



# TUNTIKORTTI-JÄRJESTELMÄN SUUNNITTELU JA TOTEUTUS

**Opinnäytetyö**

**Jani Tuovinen**

**Tietotekniikan koulutusohjelma**  
Ohjelmistotekniikka

Hyväksytty \_\_\_\_ . \_\_\_\_ . \_\_\_\_ \_\_\_\_\_

# SAVONIA-AMMATTIKORKEAKOULU TEKNIikka KUOPIO

Koulutusohjelma

Tietotekniikan koulutusohjelma

Tekijä

Jani Tuovinen

Työn nimi

Tuntikortti-järjestelmän suunnittelu ja toteutus

Työn laji

Insinöörityö

Päiväys

17.1.2011

Sivumäärä

33

Työn valvoja

Lehtori Jussi Koistinen

Yrityksen yhdyshenkilö

Toimitusjohtaja Petri Hyvärinen

Yritys

ITC-Solution Group Oy

Tiivistelmä

Tämän insinöörityön aiheena oli suunnitella ja toteuttaa Ergo-kalusteet Oy:lle tuntikortti-järjestelmä, jolla toteutettaisiin yrityksen työajanseuranta. Työ tehtiin ITC-Solution Groupille, joka toimitti järjestelmän asiakkaalle.

Järjestelmässä tuli olla käyttäjälle näkyvät tuntikorttiosio, jolla töiden aloitus ja lopetus toteutettaisiin, sekä jälkisyöttö. Jälkisyötöllä voitaisiin syöttää muualla tehdyt työt tai työajan lyhennys vapaat. Hallinnan puolella tuli löytyä käyttäjien, laskentakohteiden, tilausten, työkohteiden, työvaiheiden sekä lisien lisäys- ja muokkaus -ominaisuudet. Hallintatunnuksilla kirjautuessa käyttöön piti saada myös raportit osio, josta tuli saada tulostettua tuntiraportteja työntekijöiden töistä ja laskentakohteista sekä tilanneraportteja työvaiheista. Järjestelmä toteutettiin HTML, PHP, JavaScript ja AJAX tekniikoilla. Tietokantamootorina toimii MySQL.

Lopputuloksena saatiin toimiva työajanseurantajärjestelmä, joka helpottaa ja yksinkertaistaa asiakkaan jokapäiväistä työprosessia. Projektin läpivienti oli haastava mutta opettavainen kokemus.

Avainsanat

PHP, MySQL, työajanseuranta

Luottamuksellisuus

julkinen

# SAVONIA UNIVERSITY OF APPLIED SCIENCES

Degree Programme

Information Technology

Author

Jani Tuovinen

Title of Project

Design and Implementation of Working Time Monitoring System Tuntikortti

Type of Project

Final Project

Date

17 January, 2011

Pages

33

Academic Supervisor

Mr Jussi Koistinen, Lecturer

Company Supervisor

Mr Petri Hyvärinen, CEO

Company

ITC - Solution Group Oy

Abstract

The aim of this project was to design and produce a working time tracking software for Ergo-Kalusteet Oy. The work was done for ITC – Solution Group Oy which supplied the system to the customer.

The system required an hour card part to start and finish the work. The system also needed a part to put in afterwards the work done elsewhere. The management side of the system had to include a part with editing and inserting features of the users, customers, orders, tasks and operations together with additional payments. When logging in with management ID a part for printing reports was also required. When required, hour reports of employees or customers can be printed. A situation report for current active tasks of the system can also be printed. The system was developed by using PHP, HTML, JavaScript and AJAX. The database was implemented by using MySQL.

As a result of the project there was an efficient working time monitoring system which helps and simplifies the everyday working process of the customer. Working through the project was a challenging but very educational task.

Keywords

PHP, MySQL, working time monitoring

Confidentiality

public

## SISÄLLYSLUETTELO

LYHENTEET JA KÄSITTEET .....	6
1. JOHDANTO .....	7
2. YMPÄRISTÖ JA KÄYTETTÄVÄT TEKNIIKAT .....	8
2.1 Ohjelmointityökalut ja kehitysympäristö .....	8
2.2 WAMP .....	8
2.3 PHP .....	9
2.4 JavaScript.....	9
2.5 Ajax .....	10
2.6 MySQL.....	10
3. MÄÄRITTELY JA SUUNNITTELU .....	11
3.1. Määrittelyn lähtökohdat.....	11
3.2 Toiminnallinen määrittely .....	11
3.3 Käytettävyys .....	12
3.4 Client – Server arkkitehtuuri .....	13
4. TOTEUTUS.....	14
4.1 Käyttöliittymä ja rakenne.....	14
4.2 Tietokanta.....	15
4.3 Tiedostorakenne ja luokat.....	17
4.4 Tietoturva.....	19
4.5 Tuntikortti-osio .....	20
4.6 Jälkisyöttö-osio .....	23

4.7 Hallinta-osio.....	24
4.8 Raportti-osio .....	26
5. TESTAUS JA KÄYTTÖÖNOTTO.....	29
6. JATKOKEHITYS .....	30
6.1 Ylläpito.....	30
6.2 Ominaisuuksien kehittäminen.....	30
6.3 Järjestelmän laajennus .....	30
7. POHDINTA.....	31
7.1 Onnistumiset.....	31
7.2 Haasteet ja ongelmat.....	31
7.3 Tavoitteiden saavuttaminen.....	32
7.4 Loppusanat.....	32
LÄHTEET .....	33

## LYHENTEET JA KÄSITTEET

HTTP Request	Asiakassovelluksen pyyntö palvelinsovellukselle
AJAX	Lyhenne sanoista Asynchronous JavaScript And XML
WAMP	Kokoelma ohjelmia jotka muodostaa WWW-palvelimen
PHP	Lyhenne sanoista <i>PHP: Hypertext Preprocessor</i>

## 1. JOHDANTO

Työntekijöiden työajanseuranta on tärkeä osa suurimmassa osassa yrityksiä. Työajanseurannan avulla työntekijä sekä työnantaja pysyvät ajan tasalla kuinka paljon töitä on tehty. Myös palkanlaskenta- ja maksu ovat kiinteästi sidoksissa työajanseurantaan jos palkkauksen perustana on tuntipalkkaus tai muu aikaan sidottu korvaus.

Työn tavoitteena on suunnitella ja uudistaa Ergo-Kalusteet Oy:n työajanseuranta. Ergo-Kalusteilla on ollut viimeiset kymmenen vuotta käytössään Excel-pohjainen tuntikortti-järjestelmä. Nykyinen Excel-taulukko on toimiva, mutta suuritöinen. Jokaisella käytössä olevalla koneella täytyy olla oma versio ko. Excelistä, jotka sitten täytyy yhdistellä tietyn väliajoin. Tästä johtuen historiatietojen tallentaminen vaatii useiden samankaltaisten tiedostojen säilytystä, joka on työlästä.

Järjestelmä haluttiin uudistaa tekemällä siitä verkkopohjainen sekä saada tiedot tallennettua tietokantaan. Tällä päästäisiin useiden tiedostojen käsittelystä eroon sekä järjestelmän käyttö mahdollistuisi yrityksen sisäverkossa miltä tahansa koneelta. Historiatiedot pysyvät tallessa tietokannassa ja yhden tietokannan varmuuskopiointi on huomattavasti helpompaa kuin useiden tiedostojen hallinta.

