisen ja kutsuu tällöin fchangeclolor-funktiota. Tapahtuma luotiin kentän arvon manuaalista muokkaamista varten. Kaksoisklikkaus asettaa kenttään aina nykyisen päivämäärän. Fchangeclolor-funktiolle lähetetään parametreina kentän nimi ja id. Kentän väri muutetaan hyödyntäen document.getElementById-metodia.

```

1  function fchangeclolor(nimi,id,location) {
2      date = new Date;
3      year = date.getFullYear();
4      month = date.getMonth();
5      month = parseInt(month, 10) + 1;
6      d = date.getDate();
7
8      if (id == 0) {
9          document.getElementById("vari"+nimi).style.backgroundColor='#7EFF29';
10         var subEmail = document.getElementById("input"+nimi);
11         subEmail.style.backgroundColor = '#7EFF29';
12         subEmail.style.borderStyle = 'none';
13         document.getElementById("input"+nimi).value =d+"."+month+"."+year;
14     }
15     else if (id == 2) {
16         document.getElementById("varit"+nimi).style.backgroundColor='#7EFF29';
17         var subEmail = document.getElementById("input"+nimi);
18         subEmail.style.backgroundColor = '#7EFF29';
19         subEmail.style.borderStyle = 'none';
20         document.getElementById("input"+nimi).value =d+"."+month+"."+year;
21     }
22     //Tallennetaan viimeisen klikkauksen sijainti
23     document.getElementById("location").value = location;
24 }
25

```

Kuva 17. Fchangeclolor-funktio.

Tallenna-nappia painaessa kuitattujen kenttien päivämäärät tallentuvat tietokantaan. Projektiakataulun tehtävät on kirjoitettu PHP-taulukkoon, josta niitä on helppo käsitellä. Projektiakataulun tehtäväviikot ajoittuvat automaattisesti projektia luotaessa. Projektiviikkoihin vaikuttaa taloprojektin suunniteltu toimitusajankohta. Sivua ladattaessa tai päivittäessä PHP-skripti käy projektin projektiakataulun läpi ja merkkää tehdyt tehtävät vihreänä ja lataa tehtävän päiväkenttiin oikeat päivämäärät tietokannasta. Projektiakatauluun tehtiin nappi, jota painamalla saa tarvittaessa projektiakataulun automaattisen aikatauluttamisen pois päältä.

Kartanokoti Oy:n tehtävät	Päivä	Viikko □	Vuosi	Tilaaajan/yhteistyökumppanin tehtävät	Päivä
1 Hankintasopimus kirjoitettu	23.06.2015	26	2015		23.06.2015
2	30.6.2015	27	2015	Asiakas järjestää rakennushankkeelle pääsuunnittelijan, vastaavan työnohtajan ja kvy-vastaavan ja tilaa mahdollisen maaperätutkimuksen.	30.6.2015
3	30.6.2015	27	2015		30.6.2015
4	30.6.2015	28	2015		30.6.2015
5		29	2015	Pääsuunnittelija/maaperätutkija antaa perustamislauseannon.	
6		29	2015	Pääsuunnittelija aloittaa asemapiirustuksen piirtämisen	

Kuva 18. Projekti aikataulu.

Selatessa projekti aikataulua alaspäin sivun tallenna-nappi seuraa mukana. Näin muutokset on helppo tallentaa, eikä käyttäjän tarvitse kelata sivua alkuun. Tallenna-napin liikkuvuus toteutettiin CSS-tyylimäärittelyllä. Sivun tyylitiedostoon määritettiin napille oma luokka, ja luokan tyyleiksi position:fixed ja top:55%. Position:fixed-määrittely tarkoittaa että nappi pysyy paikallaan, vaikka sivua vieritetään. Nappi niin sanotusti kelluu elementtien päällä. Top: 55%- määrittelyllä kerrotaan selaimelle napin sijainti: selaimen yläreunasta 55 % alaspäin eli vähän yli puolivälin näytöllä.

Projekti aikataulun voi tulostaa painamalla tulosta-nappia. Napin painallus avaa erillisen tulosta.php-sivun, johon projekti aikataulu on aseteltu sopivaksi A4-sivua varten. Sivun latautuessa avautuu automaattisesti tulosta-ikkuna JavaScript-funktion (esitelty aiemmin) avulla. JavaScript tapahtumat saadaan toimimaan sijoittamalla ne ”window.onload = function()” -funktioon.

