

Web-pohjainen työkalenteri

Joonas Roivainen

Opinnäytetyö

Koulutusala Tekniikan ja liikenteen ala	
Koulutusohjelma Tietotekniikan koulutusohjelma	
Työn tekijä(t) Joonas Roivainen	
Työn nimi Web-pohjainen työkalenteri	
Päiväys 29.5.2012	Sivumäärä/Liitteet 28/8
Ohjaaja(t) lehtori Sami Lahti, lehtori Jussi Koistinen	
Toimeksiantaja/Yhteistyökumppani(t) Unelmapaketti Ay, yrittäjät Jari Laaksonen, Eeva Väisänen	
<p>Tiivistelmä</p> <p>Opinnäytetyön aiheena oli suunnitella ja toteuttaa web-pohjainen työkalentersovellus Unelmapaketti Ay:öön kuuluvalla Kodin Apujoukoille. Sovelluksen tarkoituksena oli nopeuttaa yrityksen työtehtävien hallintaa ja helpottaa työntekijöiden palkanlaskentaa sekä asiakkaiden laskutusta. Tavoitteena oli saada sovelluksesta toimiva, helppokäyttöinen sekä mahdollisimman selkeä. Työprosessin alkuvaiheessa ohjelmaa suunniteltiin yhdessä toimeksiantajan kanssa. Ohjelmointivaiheessa työtä tarkasteltiin yhdessä toimeksiantajan kanssa useita kertoja eri toimintojen valmistuttua.</p> <p>Sovelluksen ohjelmoinnissa käytettiin pääasiassa PHP-ohjelmointikieltä, jonkin verran Javascriptiä. Sen tietokantana oli MySQL ja siinä oli neljä eri taulua. Lisäksi sovelluksessa käytettiin Javascriptin jQuery-kirjastoa. Ohjelmointi suoritettiin Netbeans-ohjelmointiympäristöllä.</p> <p>Työkalenteriin sisältyi seitsemän eri verkkosivua. Esimiehen näkymän pääsivuna oli työkalenteri, jossa näkyy kaikkien työntekijöiden työtehtävät valitulta aikaväliltä. Työntekijöitä varten sovelluksessa oli viikkokalentersivu, josta työntekijät näkevät omat viikoittaiset työtehtävänsä. Palkanlaskenta-sivulla näkyvät lisätiedot valitun työntekijän työtehtävistä ja vapaapäivistä. Laskutus-sivulla nähdään valitulle asiakkaalle tehdyt työtehtävät ja niihin liittyvät tiedot. Työntekijät-sivulla hallitaan työntekijöitä ja heidän tietojaan. Asiakkaat-sivulla nähdään asiakkaiden yhteystiedot. Omat tiedot -sivulla voidaan muuttaa omia tietoja sekä salasanaa.</p> <p>Ohjelmaa testattiin sekä koko ohjelmointivaiheen ajan että sen valmistuttua kokonaisuutena. Työkalenterista tuli selkeä ja helppokäyttöinen, koska työtehtävät voitiin lisätä suoraan kalenteriin. Toimivuudesta ei ole testauksien ja korjauksien jälkeen löydetty virheitä.</p>	
Avainsanat kalenteri, työkalenteri, PHP, verkkosovellus	

Field of Study Technology, Communication and Transport			
Degree Programme Degree Programme in Information Technology			
Author(s) Joonas Roivainen			
Title of Thesis Web-Based Work Calendar			
Date	29 May 2012	Pages/Appendices	28/8
Supervisor(s) Mr Sami Lahti, Lecturer and Mr Jussi Koistinen, Lecturer			
Client Organisation/Partners Unelmapaketti Ay, Mr Jari Laaksonen, Entrepreneur and Mrs Eeva Väisänen, Entrepreneur			
<p>Abstract</p> <p>The topic of this final year project was to plan and implement a web-based work calendar application for Kodin Apujoukot which belongs to a company called Unelmapaketti Ay. The purpose of this application was to speed up the handling of jobs and to ease the calculation of salaries and billing. The goal of the application was to make it easy to use and clear. It was also supposed to be working flawlessly. The application was planned together with the commissioner. After the features were implemented in the development phase the program was viewed by the commissioner.</p> <p>The program was developed mainly with the PHP programming language but Javascript was also used to some extent. The database was made with MySQL and it contains four different tables. In addition to that jQuery library was used with Javascript. The application was developed with the Netbeans integrated development environment.</p> <p>The Work Calendar consists of seven different web pages. The Work Calendar page was the main page for the manager. The page shows all jobs from chosen times for each employee. The program also has a page for employees called Week Calendar page. On that page the employees can see their weekly work assignments. The Salaries page contains more information about chosen employee's work assignments and holidays. The Billing page shows work assignments done for a chosen client and information about them. Employees are managed on the Employees page and the Clients page contains contact information about clients. Employees can change their information on the My Account page.</p> <p>Testing was done constantly in the development phase. The program was also tested completely after it was done. The Work calendar became clear and it was easy to use because work assignments could be added straight into the calendar. There have been no errors after testing and correction.</p>			
<p>Keywords calendar, work calendar, PHP, web application</p>			