## 2. YMPÄRISTÖ JA KÄYTETTÄVÄT TEKNIIKAT

### 2.1 Ohjelmointityökalut ja kehitysympäristö

Ohjelmointityökaluja valittaessa päätettiin lähteä hyödyntämään web-ohjelmoinnin laajaa Open Source-työkalujen tarjontaa.

Itse ohjelmointityökaluksi valittiin ilmainen GPL-lisenssiin pohjautuva Notepad++ lähdekoodieditori, joka tukee useita eri ohjelmointikieliä mm. PHP. Ohjelma sisältää useita edistyneitä ominaisuuksia, mm. syntaksikorostuksen, jotka tekevätkin Notepad++:sta hyvän vaihtoehdon Windows:n mukana tulevalle perus Notepad:lle.

Kehitysympäristönä toimi ITC-Solution Group Oy:n ohjelmointikone. Koneen käyttöjärjestelmänä toimi Windows 7 Professional, johon oli asennettu ohjelmistokehityksen kannalta kaikki tarpeelliset työkalut ja alustat, kuten Notepad++ sekä WAMP, josta tarkemmin seuraavassa kappaleessa.

### 2.2 WAMP

WAMP on Windowsille tarkoitettu ohjelmistokokonaisuus, joka on muunnos LAMPista. Ensimmäinen kirjain osoittaa kummalle käyttöjärjestelmälle se on tarkoitettu.

Loput kirjaimet eli, A tarkoittaa Apachea, M tarkoittaa MySQL:ää ja P tarkoittaa PHP:tä.

WAMP on kokonaisuutena ilmainen jonka jakelu tapahtuu GPL lisenssillä. WAMPin asennus ja käyttöönotto on erityisen helppoa, koska siihen on koottu kaikki palaset yhteen pakettiin, joilla pääset tekemään omalla koneella kokonaisia sivustoja ja myös testaamaan niitä itse.



## 2.3 PHP

PHP (PHP: *Hypertext Preprocessor*) on skriptikieli, joka on tarkoitettu palvelinpuolen ohjelmointiin. WWW-palvelimelle asennettava PHP-ohjelmisto tulkaa PHP-ohjelman. Ohjelman generoima tulos palautuu asiakkaalle (yleensä www-selaimelle) esitettäväksi.

PHP:ssä on hyvin laaja luokkakirjasto jonka vuoksi se on helppokäyttöinen. PHP on komentosarja- eli skriptikieli, joka tulkitaan vasta ohjelman suoritusvaiheessa. PHP koodi upotetaan yleisesti HTML-sivujen sisään. Koodi sijoitetaan PHP:lle varattujen aloitus- ja lopetustagien sisään (<?php ja ?>). Näistä tageista serverin päässä oleva tulkkausohjelma ymmärtää suorittaa koodin ja palauttaa sen selaimelle. Selain tulostaa käyttäjälle sen, mitä serverin pään tulkki käskää sitä tulostamaan. Käyttäjälle PHP:n koodi ei näy, mutta serverillä saatetaan suorittaa useita satoja rivejä koodia. Selaimessa ei tarvita tukea PHP:lle, koska kaikki suoritetaan serverillä.

## 2.4 JavaScript

Javan luomisen aikoihin Netscape kehitti scriptauskielen, jota kutsuttiin JavaScriptiksi. JavaScript kehitettiin alun perin helpottamaan kehittäjiä muokkaamaan HTML-kielen tageja dynaamisesti, jotta verkkosivut tuottaisivat käyttäjille aiempaa paremman käyttökokemuksen. Vaikka kieltä on haukuttu sen kehitystyökalujen, havainnollisten virheilmoitusten ja virheidenjäljittämisen puutteiden vuoksi, on JavaScript tehokas tapa dynaamisten verkkosovellusten luomiseen. (Asleson & Scutta 2007: 8)

Siinä missä PHP on serverin pään skriptikieli, JavaScript suoritetaan clientin päässä. Tämän takia selaimessa on oltava tuki JavaScriptille. Sekin on erittäin yleinen web-maailmassa ja sillä toteutetaan dynaamista toiminnallisuutta web-sivuille.

## 2.5 Ajax

Ajax (Asynchronous JavaScript And XML) kuten koko nimitys kertoo, on joukko teknologioita joita käytetään web-sovelluskehityksessä luodessa vuorovaikutteisia asynkroonisia websovelluksia. JavaScriptilla voidaan lähettää HTTP-pyyntöjä serverille, joille alun perin palautettiin XML-merkkäystä. Nykyään Ajax nimitystä käytetään samankaltaiseen toimintatapaan jossa, serveriä pyydetään suorittamaan pientä toimintoa, niin ettei tarvitse ladata koko web-sivua uudelleen.

Tällä hetkellä Ajax on merkittävin vaihtoehto monipuolisten verkkosovellusten luomiseen. Ajax on enemmänkin tekniikka kuin teknologia, vaikka aiemmin esitelty JavaScript onkin Ajaxin keskeinen komponentti. Ajaxissa ei varsinaisesti ole mitään uutta. Sen uusin teknologia XMLHttpRequest julkaistiin ActiveX-komponenttina jo Internet Explorerin viidennen version myötä keväällä 1999. Ajaxin vahvuus on että se rakentuu jo olemassa olevien tekniikoiden varaan, jolloin kehittäjien ei tarvitse opetella uutta ohjelmointikieltä tai muokata jo valmiita palvelimilla olevia ratkaisuja. Ajax on asiakaspuolen ratkaisu ja se voidaan yhdistää J2EE:n, .NET-tekniikan, PHP:n, Ruby:n ja CGI-scriptien käyttöön. (Asleson & Scutta 2007: 14)

## 2.6 MySQL

PHP sopii hyvin yhteen MySQL-tietokantaohjelmiston kanssa, koska molemmat ovat yksityiselle käyttäjälle ilmaisia. MySQL on GPL-lisenssiin pohjautuva SQL-tietokannan hallintajärjestelmä joka on erittäin suosittu verkkopohjaisissa sovelluksissa. SQL tulee sanoista Structured Query Language. Kyseessä on kieli relaatiotietokantojen käsittelyyn. (Kolehmainen 2006: 281-285)

Tietokannan hallinnointi tapahtuu WAMP:iin liitetyllä graafisella phpMyAdmin ohjelmalla.

## **3. MÄÄRITTELY JA SUUNNITTELU**

### **3.1. Määrittelyn lähtökohdat**

Lähtökohtana määrittelylle toimi yrityksen päivittäisessä käytössä oleva yli kymmenen vuotta vanha Excel-pohjainen tuntikortti.

Excel-pohjasta tuli säilyttää käyttöliittymä mahdollisimman tarkasti, koska käyttäjät olivat siihen niin hyvin tottuneet ja se oli yrityksen toiminnan kannalta yhäkin varsin toimiva.

### **3.2 Toiminnallinen määrittely**

Järjestelmästä ei laadittu mitään tarkempaa määrittelydokumenttia, vaan tarkoituksena oli siirtää nykyinen Excel-pohjan toimintaperiaate suoraan sellaisenaan tietokantamoottorin päälle. Tämä antoi tiettyä helpotusta mutta myös jossain määrin tuotti vaikeuksia koska ihan kaikkea ei kuitenkaan vanhasta pohjasta selville saanut.

Excelistä piti selvittää myös mitä kaikkea tietokantaan olisi saatava tallennettua.