6.3.3 Muut osiot

Projektin muita osioita ovat hankintasopimus, muutossopimukset, muut liitetiedostot, energiasuunnitelma ja tilausvahvistukset. Hankintasopimus-osiossa tallennetaan asiakkaan hankintasopimukseen liittyviä liitteitä ja päästään tarkastelemaan tehtyä hankintasopimusta. Lisättävät liitteet on valmiiksi määritetty. Hankintasopimus-osion liitteet näkyvät myös asiakkaan näkymässä. Muutossopimukset-osioista tehtiin teknisesti vaikeampi. Sivulta näkyy, mikä muutossopimus on lisätty ja onko sitä palautettu allekirjoitettuna. Muutossopimuksia voi myös poistaa. Liitteen palauttamisen asiakas pystyy tekemään omilla tunnuksillaan asiakas-näkymässä.

Muutossopimukset sivulla pystyy tekemään uuden muutossopimuksen painamalla ”Lisää uusi muutossopimus” -linkkiä, jolloin avautuu uutta sopimusta varten tehty lomake. Lomakkeessa kirjoitetaan sopimukselle aihe, otsikko, muutoskohteen tiedot, hinta ja päivämäärä. Asiakkaan tiedot haetaan automaattisesti tietokannasta. Muutoskohteita voi samaan sopimukseen lisätä useampia. Näkyvillä on valmiina yksi muutoskohde, mutta nappia painamalla niitä saa lisää. Toiminto toteutettiin JavaScriptillä. Napin painaminen kutsuu JavaScriptin `fshow()` -funktiota (kuva 19), johon on määritetty tapahtumiksi uuden muutoskohde-laatikon näyttäminen sivulle ja samalla kasvatetaan sivun `kohteetlkm`-muuttujan arvoa. PHP- sivu hakee `kohteetlkm`-arvon, jotta skripti hakee myös useampien kuin vain yhden muutoskohteen tiedot.

JavaScriptissä uusi muutoskohde saadaan näkyville `document.getElementById`-metodilla. Metodiin asetetaan lisäksi `style.display="inline"`, jolla määrätyn `id`:n tyyli vaihdetaan näkyväksi. Lisättävät muutoskohteet on valmiiksi kirjoitettu html-koodiin, mutta ne on piilotettu. Jokaisella muutoskohteella on oma hinta-kenttä. Hinta-kentän arvon vaihtaminen kutsuu `fchange()`-funktiota (kuva 20), joka laskee muutoskohteiden hinnan automaattisesti yhteen ja lisää sen yhteishinta-kenttään. Hintojen laskeminen ja haku tapahtuu `for`-silmukan avulla.

```

1 function fshow() {
2     nro++;
3     document.getElementById("kohde"+nro).style.display="inline";
4     document.getElementById("kohteetlkm").value=nro;
5 }

```

Kuva 19. Fshow()-funktio.

```

1 function fchange() {
2     summa=0;
3     for(i=1; i<=nro; i++) {
4         temp = document.getElementById("hint"+i).value;
5         temp = parseInt(temp) || 0;
6         summa = summa+temp;
7     }
8     summa = parseInt(summa) || 0;
9     document.getElementById("tietohinta").value=summa+" €";
10 }

```

Kuva 20. Fchange()-funktio.

Muutossopimus tallennetaan pdf-liitteeksi (kuva 21). Html-sivun kääntäminen pdf-muotoon toteutettiin vapaan lähdekoodin TCPDF PHP-luokalla. Kääntämisen jälkeen pdf-liite tallennetaan palvelimelle ja tietokantaan merkitään liitteen sijainti. TCPDF tukee UTF-8-, Unicode-, RTL-kieliä, XHTML:ää, JavaScript:iä, digitaalisia allekirjoituksia ja viivakoodeja. TCPDF-luokan saa ladattua käyttöön TCPDF:n kotisivuilta osoitteesta www.tcpdf.org. Sivustolla on paljon hyviä esimerkkejä, joita kokeilemalla pääsee nopeasti perille, kuinka luokka toimii.