Alkusanat

Tämä opinnäytetyö tehtiin keväällä 2012 Unelmapaketti Ay:lle. Työn ohjaajina toimivat lehtorit Sami Lahti ja Jussi Koistinen Savonia-ammattikorkeakoulusta. Haluan kiittää ohjaajia opinnäytetyön ohjauksesta sekä tarkastamisesta. Erityisesti haluan kiittää Unelmapaketti Ay:n yrittäjiä, Jari Laaksosta sekä Eeva Väisästä yhteistyöstä sekä aiheen tarjoamisesta.

Hyvinkäällä 19.5.2012

Joonas Roivainen

SISÄLTÖ

1	JOHDANTO	8
2	TYÖN LÄHTÖKOHDAT.....	9
3	KÄYTETYT TEKNIIKAT JA TYÖKALUT	10
3.1	PHP	10
3.2	JavaScript.....	10
3.3	Netbeans IDE	11
3.4	WampServer.....	11
3.5	WinSCP.....	11
4	OPINNÄYTETYÖPROSESSI	12
4.1	Määrittely.....	12
4.2	Suunnittelu.....	12
4.3	Toteutus	13
5	TIETOKANTA	14
6	TYÖKALENTERI.....	16
6.1	Sisäänkirjautuminen ja navigointi	16
6.2	Viikkokalenteri	17
6.3	Työkalenteri-sivu.....	18
6.4	Työntekijöiden hallinta ja Asiakkaat-sivu	20
6.5	Omat tiedot -sivu	21
6.6	Palkanlaskenta.....	21
6.7	Laskutus	23
7	TESTAUS	24
8	TYÖN ARVIOINTI	26
9	POHDINTA.....	27
	LÄHTEET	28

LIITTEET

Liite 1 Projektisuunnitelma

Termit ja lyhenteet

HTML	Hypertext Markup Language on pääasiassa verkkosivuilla käytettävä merkkäuskieli.
jQuery	Javascript-kirjasto, jolla saadaan parannettua käyttäjän ja sivuston välistä vuorovaikutusta
jQuery UI	jQueryn avulla toteutettu avoimen lähdekoodin kirjasto, joka sisältää erilaisia käyttöliittymä komponentteja.
Ajax	Asynchronous Javascript and XML on joukko tekniikoita, joiden avulla verkkosivuista saadaan dynaamisempia. Tekniikoiden avulla on mahdollista hakea tietoja palvelimelta ilman sivun uudelleenlatausta
Regular Expression	Merkkijono, jolla kuvataan hakukaavaa. Sen avulla voidaan esimerkiksi etsiä tekstin joukosta erilaisia sanoja tai vaikka sähköpostiosoitteita.
CSS	Cascading Style Sheets on kuvauskieli, jolla kuvataan miten HTML-elementit näkyvät käyttäjälle.
Notepad++	Ilmainen tekstieditori.
SFTP	SSH File Transfer Protocol on turvalliseen tiedonsiirtoon tarkoitettu protokolla.

1 JOHDANTO

Opinnäytetyön aiheena on suunnitella ja toteuttaa internetissä toimiva työkalentersovellus Kodin Apujoukoille, joka on osana vuonna 2006 perustettua Unelmapaketti Ay:tä. Kodin Apujoukot on vuodesta 2004 toiminut kodinhoitopalveluyritys, jonka päätoimialueena ovat Hyvinkää, Nurmijärvi sekä niiden lähialueet. Unelmapaketti Ay:öön kuuluu myös tietotekniikkapalveluja tarjoava Uudenmaan tietotiimi. Työkalenterin on tarkoitus korvata Kodin Apujoukkojen aiemmin käyttämä Microsoft Excel -pohjainen työtuntilista.