### 3.3 Käytettävyys

”Jos uusi järjestelmä toimii käyttäjälle niin kuin vanhakin, niin se toimii hyvin” oli asiakkaan kommentti kun mietimme järjestelmän käyttöliittymää ja käytettävyyttä. Tästä oli hyvä lähteä rakentamaan toimivaa käyttöliittymää sovellukselle.

Henkilö Testi Käyttäjä	Urakka vai tuntityö <input type="radio"/> Urakka <input checked="" type="radio"/> Tuntityö	Lisät <input type="checkbox"/> Vuorollisä <input type="checkbox"/> Ylityö <input type="checkbox"/> olosuhdelisä <input type="checkbox"/> 1/2-Pvr <input type="checkbox"/> 1/1-Pvr <input type="checkbox"/> Km-korv <input type="text" value="Km"/>
Laskentakohde Työ 16	Kappalemäärä <input type="text"/>	<b>TYÖN ALOITUS</b>
Työkohde kohde2	Kappalehinta 0	<b>TYÖN LOPETUS</b>
Työvaihe asennus		<b>JÄLKISYOTTO</b>

Pvm	Alkoi	Loppu	Nimi	Laskentakohde	Työkohde	Työvaihe
8.11.2010	13:43	15:10	Testi Käyttäjä	Työ 16	kohde2	asennus
Jälkisyöttö d.9.yyyy			Testi Käyttäjä	Työ 16	kohde2	asennus

**OOT HUILISSA !!**

<b>Viikkokertymä</b>	<b>1,44</b>
----------------------	-------------

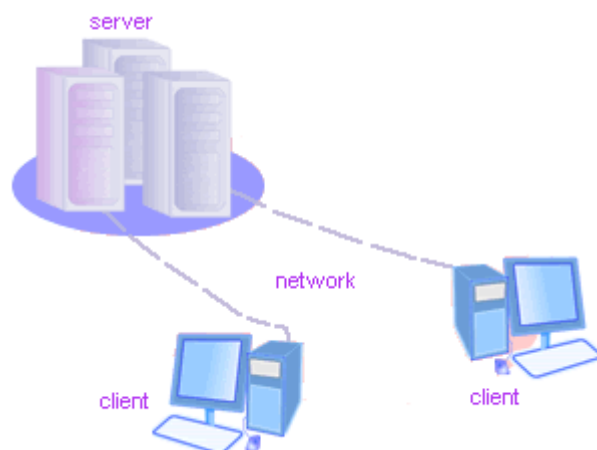
Kuva 1. Excel-pohjan käyttöliittymä

Vanhasta käyttöliittymästä ei tarvittu aivan kaikkia osioita, urakkatöitä yrityksessä ei enää tehdä joten se jäi pois tarpeettomana, samoin myös kilometrikorvaukset ja päivärahat jätettiin pois.

Henkilö ja tarkemmat työvalinnat kaivoin myös Excelistä. Tässä vaiheessa ilmeni että laskentakohteita ja työkohteita ei voisi kaikkia ladata suoraan tietokannasta alasvetovalikkoihin vaan tarvittaisiin dynaaminen toiminto jossa generoidaan tietyt kohdat tiettyjä valintoja varten.

### 3.4 Client – Server arkkitehtuuri

Vanhan Excel-pohjan suurimpia ongelmia on että monelta eri koneelta työaikoja leimattaessa, tiedot jäävät aina paikalleen. Jonkun on vähintään kerran kuukaudessa tehtävä erikseen työtä niiden keskittämiseen. Tätä varten päätimme, että sovellus tulee toteuttamaan Client – Server arkkitehtuuria.



**Kuva 2. Client – Server arkkitehtuuri**

Palvelin tarjoaa tietokannan, sivuston tiedostot sekä tietoturvaan liittyvät resurssit. Käyttöliittymä toimii internet-selaimen välityksellä asiakkaana sovelluksessa.

Tällä tavalla päästään turhasta tietojen manuaalisesta yhdistämisestä. Mallin muita etuja on että sovelluksen käyttö on paikkariippumaton yrityksen sisällä kun kone on kiinni yrityksen sisäverkossa. Yrityksen leimauskoneet ovat erittäin vanhoja, joten client-server arkkitehtuurilla nämä koneet voisivat jatkaa yhäkin toimintaansa, koska sovelluksen suoritus tapahtuu serverillä.

Huonona puolena tässä arkkitehtuurissa on että, jos serveri tai verkko jumiutuu niin, asiakaskoneet eivät voi ohjelmaa suorittaa.

## 4. TOTEUTUS

### 4.1 Käyttöliittymä ja rakenne

Yrityksen työntekijät ovat käyttäneet reilut kymmenen vuotta saman Excelin käyttöliittymää, joten toteutuksessa lähdettiin seuraamaan suoraan sitä. Usean työntekijän yrityksessä tuntileimauksen käyttöliittymän täytyy olla sujuvatoiminen sekä mahdollisimman pitkälle automaattinen. Silloin käyttäjän ei tarvitsisi miettiä mitä hän aamulla tekee, eikä leimauspisteelle syntyisi jonoja.

ERGO KALUSTEET OY

Käyttäjä: Admin Admin | Käyttöoikeus: Admin | [Kirjaudu ulos](#)

09:00

Tuntikortti  
Jälkisyöttö  
Hallinta  
Raportit

Henkilö: Admin Admin

Laskentakohde: ITC

Tilaus: 10101010

Työkohde: ITC-toimisto

Työvaihe: Asennus

Lisät:  
 Vuorolisä  
 Olosuhdeisiä  
 Yityö

Vaihe valmis ?

SYÖTÄ OMA KOODI:

Tehdyt tunnit : 6.50

**Aloita**

Avaa	Työntekijä	Älku	Loppu	Laskentakohde	Tilaus	Työkohde	Työvaihe
	Admin Admin	16.11.2010 09:00:00	16.11.2010 13:00:00	ITC	10101010	ITC-toimisto	Asennus
	Admin Admin	01.11.2010 13:30:00	01.11.2010 14:30:00	ITC	10101010	ITC-toimisto	Esivalmistelu
	Admin Admin	01.11.2010 12:00:00	01.11.2010 13:30:00	ITC	10101010	ITC-toimisto	Esivalmistelu

© ITC-Solution Group Oy 2010

Kuva 3. Käyttöliittymä

Järjestelmän rakenne on kaksitasoinen, käyttäjillä on joko admin- tai user-oikeudet jotka määrittelee mitä tasoja voidaan käyttää. Järjestelmästä löytyy neljä pääosiota joiden alta löytyy aliosioita. Pääosioita järjestelmässä on V Tuntikortti, Jälkisyöttö, Hallinta ja Raportit. Sivustolle tultaessa index.php sivun kautta käyttäjä ohjataan ilman kirjautumista Tuntikortti-sivulle joka toimii järjestelmän päänäyttönä. Päävalikosta voidaan navigoida hallinta-osioon joka, vaatii sisäänkirjautumisen vähintään admin tason oikeuksilla. Hallintaan kirjautumisen jälkeen avautuu linkki raportti-osioon.