KARTANOKOTI OY		Muutossopimus Testi
Asiakas:	Eetu Esimerkki	
Puhelin:	0441232011	
Sähköposti:	esimerkki.eetu@gmail.com	
Myyjä:	Matias Määttä	
Puhelin:	044123456789	
Sähköposti:	example@example.com	
Muutoskohde 1		Hinta: 500 €
<hr/>		
Kuvaus:	Takka vaihdetaan toiseen.	
Muutoskohde 2		Hinta: 50 €
<hr/>		
Kuvaus:	Takan väri vaihdettu.	
Muutokset yhteensä (sis. alv:n 24 %): 550 €		
Hyväksyn muutossopimuksen.		
<u>Paikka ja päivämäärä:</u>	<hr/>	
<u>Asiakkaan allekirjoitus:</u>	<hr/>	
<u>Myyjän allekirjoitus: Matias Määttä</u>	<hr/>	
Kartanokoti Oy		

Kuva 21. Valmis pdf-liite.

Muut liitetiedostot- ja energiasuunnitelmat-osiossa voi liittää liitetiedostoja. Muut liitetiedostot-osion liitteet ovat asiakkaan lisäämiä. Asiakas lisää liitteet omassa näkymässään ja sieltä ne myös näkyvät projektinäkymässä. Tilausvahvistukset-osiossa näkyvät kaikki taloprojektin tilausvahvistukset liitteineen. Jokaisen tilausvahvistuksen kohdalta voi valita, näytetäänkö tilausvahvistus asiakkaalle hänen näkymässään. Oletuksena on, että tilausvahvistukset eivät näy ja näytettävät vahvistukset määritetään projektikohtaisesti.

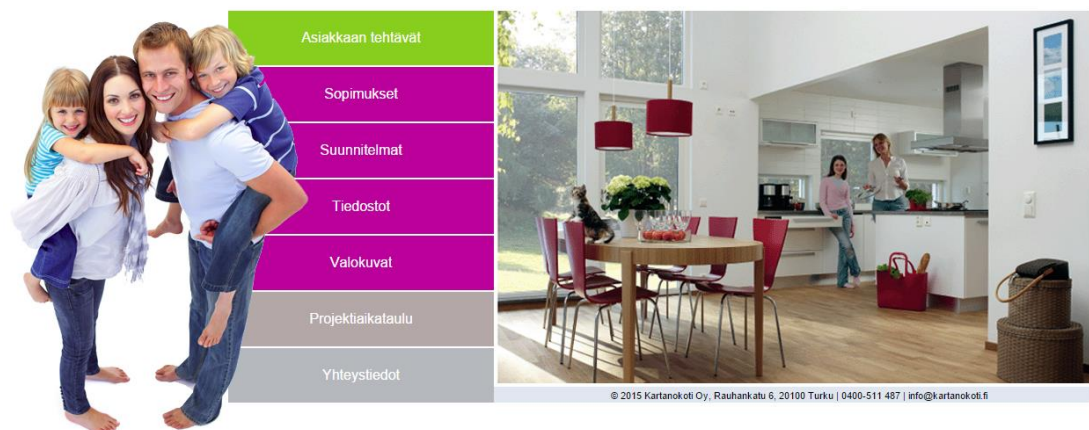
6.4 Asiakkaan näkymä

Kirjautumisen yhteydessä tutkitaan kirjautujan oikeudet ja hänet siirretään oikealle sivulle. Kun asiakas kirjautuu, siirrytään asiakkaan omaan projektinäkymään. Kutsun tätä asiakkaan näkymäksi (kuva 22). Asiakkaan näkymässä ovat seuraavat osiot: asiakkaan tehtävät, sopimukset, suunnitelmat, tiedostot, valokuvat, projekti aikataulu ja yhteystiedot. Ulkoasu ja väritys suunniteltiin yhdessä. Ulkoasu toteutettiin HTML:llä ja CSS-tyylimäärittelyillä. Etusivun oikealla puolella oleva isokuva vaihtuu aina, kun hiiren kursori viedään eri valikon päälle. Tämä toteutettiin JavaScriptin onmouseover- ja onmouseout-tapahtumilla.

Onmouseover-tapahtuma tunnistaa, jos hiiren kursori on alueen päällä ja kutsuu `mOver()`-funktioita. Samaten onmouseout-tapahtuma tunnistaa, jos hiiren kursori viedään pois alueen päältä, ja tällöin se kutsuu `mOut()`-funktioita. Funktiot asettavat uuden taustakuvan `document.getElementById("tausta").src`-metodilla. `Mover()`-funktioille kerrotaan parametrina, mikä valikko on kyseessä.

KARTANOKOTI
HOME IS MY CASTLE

Eetu Esimerkki Kirjautu ulos



Kuva 22. Asiakkaan näkymä.

Asiakkaan-tehtävät osiossa on asiakasta varten tehtäviä, jotka suorittamalla taloprojekti valmistuu aikataulussaan. Tehtävät ovat sellaisia, joita Kartanokoti ei pysty tekemään. Asiakkaan pitää esimerkiksi hakea rakennuslupa rakennusvalvonnalta. Tehtävälaatikossa näkyy tehtävän numero, viikko, vuosi, tehtävä ja suoritettu merkki. Merk-

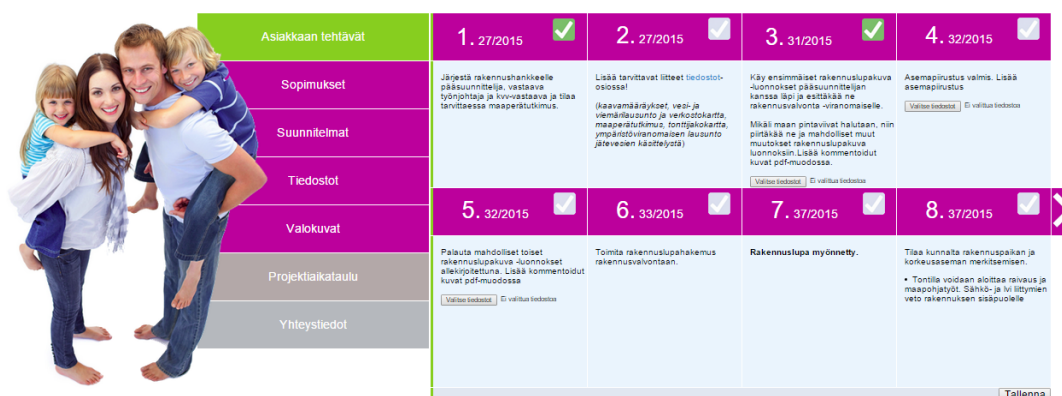
kiä klikkaamalla tehtävän suoritettu-merkki muuttuu harmaasta vihreäksi. Osassa tehtävissä on mahdollista tallentaa liite. Tällöin tehtävää ei voi asettaa tehdyksi, ennen kuin liite on asetettu.

Tallennettaessa suoritettut tehtävät tallentuvat tietokantaan. Kentän arvoksi tallennetaan nykyinen päivämäärä. Päivämäärät tulevat näkyviin pääkäyttäjän näkymään ja asiakkaan projektin projektiakatauluun. Suoritettun tehtävän kenttä muuttuu vihreäksi. Asiakkaan tehtävät osio toteutettiin Html:llä, PHP:llä ja JavaScriptillä (kuva 23). Sivuston ulkoasu määritettiin CSS-tyyleillä. Asiakkaan tehtävät-kenttien sisältö haetaan PHP-taulukosta. Taulukosta on helppo käsitellä tietoja ja sijoittaa ne oikeille paikoilleen. PHP-skripti tutkii projektiakataulun tietokannasta, mikä tehtävä on tehty viikoksi ja automaattisesti avaa kesken jääneen tehtäväsivun.

Tarkistuksen yhteydessä ensimmäiseksi tutkitaan, onko tehtävä myöhässä ja haetaan mahdolliset liitetiedostot. Myöhässä olevan tehtävän tehtäväkentän väri muutetaan punaiseksi. Toiseksi tarkistetaan, onko tehtävä jo tehty. Tehdyn tehtävän checkbox-valintaruutu lukitaan, jolloin sitä ei voi enää merkitä uudelleen tehdyksi. Kolmanneksi tehtäväruuduille haetaan otsikoksi oikea viikko ja vuosi. Samalla haetaan myös tehtäväteksti taulukosta. Tämän jälkeen tiedot tulostetaan näytille alla olevan kuvan mukaisesti HTML-taulukkoon.