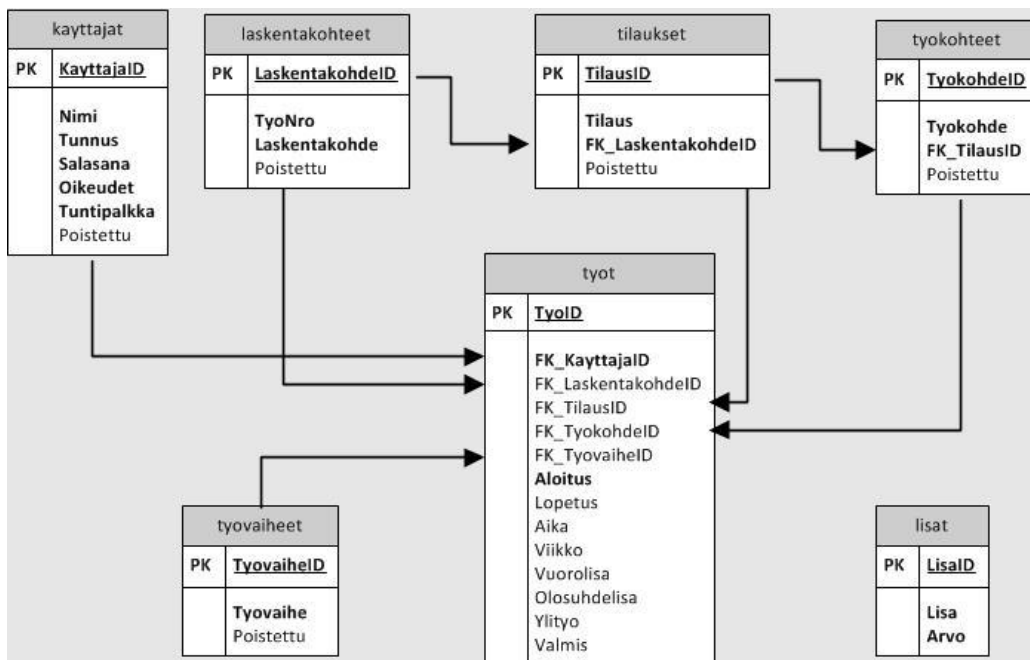
## 4.2 Tietokanta

Tietokannan toteutuksessa lähdettiin siitä, että työt saataisiin tallennettua sisältäen tarpeelliset tiedot. Kannassa tauluina ovat: käyttäjät, laskentakohteet, tilaukset, työkohteet, työvaiheet, työt ja lisat. Kaikkiin tauluihin generoidaan pääavaimet automaattisesti kasvavalla numerolla.

Seuraavassa kuvataan lyhyesti tietokannan taulut:

- Käyttäjät tauluun tallennetaan järjestelmän käyttäjien tiedot.
- Laskentakohteet tauluun tallennetaan kustannuspaikat työnumeroineen.
- Tilaukset tauluun tallennetaan laskentakohteiden tilaukset, joita voi olla useampia yhtä laskentakohdetta kohden. Jotta tilaukset saadaan linkitettyä tiettyyn laskentakohteeseen, sille tallennetaan laskentakohteen id numero.
- Työkohteet tauluun tallennetaan tilauksille kuuluvat työkohteet joita myös voi olla useita yhtä tilausta kohden.
- Työvaiheet tauluun tallennetaan työvaiheet jotka pysyvät käytännössä aina samoina.
- Lisat tauluun on tallennettu olosuhde- ja vuorolisä joille voidaan tallentaa vaihtuvia arvoja.

- Työt tauluun tallennetaan id numerot käyttäjästä, laskentakohteesta, tilauksesta, työkohteesta sekä työvaiheesta. Työn aloituksessa tallennetaan vain käyttäjän id numero sekä työn aloitusaika. Työtä lopetettaessa sille lisätään käyttöliittymän valintojen mukaan loput tiedot sekä lopetusaikaleima.



**Kuva 4. Tietokantakaavio**

Tietokanta ei syntynyt tyhjästä suoraan vaan sitä alettiin kasvattaa tarpeen mukaan. Kehitys tapahtui moduuli kerrallaan. Kirjautumista tehtäessä syntyi käyttäjä taulu ja työn tallennuksen mukana rakennettiin työt taulu ja sitä rataa. Moduulien sisäiset taulutkaan eivät rakentunut kerralla valmiiksi vaan esimerkiksi työt taulu kasvoi useita kertoja sitä mukaan kun tarvittiin tallentaa lisätietoja.



### 4.3 Tiedostorakenne ja luokat

Tiedostorakenne koostuu PHP- ja INC-tiedostoista. Järjestelmä rakentuu kokonaan etusivu.php-tiedoston päälle johon muut inc-päätteiset alisivut liitetään require\_once()-funktiolla. INC-tiedoston sisältämä koodi liitetään siihen kohtaan PHP-sivua missä sitä kutsutaan. Näin ollen sivun html rakennetta ei tarvitse kirjoittaa muualle kuin etusivulle.

Käyttäjän navigoita jollekin sivulle, komento kuljetetaan GET-metodilla c-muuttujassa etusivu.php-tiedostolle jossa se otetaan kiinni ja sijoitetaan action muuttujaan. Action muuttujaa verrataan if-lauseessa jossa katsotaan mitä sivua valinta koskee.

```
if(isset($_GET["c"]))
{
    $action = $_GET["c"];
    if ($action == "tuntikortti")
    {
        require_once ("tuntikortti.inc");
    }
    else if ($action == "hallinta")
    {
        require_once ("hallinta.inc");
    }
}
```

Kuva 5. Navigointi

Järjestelmässä on DBManager-, CheckLogin-, Error-, sekä Hashluokat. DBManagerluokka sisältää kaikki tietokantaan ja sen modifiointiin liittyvät funktiot.

Eräitä tärkeimpiä funktioita tässä luokassa on tietokantayhteyden avaus- ja sulkufunktiot sekä DoSQL-funktio. Luokka sisältää pääasiassa vain tietokannasta hakemiseen liittyvät funktiot. Muut muokkaukset ja lisäykset tehdään save\_changes.php tiedoston avulla. Save\_changes ei varsinaisesti ole luokka

vaan se on tiedosto joka vastaanottaa formeilta tulevat käskyt vastaan ja kokoaa SQL kyselylauseen joka sitten suoritetaan DoSQL-funktiolla.

CheckLogin-luokka hoitaa kaikki käyttäjien oikeuksien valvontaan liittyvät toimenpiteet. Käyttäjän syöttäessä työtä tai kirjautuessa hallintaan CheckLogin tarkastaa käyttäjän tunnuksen ja salasanan tietokannasta sekä onko käyttäjällä riittävät oikeudet kyseiseen toimintoon. CheckLogin myös sisältää funktion joka tarkastaa kuinka pitkään järjestelmä on ollut käyttämättömänä ja tunnin jälkeen se poistaa aktiiviset kirjautumiset jolloin käyttäjän on kirjaututtava uudestaan.

Error-luokan nimi on hieman harhaanjohtava sillä se ei pelkästään käsittele virhe tilanteita vaan informoi käyttäjälle myös jos toiminto on suoritettu onnistuneesti. Esimerkiksi jos käyttäjä muokkaa jo syötettyä työtä niin Error-luokka kertoo onnistuiko muutoksien tallennus vai ei.

Hash-luokan tehtävänä on kryptata käyttäjien salasanat niitä lisättäessä järjestelmään tai salasanaa vaihdettaessa. Myös kirjautumisen yhteydessä salasanaa tarkistettaessa käytetään hash-luokkaa tekemään käyttäjän syötteestä kryptattu versio, jota sitten peilataan tietokannassa olevaan kryptattuun salasanaan.

## 4.4 Tietoturva

Järjestelmässä on kaksi käyttäjätasoa jolla tietoturva on toteutettu. User käyttäjätasolla toimivat käyttäjät voivat käyttää ainoastaan tuntikortti-osiota tuntiensa leimaamiseen ja admin tason käyttäjille on annettu oikeus kirjautua myös hallintaan ja raportteihin.

Salasanat on kryptattu MD5-algoritmilla ennen kuin ne on tallennettu tietokantaan. Sisäänkirjautumisen yhteydessä käyttäjän syöttämä salasana kryptataan joka kerta uudestaan ja kryptattua versiota verrataan tietokantaan tallennettuun. MD5-kryptaus on tehokas tapa kryptata salasanoja tietokantaan, koska siitä ei käytännössä voida enää tuottaa selvää tekstiä. Ainoa tapa on kryptata myös toinen teksti ja verrata näitä keskenään.

*Todellisuudessa ei ole mahdollisuutta tuottaa MD5-tiivisteestä alkuperäistä viestiä. Jos tämä olisi mahdollista, sen käyttö tiedon pakkaamisessa olisi äärimmäisen suositeltavaa, koska esimerkiksi neljän gigatavun tiedosto voitaisiin rakentaa vain 32 merkin jonosta. Siksi sitä kutsutaan yhden suunnan hajakoodausalgoritmiksi. (Wikipedia. 2010, hakusana MD5)*

Tiedostorakenteesta kerroin että osa järjestelmästä rakentuu .inc tiedostojen päälle ja näihin liittyy myös tietoturvariski. Jos käyttäjä jotenkin osaisi osoitekenttään suoraan kirjoittaa esimerkiksi edellä mainitun dbmanager.inc tiedoston polun, web serveri näyttäisi sen puhtaasti tekstitiedostona. Tämä pitää estää sijoittamalla .htaccess-tiedosto järjestelmän juureen jossa on koodi:

```
<Files ~ "\.inc$">  
Order deny,allow  
Deny from all  
</Files>
```

Tällä estetään kaikilta käyttäjiltä .inc päätteisten tiedostojen katselun.

Dbmanager.inc tiedoston vapaa tarkastelu näyttäisi melkein kaikki järjestelmän käyttämät tietokannan käsittelyyn liittyvät funktiot joiden avulla järjestelmän väärinkäyttö helpottuisi huomattavasti.

## 4.5 Tuntikortti-osio

Tuntikortti-osion päätoimintona on työntekijöiden tuntien ylöskirjaaminen. Aamulla kun työntekijä saapuu, hän tulee leimauspisteelle ja valitsee nimensä listasta sekä syöttää oman tunnuksen ja painaa aloita. Kun tietty työtehtävä on sillä kertaa ohitse, hän tulee uudestaan pisteelle ja valitsee laskentakohteen, tilauksen, työkohteen, työvaiheen sekä tarvittaessa lisät. Lisäksi hän laittaa rastin ruutuun, jos kyseessä oleva työvaihe on valmis. Valintojen jälkeen hän syöttää oman tunnuksensa ja painaa lopeta. Aikaisemmin aloitetulle työlle päivittyä loppumisaika ja muut syötetyt tiedot.

Työntekijälle näkyy myös viikottaiset tehdyt tunnit ja viisi viimeisintä työtä. Viimeisimmistä töistä työntekijät voivat muokata loppumisaikaa jos työ on jostain syystä jäänyt päälle töistä lähdeittäessä. Kuvasta 6 näkyy tuntikortin pääominaisuudet ja toiminnot.

Avaa	Työntekijä	Alku	Loppu	Laskentakohte	Tilaus	Työkohte	Työvaihe
	Admin Admin	16.11.2010 09:00:00	16.11.2010 13:00:00	ITC	10101010	ITC-toimisto	Asennus
	Admin Admin	01.11.2010 13:30:00	01.11.2010 14:30:00	ITC	10101010	ITC-toimisto	Esivalmistelu
	Admin Admin	01.11.2010 12:00:00	01.11.2010 13:30:00	ITC	10101010	ITC-toimisto	Esivalmistelu

Kuva 6. Tuntikortti-osio

Käyttäjän valinnat toteutettiin dynaamisesti Ajaxilla. Kun käyttäjä valitsee listasta nimen, määrätyt sivun osat vaihtuvat käyttäjän edellisten valintojen perusteella.

Henkilövalikon onchange() event lähettää http-requestin serverille chainfunc.php tiedostolle jossa pyyntö käsitellään ja haetaan tietokannasta onko käyttäjällä jo tallennettuja töitä. Jos käyttäjällä ei ole yhtään työtä tietokannassa, laskentakohde valikkoon generoidaan laskentakohteet, loput alasvetovalikot pysyvät tyhjinä ja viimeisien töiden kohdalla käyttäjää informoidaan asiasta. Kun käyttäjä valitsee laskentakohteen, lähetetään uusi request jolloin pyydetään serveriä palauttamaan kyseessä olevalle laskentakohteelle kuuluvat tilaukset. Tietty tilaus valittaessa taas haetaan sille kuuluvat työkohteet.

Käyttäjällä jolla on jo tehtyjä töitä, haetaan valinnat alasvetovalikkoihin viimeisen tehdyn työn perusteella ja haetaan taulukkoon maksimissaan viiden viimeisimmän työn tiedot.

Viittä viimeisintä työtä voidaan tarkastella tarkemmin taulukosta Avaa kohdasta. Jos työntekijä on epähuomiossa lähtenyt työpäivän jälkeen kotiin ja seuraavana aamuna huomaa, että työ on jäänyt päälle ja tunnit ovat juosseet koko yön. Tätä kautta he voivat muokata lopetusajan oikeaksi. Muokkauksesta jää luonnollisesti jälki, jotta töitä ei muokattaisi ihan mielivaltaisesti.

09:00

Tuntikortti  
Jälkisyöttö  
Hallinta  
Raportit

Tehty työ:

Työntekijä: Admin Admin  
Laskentakohde: ITC  
Tilaus: 101010  
Tyokohde: ITC-toimisto  
Tyovaihe: Asennus  
Aloitus: 16.11.2010 09:00:00  
Lopetus: 16.11.2010 13:00

Tallenna

Kuva 7. Työn tarkastelu / muokkaus

Lopetus ajan voi joko syöttää itse tai käyttää JavaScriptillä toteutettua DateTime Pickeriä joka aukeaa pienestä kalenterin kuvasta. Samaa lisäosaa on käytetty muuallakin järjestelmässä.



Kuva 8. Päivämäärän ja ajan valitsin

## 4.6 Jälkisyöttö-osio

Toisinaan työntekijöille tulee tilanne että he ovat tekemässä työtä tehtaan ulkopuolella ilman web yhteyksiä. Nämä muualla tehdyt työt tai vaikka pekkaspäivät voidaan merkata jälkisyöttönä järjestelmään. Sitä varten järjestelmään tehtiin erillinen jälkisyöttö-osio.

Avaa	Työntekijä	Alku	Loppu	Laskentakohde	Tilaus	Työkohde	Työvaihe
	Admin Admin	11.11.2010 08:00:00	11.11.2010 16:25:35	ITC	101010	ITC-toimisto	Asennus

**Kuva 9. Jälkisyöttö-osio**

Jälkisyötön ulkoasu on miltei samanlainen kuin tuntikortinkin, mutta toiminta eroaa jonkin verran. Käyttäjä valitsee työtehtävänsä alasvetovalikoista ja syöttää työn päivämäärät aloitus ja lopetus aikoiheen ja tallentaa työn. Työ tallentuu tietokantaan normaalisti, mutta se on merkattu jälkisyötetyksi, jotta niitä voidaan seurata. Jälkisyötetty työ näkyy myös tuntikortin viimeisimmissä tehdyissä töissä mutta jälkisyöttö-osiossa ei näy kuin jälkisyötetyt työt.