KARTANO KOTI
HOME IS MY CASTLE

Eetu Esimerkki Kirjautu ulos



Asiakkaan tehtävät	1. 27/2015	2. 27/2015	3. 31/2015	4. 32/2015
Sopimukset	Järjestä rakennushankkeelle pääsuunnitelja, vastaava työjohtaja ja liivestava ja tilaa tarvittaessa maaperätutkimus.	Lisää tarvittavat liitteet tiedostol- osossa! (kaavamääräykset, veo- ja viemäri-suunnit ja verkostokaarta, maaperätutkimus, fonttigakortti, ympäristöviranomaisen luvuunto järeveerin käsittelystä)	Käy ensimmäiset rakennuslupakuva- luonnokset pääsuunniteljan kanssa läpi ja esittää ne rakennusvalvonta -viranomaiselle.	Asemapiirustus valmis. Lisää asemapiirustus
Suunnitelmat			Mikäli maan pintaviivat halutaan, niin pitääkää ne ja mahdolliset muut muutokset rakennuslupakuva luonnoksiin. Lisää kommentoidut kuvat pdf-muodossa.	<input type="button" value="Valitse tiedosto"/> <input type="button" value="Vaihda tiedosto"/>
Tiedostot				
Valokuvat				
Projektiakataulu	5. 32/2015	6. 33/2015	7. 37/2015	8. 37/2015
Yhteystiedot	Palautu mahdolliset toiset rakennuslupakuva luonnokset allekirjoitettuna. Lisää kommentoidut kuvat pdf-muodossa	Toimita rakennuslupahakemus rakennusvalvontaan.	Rakennuslupa myönnetty.	Tilaa kunnalta rakennuspaikan ja korjauksessaan merkitsemän. • Tontilla voidaan aloittaa rakaus ja maasäätelyt. Säähio- ja iv liittimen veto rakennuksen sisäpuolelle

Kuva 23. Asiakkaan tehtävät.

Sopimukset, suunnitelmat ja tiedostot-osiossa asiakas näkee taloprojektia koskevia liitteitä ja pystyy myös lisäämään niitä itse. Valokuvat-osioon tulee näkyville taloprojektin eri rakennusvaiheista otettuja kuvia. Asiakas pystyy tehtävistä, projekti aikataulusta ja kuvista seuraamaan, miten talonrakennus edistyy. Valokuvat haetaan tietokannasta. Projekti aikataulu-osioista asiakas pääsee seuraamaan tekstimuodossa, missä vaiheessa projektia mennään ja mitä tapahtuu milloinkin (kuva 24). Projekti aikataulussa näkyvät suoritettavat tehtävät vihreällä värillä. Projekti aikataulu-sivussa käytettiin CSS:n `position:fixed`-ominaisuutta, jolloin sivusta saatiin kelluva elementti. Kuvat ja palkit pysyvät paikallaan, vaikka sivua vieritetään alaspäin. Ainoastaan projekti aikataulun tekstikentät liikkuvat.

KARTANOKOTI
HOME IS MY CASTLE

Eetu Esimerkki Kirjautu ulos



	Kartanokoti Oy:n tehtävät	Päivä	Viikko	Vuosi	Tilaajan/yhteistyökumppanin tehtävät	Päivä
1	Hankintasopimus kirjoitettu	23.06.2015	26	2015	Asiakkaalle annetaan Kartanokoti tunnukset.	23.06.2015
2		30.6.2015	27	2015	Asiakas järjestää rakennushankkeelle pääsuunnittelijan, vastaavan työnjohtajan ja kvv-vastaavan ja tilaa mahdollisen maaperätutkimuksen.	30.6.2015
3		30.6.2015	29	2015	Pääsuunnittelija/maaperätutkija antaa perustamislauseannon.	30.6.2015
4		30.6.2015	29	2015	Pääsuunnittelija aloittaa asemapiirustuksen piirtämisen	30.6.2015
5	Rakennuslupakuvaluonnokset 1, lähetetään asiakkaalle. (PDF-muodossa.)	30.6.2015	30	2015		30.6.2015
6		30.6.2015	31	2015	Asiakas käy rakennuslupakuvaluonnokset pääsuunnittelijan kanssa läpi ja esittää ne rakennusvalvonta-viranomaiselle. Mikäli maan pintaviivat halutaan julkisuuksiin, asiakas piirtää ne pääsuunnittelijan kanssa. Mahdolliset muut muutokset piirretään myös kuviin ja palautetaan.	30.6.2015
7	Rakennuslupakuvaluonnokset 2, lähetetään asiakkaalle (PDF-muodossa) nähtäväksi.		32	2015	Asemapiirustus valmis. Asiakas palauttaa asemapiirustuksen	
8			32	2015	Asiakas palauttaa rakennuslupakuvaluonnokset allekirjoitettuna.	

Kuva 24. Asiakkaan projekti aikataulu.

Yhteystiedot-osiossa asiakas näkee omat yhteystiedot (kuva 25). Asiakkaan tehtävänä on tarvittaessa lisätä tai päivittää pääsuunnittelijan, vastaavan työnjohtaja-, kvv-vastaavan yhteystiedot. Yhteystiedot haetaan asiakkaan projektin tietokannasta ja tarvittaessa päivitetään tietokantaan.



Asiakkaan tehtävät	Omat yhteystiedot		Pääsuunnittelijan yhteystiedot	
Sopimukset	Nimi:	Eetu Esimerkki	Nimi:	<input type="text"/>
Suunnitelmat	Rakennuspaikan osoite:	Esimerkintie 7, 22100 Example	Puhelin:	<input type="text"/>
Tiedostot	Osoite:	Merkinesintie 4, 22100 Example	Sähköposti:	<input type="text"/>
Valokuvat	Puhelin:	0441232011	KVV- vastaavan yhteystiedot	
Projektiakataulu	Sähköposti:	esimerkki.eetu@gmail.com	Nimi:	<input type="text"/>
Yhteystiedot	Vastaavan työnjohtajan yhteystiedot		Puhelin:	<input type="text"/>
	Nimi:	<input type="text"/>	Sähköposti:	<input type="text"/>
	Puhelin:	<input type="text"/>	<input type="button" value="Tallenna"/>	
	Sähköposti:	<input type="text"/>		

© 2014 Kartanokoti Oy, Rauhankatu 6, 20100 Turku | 0400-511 487 | info@kartanokoti.fi

Kuva 25. Asiakasnäkymän yhteystiedot-osio.

7 TIETOTURVA

7.1 Merkkijonojen tarkistaminen

Lomakkeelta tulevat tiedot pitää tarkistaa, ennen kuin niitä tallennetaan tietokantaan tai käytetään tietokantakyselyissä. Lomakkeelta tulevista tiedoista muutettiin mahdolliset erikoismerkit html-muotoon käyttäen PHP:n htmlspecialchars- ja MySQL:n mysql_real_escape_string- funktioita. Alla olevassa kuvassa on esimerkki tietojen tarkistamisesta.

```

1 <?php
2     echo htmlspecialchars("<b><u>Esimerkki</u></b>");
3 >?

```

<u>Esimerkki</u>

```

1 <?php
2     $tulostus = " 'Esimerkki ' OR '' = 'Esimerkki'";
3     $tulostus = mysql_real_escape_string($tulostus);
4     echo $tulostus;
5 >?

```

\'Esimerkki \' OR \'\' = \'Esimerkki\'

Kuva 26. Esimerkki ja lopputulos erikoismerkkien muuttamisesta.

7.2 Salasanojen suojaaminen

Salasanoja ei voi tietoturvasyistä tallentaa selväkielisenä tietokantaan, vaan ne pitää kryptata. Salasanojen kryptaamiseen käytettiin PHP:n md5-funktiota (kuva 27). Md5 luo salasanaa 128-bittisen tiivisteen, joka on yleensä 32 merkkiä pitkä. Ennen tiivisteen tekemistä salasanaan lisättiin niin sanottu suola eli merkkijono pidentämään annettua salasanaa. Kirjautumisen yhteydessä annettua salasanaa verrataan tietokannassa olevaan tiivisteeseen.

```
1 <?php
2     $salasana = "salasana";
3     $salasana = md5($salasana . "suola");
4     echo "Alkuperäinen salasana: salasana
5         \n Md5-tiiviste: $salasana";
6 ?>
```

```
Alkuperäinen salasana: salasana
Md5-tiiviste: 21ac94be0a6e7f4b7af2c1c744cc7734
```

Kuva 27. Salasanan suojaaminen.

7.3 Tiedostojen tallentaminen

Tiedostojen tallentamisessa pitää huomioida palvelimelle tallennettavan tiedoston tiedostomuoto. Pitää estää tiedostomuodot, joita pystyisi palvelimella suorittamaan tai sallia vain ne tiedostomuodot, joita palvelimelle halutaan tallennettavan. Ehdottomasti täytyy estää PHP-tiedostojen lataaminen palvelimelle. Kun tiedosto lähetetään selaimen kautta PHP-sivulle, tiedostomuodon tarkistaminen tapahtuu PHP:n \$_FILES type-funktiolla. Tiedostomuotoja vertailtiin if-lauseella ja sallittiin vain pdf-, jpg-, jpeg-, png- ja zip-tiedostomuodot.