## 4.7 Hallinta-osio

Käyttäjän navigoidessa hallinta osioon, tarkastetaan sessionista onko käyttäjä kirjautunut ja onko käyttäjällä riittävät oikeudet. Jos käyttäjä on kirjautunut, mutta kuuluu user-tasolle, hänet ohjataan virhesivulle. Siellä hänelle informoidaan, että hänellä ei ole oikeuksia. Kirjautumaton käyttäjä ohjataan kirjaudu.php sivulle, jossa pyydetään antamaan käyttäjätunnus ja salasana. Admin-tason käyttäjätunnukset syötettäessä salasana tarkastetaan CheckLogin-luokassa, jossa salasana kryptataan ja verrataan tietokannassa olevaan kyseisen tunnuksen alle tallennettuun salasanaan.

Tunnuksien ollessa oikeat, käyttäjä ohjataan hallintasivulle jossa päänäyttönä näytetään järjestelmän käyttäjä listaus. Myös raportit osion linkki tulee näkyviin päävalikkoon. Sivun ylälaidasta löytyy valikko josta voidaan navigoida hallinnan alasuviilla.

The screenshot displays a web application interface for user management. On the left, there is a sidebar with a clock showing 09:00 and a menu with options: Tuntikortti, Jälkisyöttö, Hallinta, and Raportit. The main content area has a navigation bar with tabs: Käyttäjät (selected), Laskentakohteet, Tilaukset, Työkohteet, Työvaiheet, and Lisät. Below the navigation bar, there are sub-tabs: Uusi käyttäjä and Käyttäjät. The main section is titled 'Järjestelmän käyttäjät:' and contains a table with the following data:

Muokkaa	Nimi	Käyttäjätunnus	Käyttötaso	Tuntipalkka	P
	Admin Admin	admin	Admin	14.5	E
	User User	user	User	9.5	E

Below the table, there is a form titled 'Uuden käyttäjän tiedot:' with the following fields:

- \*Nimi: Admin
- \*Käyttäjätunnus: admin
- \*Käyttötaso: Admin (dropdown menu)
- \*Tuntipalkka: 14.5
- \*Salasana: •••••
- \*Varmista salasana: •••••

A 'Tallenna' button is located at the bottom right of the form.

Kuva 10. Hallinta-osio



Hallinnan ensimmäisellä sivulla eli käyttäjät voidaan hallinnoida käyttäjien tietoja. Sivulla on listaus kaikista käyttäjistä ja niistä nähdään tärkeimmät tiedot. Yksittäisen käyttäjän tarkempia tietoja voidaan tarkastella ja muokata käyttäjän muokkaus sivulla jonne pääsee klikkaamalla Muokkaa kuvaketta. Muokkaus sivulla voidaan myös poistaa tai palauttaa käyttäjä. Poistettaessa käyttäjä ei häviä järjestelmästä kokonaan vaan se vain merkitään poistetuksi, näin ollen tietokannan viittaukset tehtyihin töihin ei häviä.

The screenshot displays a web application interface with four main sections, each with a navigation bar and a data table/form below it.

**Section 1: Laskentakohteet**

Navigation: Käyttäjät | **Laskentakohteet** | Tilaukset | Työkohteet | Työvaiheet | Lisät

Table: Laskentakohteet

Muokkaa	Työnumero	Laskentakohde
	12345	ITC
	876543	Testilaskentakohde

Form: Lisää uusi laskentakohde

Anna työnumero:

Anna laskentakohde:

Tallenna

**Section 2: Tilaukset**

Navigation: Käyttäjät | Laskentakohteet | **Tilaukset** | Työkohteet | Työvaiheet | Lisät

Table: Tilaukset

Muokkaa	Tilaus	Laskentakohde
	101010	ITC
	101100	ITC

Form: Lisää uusi tilaus

Valitse laskentakohde:

Anna tilausnumero:

Tallenna

**Section 3: Työkohteet**

Navigation: Käyttäjät | Laskentakohteet | Tilaukset | **Työkohteet** | Työvaiheet | Lisät

Table: Työkohteet

Muokkaa	Tilaus	Työkohde
	101010	ITC-toimisto
	101100	Testityökohde

Form: Lisää uusi työkohde

Valitse tilaus:

Anna työkohde:

Tallenna

**Section 4: Työvaiheet**

Navigation: Käyttäjät | Laskentakohteet | Tilaukset | Työkohteet | **Työvaiheet** | Lisät

Table: Työvaiheet

Muokkaa	Työvaihe
	Ohjelmointi
	Asennus

Form: Lisää uusi työvaihe

Anna työvaihe:

Tallenna

Kuva 11. Hallinta-osion osat

## 4.8 Raportti-osio

Raportti-osion tuntiraportit alaosiosta voidaan generoida neljä erilaista raporttia kirjanpidolle ja palkanlaskentaan. Työntekijöistä voidaan tulostaa kokonaisraportti tai sitten yksittäisen työntekijän tunnit tietyltä aikaväliltä. Myös laskentakohteista voidaan tulostaa samanlaiset raportit.

The screenshot shows a web interface for generating reports. On the left is a sidebar with a clock showing 09:00 and menu items: Tuntikortti, Jälkisyöttö, Hallinta, and Raportit. The main area is divided into two sections:

**Section 1: Luo raportti työntekijöistä :**

- Buttons: Tuntiraportit (active), Tilanneraportti
- Form fields: Alkupäivämäärä (start date), Loppupäivämäärä (end date), Valitse työntekijä: (dropdown menu)
- Button: Luo raportti

**Section 2: Luo raportti laskentakohteista :**

- Form fields: Alkupäivämäärä (start date), Loppupäivämäärä (end date), Valitse laskentakohte: (dropdown menu)
- Button: Luo raportti

**Section 3: Luo tilanneraportti :**

- Buttons: Tuntiraportit, Tilanneraportti (active)
- Form field: Valitse laskentakohte: (dropdown menu)
- Button: Luo raportti

Kuva 11. Raportit-osio

Luo raportti nappia painettaessa lähetetään käyttäjät antamat päivämäärät tekijatuntirap.php tiedostolle jossa ensimmäisenä tarkastetaan, että käyttäjä on kirjautunut adminina ja sitten otetaan lähetetyt muuttujat vastaan POST-metodilla. Jos käyttäjä ei ole valinnut työntekijää tai laskentakohteita tulostetaan kaikkien työntekijöiden tai laskentakohteiden listaus. Valitsemalla joku työntekijä tai laskentakohte alavetovalikosta, saadaan valinnan mukainen listaus.

## Työntekijän tunnit:

Aloitus	Lopetus	Nimi	Lask.kohde	Tilaus	Työkohde	Työrahe	Tunnit	50%	100%	Vuor.	Olos.	Muok.	Jalk.	€	Lisär	Yht	
21.12.2010 06:09:13	21.12.2010 07:12:00	Testikäyttäj1	Testikäyttäj1	Testikäyttäj1	Testikäyttäj1	Testikäyttäj1	1.05	-	-	-	-	-	-	12.56	0.00	12.56	
21.12.2010 07:12:53	21.12.2010 08:39:03	Testikäyttäj1	Testikäyttäj1	Testikäyttäj1	Testikäyttäj1	Testikäyttäj1	1.44	-	-	-	-	-	-	17.23	0.00	17.23	
21.12.2010 14:53:54	21.12.2010 14:50:00	Testikäyttäj1	Testikäyttäj1	Testikäyttäj1	Testikäyttäj1	Testikäyttäj1	-0.07	-	-	-	-	-	-	-0.78	0.00	-0.78	
21.12.2010 08:57:28	21.12.2010 14:53:44	Testikäyttäj1	Testikäyttäj1	Testikäyttäj1	Testikäyttäj1	Testikäyttäj1	5.94	-	-	-	-	-	-	71.25	0.00	71.25	
21.12.2010 05:57:14	22.12.2010 11:27:07	Testikäyttäj1	Testikäyttäj1	Testikäyttäj1	Testikäyttäj1	Testikäyttäj1	5.50	-	-	-	-	-	-	65.98	0.00	65.98	
22.12.2010 11:49:34	22.12.2010 13:05:32	Testikäyttäj1	Testikäyttäj1	Testikäyttäj1	Testikäyttäj1	Testikäyttäj1	3.27	-	-	-	-	-	-	39.19	0.00	39.19	
23.12.2010 06:08:23	23.12.2010 11:33:03	Testikäyttäj1	Testikäyttäj1	Testikäyttäj1	Testikäyttäj1	Testikäyttäj1	5.41	-	-	-	-	-	-	64.93	0.00	64.93	
23.12.2010 11:58:20	23.12.2010 13:43:35	Testikäyttäj1	Testikäyttäj1	Testikäyttäj1	Testikäyttäj1	Testikäyttäj1	1.75	-	-	-	-	-	-	21.05	0.00	21.05	
27.12.2010 06:04:19	27.12.2010 11:40:59	Testikäyttäj1	Testikäyttäj1	Testikäyttäj1	Testikäyttäj1	Testikäyttäj1	5.61	-	-	-	-	-	-	67.33	0.00	67.33	
27.12.2010 11:51:22	27.12.2010 14:51:22	Testikäyttäj1	Testikäyttäj1	Testikäyttäj1	Testikäyttäj1	Testikäyttäj1	3.00	-	-	-	-	-	-	36.00	0.00	36.00	
														Yht:	394.75	0.00	394.75

## Työntekijöiden tuntilistaus:

Nimi	Tunnit yht.	50% (h)	100% (h)	Olosuhdelisa (h)	Vuorolisa (h)	Lisät €	Palkka €	Yht €
Admin Admin	4.00	0.00	0.00	0.00	0.00	0.00	40.00	40.00
User User	64.65	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Testikäyttäj1	68.25	0.00	0.00	0.00	0.00	0.00	819.02	819.02
Testikäyttäj2	67.94	2.00	1.53	0.00	0.00	29.91	804.36	834.27

## Kuva 12. Työntekijä raportit

Raportti työntekijöistä generoidaan annettujen päivämäärien välille. Raportti sisältää erittelyt tunneista, 50 % ylitöistä, 100 % ylitöistä, olosuhdelisistä, vuorolisistä ja palkat lisät eriteltynä sekä yhteensä. Tietyn työntekijän tunnit listaa tehdyt työtehtävät ja siitä näkee tarkemmin myös onko työtehtävää muokattu tai onko se jälkisyötetty.

## Laskentakohteen tunnit:

Aloitus	Lopetus	Lask.kohde	Tekijä	Tilaus	Työkohde	Työrahe	Tunnit	50%	100%	Vuor.	Olos.	Muok.	Jalk.	€	Lisär	Yht	
21.12.2010 06:09:13	21.12.2010 07:12:00	Testikäyttäj1	Testikäyttäj1	Testikäyttäj1	Testikäyttäj1	Testikäyttäj1	1.05	-	-	-	-	-	-	12.56	0.00	12.56	
21.12.2010 05:53:32	21.12.2010 07:13:15	Testikäyttäj1	Testikäyttäj1	Testikäyttäj1	Testikäyttäj1	Testikäyttäj1	1.33	-	-	-	-	-	-	15.73	0.00	15.73	
21.12.2010 07:12:53	21.12.2010 08:39:03	Testikäyttäj1	Testikäyttäj1	Testikäyttäj1	Testikäyttäj1	Testikäyttäj1	1.44	-	-	-	-	-	-	17.23	0.00	17.23	
21.12.2010 14:53:54	21.12.2010 14:50:00	Testikäyttäj1	Testikäyttäj1	Testikäyttäj1	Testikäyttäj1	Testikäyttäj1	-0.07	-	-	-	-	-	-	-0.78	0.00	-0.78	
21.12.2010 08:57:28	21.12.2010 14:53:44	Testikäyttäj1	Testikäyttäj1	Testikäyttäj1	Testikäyttäj1	Testikäyttäj1	5.94	-	-	-	-	-	-	71.25	0.00	71.25	
21.12.2010 08:23:56	21.12.2010 14:54:51	Testikäyttäj1	Testikäyttäj1	Testikäyttäj1	Testikäyttäj1	Testikäyttäj1	6.52	-	-	-	-	-	-	77.14	0.00	77.14	
														Yht:	193.13	0.00	193.13

## Laskentakohteiden tuntilistaus:

Laskentakohde	Tunnit	Yhtyo 50%	Yhtyo 100%	Olosuhdelisa	Vuorolisa	€	Lisär	Yht.
Testikäyttäj1	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Testikäyttäj2	21.63	0.00	0.00	0.00	0.00	23.85	0.00	23.85
Testikäyttäj3	42.71	0.00	0.00	0.00	0.00	509.77	0.00	509.77
Testikäyttäj4	4.00	0.00	0.00	0.00	0.00	40.00	0.00	40.00

## Kuva 13. Laskentakohde raportit

Laskentakohteiden listaus toimii samalla periaatteella kuin työntekijä raportit mutta kokonaisuromääriä laskiessa otetaan huomioon kunkin työtehtävän tehneen työntekijän tuntipalkka.

**Tilanneraportti:**

Tyonumero	Laskentakohde	Tilaus	Työkohte	Työvaihe	Tilanne
1050					Valmis
1050					Valmis
1050					Valmis
1050					Valmis

**Kuva 14. Tilanneraportti**

Tilanneraportti hakee tietokannasta viimeksi syötettyjen töiden perusteella jokaisen työkohteen viimeisimmäksi valmiiksi merkityt työvaiheet. Raporttia voidaan filteröidä niin että listataan vain tietyn laskentakohteen tilanne.

## 5. TESTAUS JA KÄYTTÖNOTTO

Projektin aikataulun hieman venyessä, varsinaiselle järjestelmätestaukselle ei löytynyt aikaa. Koska ohjelmaa oli tekemässä vain yksi henkilö, ei varsinaista testisuunnitelmaa testitapauksineen ollut järkevää tehdä.