Tiedostomuodoista täytyi etsiä oikeat viralliset nimet. Esimerkiksi pdf-tiedostojen tiedostopääte on application/pdf -niminen. Toinen turvallisuuteen vaikuttava seikka on liitteiden alkuperäisten nimien vaihtaminen, samalla pitkät tiedostonimet lyhennettiin automaattisesti. PHP:n explode-funktiolla merkkijonon saa katkaistua halutusta kohdasta. Tässä tapauksessa merkkijono katkaistiin pisteen kohdalta eli juuri ennen tiedostopäätettä. Tämän jälkeen merkkijono vaihdettiin ja tarvittaessa lyhennettiin.

8 TIETOKANNAN TAULUT JA RAKENNE

Toiminnanohjausjärjestelmä pitää sisällään neljä tietokantaa. Yksi tietokannoista on varattu taloprojektien kuvia varten, toinen projekteja varten, kolmas projektiaikatauluja varten ja neljäs tietokanta käyttäjätietoja varten.

Taulukko1. Kuvat-tietokannan taulun rakenne.

Sarake	Tyyppi
image	varchar(40)
image_small	varchar(40)
album	varchar(150)
id	int(11)

Image- ja image_small -sarake on kuvatiedoston nimiä eli kuvapolkuja varten. Image_small -sarakeeseen tulee pienennetyn kuvatiedoston nimi. Album-sarakeeseen tulee kansion nimi, missä kuvakansiossa kuva sijaitsee. Varchar-tyyppi tarkoittaa merkkijonoa ja sulkeissa oleva numero merkkijonon pituutta. Int-tyyppi tarkoittaa kokonaislukua. Projekti-tietokanta pitää sisällään hankintasopimukset-, projektit- ja liitteet-taulut.

Taulukko 2. Hankintasopimukset-taulun rakenne.

Sarake	Tyyppi	Sarake	Tyyppi
tilaaja1	varchar(50)	rahoitusehto	varchar(10)
hlotunnus1	varchar(10)	muuehtomuu	varchar(50)
sposti1	varchar(30)	muuehtool	varchar(5)
puhnro1	varchar(20)	kauppa-summa	varchar(20)
tilaaja2	varchar(50)	toimitusaika	varchar(20)
hlotunnus2	varchar(10)	rakennusaika	varchar(20)
sposti2	varchar(30)	rakennusmuu	varchar(50)
puhnro2	varchar(20)	maksuehdot	varchar(500)
laskutusosoite	varchar(50)	lisatiedot	varchar(200)
toimitusosoite	varchar(50)	liite1	int(3)
rakennuspaikka-kunta	varchar(30)	liite2	int(3)
kosa	varchar(30)	liite3	int(3)

kortteli	varchar(30)	liite4	int(3)
tontti	varchar(30)	paikkajaaika	varchar(50)
malli	varchar(50)	nimenselven- nys1	varchar(30)
rakennus	varchar(20)	nimenselven- nys2	varchar(30)
toimitussisälto	varchar(20)	nimenselven- nys3	varchar(30)
rakennuslupa	varchar(10)	id	int(11)
rakennuslupaehto	varchar(10)	auto	int(10)

Hankintasopimukset-taulu pitää sisällään hankintasopimukseen liittyviä tietoja.

Taulukko 3. Projektit-taulun rakenne.

Sarake	Tyyppi	Sarake	Tyyppi
Asiakas	varchar(50)	puhpaasuu	varchar(50)
Rakennus- paikka	varchar(50)	spostipaasuu	varchar(50)
asiakasosoite	varchar(50)	nimikvv	varchar(50)
Puhelin1	varchar(50)	puhkvv	varchar(50)
Sposti1	varchar(50)	spostikvv	varchar(50)
Tyonjohtaja	varchar(50)	Suunnittelija	int(5)
Puh2	varchar(50)	autoviikot	int(5)
Sposti2	varchar(50)	Id	int(11)
Rakennus	varchar(30)	autoid	int(10)
nimipaasuu	varchar(50)		

Projektit-taulu pitää sisällään projekteihin liittyviä lisätietoja, joita hankintasopimuk-
sessa ei näy.

Taulukko 4. Liitteet-taulun rakenne.

Sarake	Tyyppi	Sarake	Tyyppi
id	int(10)	lampohavioident	varchar(250)
rakennuslupakuvat	varchar(230)	takkamuutos	varchar(250)
perustuksenmittakuvat	varchar(230)	sahkomuutos	varchar(250)
lvisuunnitelmatulk	varchar(230)	lvimuutos	varchar(250)
rakennesuunnitelmat	varchar(230)	sispinnoitemuutos	varchar(250)
perustuksenrakennekuva	varchar(230)	muumuutos	varchar(250)
sahkosuunnitelma	varchar(230)	thtliite9	varchar(250)

lvisuunnitelmatsis	varchar(230)	thtliite10	varchar(250)
kalustesuunnitelma	varchar(230)	thtliite11	varchar(250)
hankintasopimuspdf	varchar(230)	thtliite12	varchar(250)
etusivu	varchar(230)	thtliite61	varchar(250)
talopakettitoimerit	varchar(230)	thtliite62	varchar(250)
raktektyotjalisatied	varchar(230)	thtliite63	varchar(250)
huonetiiaerittely	varchar(230)	thtliite64	varchar(250)
rakennuksenkuvat	varchar(230)	thtliite68	varchar(250)
suunnittelijanrakennuksenkuvat	varchar(250)	thtliite69	varchar(250)
hankintasopimuskeh	varchar(230)	thtliite70	varchar(250)
toimituserittelykeh	varchar(230)	thtliite71	varchar(250)
laskentakeh	varchar(230)	thtliite83	varchar(250)
toimituskuvatkeh	varchar(230)	thtliite84	varchar(250)
kaavamaaraykset	varchar(250)	thtliite119	varchar(250)
vesijaviemlausunto	varchar(250)	thtliite120	varchar(250)
maaperatutkimus	varchar(250)	thtliite125	varchar(250)
tonttijakokartta	varchar(250)	thtliite126	varchar(250)
ympvirovaisenlaus	varchar(250)	thtliite127	varchar(250)
energiatodistus	varchar(250)	raklupakuvaluon1	varchar(230)
energiaselvitys	varchar(250)	raklupakuvaluon2	varchar(230)
energialaskennanl	varchar(250)		

Liitteet-taulu pitää sisällään projektin liitetiedostot.

Taulukko 5. Käyttäjätiedot-tietokannan käyttäjät-taulun rakenne.

Sarake	Tyyppi
id	int(11)
nick	varchar(50)
password	varchar(50)
varmenne	varchar(30)
role	varchar(30)
name	varchar(100)
projekti	int(11)

9 TESTAUS JA YHTEENVETO

Toiminnanohjausjärjestelmää testasivat Kartanokodin henkilökunta ja muutamat asiakkaat. Palautetta saatiin ja järjestelmää korjattiin ja päivitettiin. Tekstimuotoihin ja ymmärrettävyyteen piti kiinnittää enemmän huomiota. Järjestelmän kehitystä jatkettiin kesäloman jälkeenkin. Järjestelmän ylläpito ja kehitys jatkuvat edelleen. Toiminnanohjausjärjestelmän tekeminen oli mielenkiintoinen projekti.

Projektista tuli odotettua laajempi, ja se oli myös haastava. Projektin tekemisen aikana ei voinut välttyä uusilta asioilta ja toiminnoilta. Internetiä tutkimalla löytyi kuitenkin neuvoja, kuinka asioita sai toteutettua, ja ongelmista selvittiin. Projektia tehdessä suunnittelu on tärkeä asia, tämän opin hyvin. Tähän pätee vanha sanonta: ”Hyvin suunniteltu on puoliksi tehty”.

LÄHTEET

w3schools.com. HTML(5) tutorial. 2015. Viitattu 10.7.2015.

<http://www.w3schools.com/html>.

w3schools.com. PHP tutorial. 2015. Viitattu 10.7.2015.

<http://www.w3schools.com/php>.

w3schools.com. SQL tutorial. 2015. Viitattu 10.7.2015.

<http://www.w3schools.com/sql>.

tcpdf.org. 2015. Viitattu 26.7.2015.

<http://www.tcpdf.org>.

slideshare.net. Toiminnanohjausjärjestelmä (ERP). 2015. Viitattu 27.10.2015.

<http://www.slideshare.net/villekaseva>