Ohjelman osia kyllä testattiin moduuleittain sitä mukaa kun niitä tehtiin ja ne valmistuivat. Jokaista moduulia ja toimintoa testattiin tekovaiheessa erilaisilla arvoilla ja muilla keksityillä mahdollisimman monipuolisilla tilanteilla. Kun virhetilanteita havainnoitiin, ne korjattiin saman tien. Moduulikohtaista testausta jatkettiin yleensä niin pitkään että moduuli havaittiin virheettömäksi joten uutta osiota ei aloitettu ennen kuin edellinen oli valmis ja testattu. Löytyneet virheet liittyivät yleensä virheellisiin tietokannan hakulauseisiin tai matemaattisiin laskutoimituksiin.

Testauksen suhteellisesta vähyydestä johtuen käyttöönotto päätettiin suorittaa yrityksessä niin että kaikki työntekijät pysyisivät vielä vanhan järjestelmän piirissä, mutta muutama henkilö koekäyttäisi uutta järjestelmää samanaikaisesti. Näin saamme käyttökokemuksia muiltakin kuin järjestelmän ainoalta tekijältä sekä löydämme mahdolliset virhetilanteet ennen kuin uusi järjestelmä on pääasiallisessa käytössä. Itse tekijä ei sovellu testaajaksi kovin hyvin, koska hän ei välttämättä osaa käyttää järjestelmää väärin niin kuin normaali käyttäjä.

## **6. JATKOKEHITYS**

### **6.1 Ylläpito**

Järjestelmän ylläpitoon kuuluu tietokannan varmuuskopionti sekä tietokannan eheyden varmistaminen sekä mahdollisten virhetilanteiden korjaus.

Järjestelmän kivijalkana toimivan WAMP:n päivitys kuuluu myös ylläpidon piiriin. WAMP:iin tulevat päivitykset saattavat sisältää päivityksiä niin PHP:hen, MySQL:ään tai Apacheen. Ennen päivitystä uusi versio asennetaan ohjelmointikoneelle ja tuntikortti-järjestelmä testataan varmasti toimivaksi testiympäristössä ennen kuin sitä ollaan viemässä tuotantoympäristöön.

### **6.2 Ominaisuuksien kehittäminen**

Järjestelmän käyttöönottovaiheessa tuli heti ilmi muutamia kehitysideoita, joita ohjelmaan tullaan tekemään. Raportit osioon halutaan enemmän raporttipohjia sekä kätevämpi valikko, mistä voidaan valita mitä raporttia tahdotaan tulostaa.

### **6.3 Järjestelmän laajennus**

Tuntikortti-järjestelmän oli tarkoitus vain olla ensimmäinen osa suurempaa projektia, jossa koko yrityksen tuotannonohjausjärjestelmä tullaan uusimaan. Laajennusprojekti on jo määrittely- ja suunnitteluasteella.

## 7. POHDINTA

### 7.1 Onnistumiset

Siihen nähden, että tämä oli ensimmäinen tämän kokoluokan projekti jonka vedin alusta loppuun, projektin läpiviennissä onnistuttiin kohtuullisen hyvin.

Järjestelmään saatiin toteutettua ominaisuudet mitä haluttiin.

Tuntikortin dynaaminen käyttöliittymä oli suurin onnistuminen, sillä sen lähdin toteuttamaan tekniikalla, josta ei ollut mitään aikaisempaa kokemusta ja siitä saatiin kuitenkin ajan kanssa erittäin toimiva kokonaisuus.

### 7.2 Haasteet ja ongelmat

Määrittelyn laiminlyönti ja liian paljon vanhaan Excel-pohjaan nojautuminen tuotti projektiin omat vaikeutensa.

Web-pohjaisissa tietojärjestelmissä yleisesti on kirjautuminen ensimmäinen toimenpide ennen kuin mitään saadaan käytettyä. Määrittelyssä emme ottaneet kaikkea huomioon joten alun perin menimme heti startissa väärään suuntaan. Toteutin järjestelmään kirjautumisen niin että jokainen käyttäjä kirjautuu ennen etusivua jotta he voisivat sitten vapaasti tehdä haluamansa ollessaan kirjautuneena. Ensimmäistä versiota asiakkaalle näytettäessä saimme heti lähteä muuttamaan sen niin että käyttäjän yksilöiminen kysytään tuntikortissa työtä aloittaessa ja työtä lopetettaessa sekä muokattaessa.

Projektin yhdeksi haasteeksi nousi uusien tekniikkojen opettelu täysin nollasta. Käyttöliittymän dynaamisuuden toteuttamiseksi tarvitsin opetella jQuery ja Ajax aivan nollista. Koulussa tarjottu olemattomaksi jäänyt JavaScript oppi ei auttanut juurikaan.

Raporttien laskentakaavat olivat myös ongelmallisia. Ylitöiden osuudet 50 %:n ja 100 %:n osalta oli melko haastava saada laskemaan täysin oikein.

Laskentakohteiden euromäärien laskussa myös jouduttiin ottamaan ylityöt sekä käyttäjien henkilökohtaisen tuntipalkan osuudet, tämä tuotti myös hankaluuksia.

### **7.3 Tavoitteiden saavuttaminen**

Lähtötilanteen toiminnot saatiin tehtyä toivomusten mukaisesti. Projektin käyttöönottovaiheessa tosin saatiin heti uusia kehitysideoita. Mm. erilaisia raportteja toivottiin lisää ja raporttien tulostuksen käyttöliittymään keksittiin parannuksia.

Omista tavoitteista saavutettiin ainakin kokonaisen projektin läpiviennin näkeminen. Ajax, JavaScript, jQuery oppia tuli projektin mukana rutkasti.

### **7.4 Loppusanat**

Opinnäytetyön läpivienti oli erittäin opettava ja haasteellinen kokemus. Erityisesti projektissa sai arvokasta kokemusta miten projekteja vedetään käytännössä oikeassa tuotantoympäristössä. Seuraavaa samankaltaista projektia lähtiessäni viemään läpi, panostaisin alussa määrittelyyn huomattavasti enemmän. Sillä saataisiin kitkettyä paremmin pois tässä projektissa välillä ilmenneitä hieman tulkinnan varaisiksi jääneitä kohtia kuten raportteja. Myös tilanteet jossa, tein oman näkemyksen mukaan ja asiakkaalle näytettäessä se ei edes ollut ollenkaan sinnepäin, saataisiin vähemmälle tarkemmalla määrittelyllä. Tätä kautta säästettäisiin aikaa ja kokonaistyömäärää.



## LÄHTEET

Asleson, R. & Scutta, N.T. 2007. *Rakenna dynaamisia verkkosivuja*. Readme.fi.

Sarja, J. 2010. *AV-Viestinnän keittokirja: WAMP* [verkkojulkaisu]. [viitattu 23.11.2010]. Saatavissa:  
<http://www.avkymppi.net/joomla/wamp.html>

Chaffer, J. & Swedberg, K. 2009. *Learning jQuery 1.3*. Birmingham: Packt Publishing Ltd.

Kolehmainen, K. 2006. *PHP & MySQL Teoriasta käytäntöön*. Readme.fi

Lehtimäki, T. 2006. *Ohjelmistoprojektit käytännössä*. Readme.fi

Rantala, A. 2005. *Web-ohjelmointi*. Docendo Finland Oy

Wikipedia. Vapaa tietosanakirja 2010. MD5 [verkkojulkaisu]. [viitattu 15.11.2010]. Saatavissa:  
<http://fi.wikipedia.org/wiki/MD5>

Wikipedia. Vapaa tietosanakirja 2010. PHP [verkkojulkaisu]. [viitattu 17.11.2010] Saatavissa:  
<http://fi.wikipedia.org/wiki/PHP>