Opinnäytetyön aihe sai alkunsa yrityksen halusta uudistaa töiden organisointiin käytettävää järjestelmää, koska vanhan järjestelmän käytössä oli hankaluuksia. Ongelmana Excel-pohjaisessa työlistassa oli turha käsityö, koska siihen jouduttiin kirjoittamaan kaikki tiedot käsin. Tämä on johtanut siihen, ettei työtuntilistassa ole ollut kaikkia hyödyllisiä tietoja. Työlistaa ei voinut myöskään jakaa työntekijöille internetissä, mistä seurasi esimiehelle turhaa tulostustyötä. Excel-pohjaisessa työlistassa oli ongelmana myös käyttäjän inhimilliset virheet. Työtuntilista ei myöskään ollut kovin selkeä, joten sen käyttäminen ei ollut tehokasta, vaan monimutkaista ja hajanaista. Näiden ongelmien vuoksi Kodin Apujoukot halusi korvata vanhan Excel-pohjaisen työtuntilistan helppokäyttöisemmällä sovelluksella.

Työn tavoitteena on saada aikaan toimiva web-kalentersovellus Kodin Apujoukoille. Tavoitteeseen pyritään tekemällä sovelluksesta mahdollisimman helppokäyttöinen ja selkeä. Opinnäytetyön tarkoituksena on nopeuttaa yrityksen työtehtävien hallintaa, helpottaa työntekijöiden palkanlaskentaa sekä asiakkaiden laskutusta.

Tässä raportissa kerrotaan ensin työn lähtökohdista ja sovelluksessa käytetyistä tekniikoista. Tämän jälkeen perehdytään projektin vaiheisiin ja sitten esitellään sovelluksen tietokanta sekä toteutettu sovellus. Lopuksi kerrotaan, miten sovellusta testattiin ja miten työ onnistui.

2 TYÖN LÄHTÖKOHDAT

Järvenpään ja Hännisen (2011, 22) mukaan yksi keino liiketoiminnan tuottamisen parantamiseen on uuteen tietotekniikkaan investoiminen. Unelmapaketti Ay halusi jo ennen opinnäytetyöstä sopimista panostaa uuteen tietotekniikkaan osana toiminnan tehostamista. Yrityksellä oli toive sille räätälöidystä sovelluksesta, joka kuitenkin pystyttäisiin tarvittaessa muokkaamaan samankaltaisten yritysten tarpeisiin sopivaksi, jotta sovelluksella olisi jälleenmyyntiarvoa.

Erilaisia työtä hidastavia tai haittaavia tekijöitä kutsutaan tuottavuusjarruiksi. Kodin Apujoukkojen aiemmin käyttämässä Microsoft Excel -pohjaisessa järjestelmässä oli tuottavuusjarruja erityisesti käytettävyydessä ja selkeydessä. (Järvenpää & Hänninen 2011, 11–12.) Näistä tuottavuusjarruista haluttiin päästä eroon tekemällä sovelluksesta mahdollisimman havainnollinen ja helppokäyttöinen.

Yritysten käyttöön tarkoitetuista tietotekniikan sovelluksista suurin osa on tietojärjestelmiä. Näistä tietojärjestelmistä suurinta ryhmää voidaan luonnehtia operatiivisiksi bisnesjärjestelmiksi, joiden tavoitteena on toiminnan tehostaminen automatisoinnilla, materiaalin käytön sekä varastoinnin vähentäminen ja turhan käsityön karsiminen. (Tiirikainen 2010, 13–14.) Tällaisen tietojärjestelmän ominaisuuksia Unelmapaketti Ay tavoitteli työkalenterisovellukseen. Turhaa käsityötä haluttiin karsia esimerkiksi automatisoimalla tietojen käsittelyä työtuntien ja vapaapäivien laskemisella sekä sillä, että työkalenterin työtehtävät näkyisivät sekä esimiehelle että työntekijöille heille sopivissa kalentereissa.

Operatiivisten bisnesjärjestelmien lisäksi Tiirikainen (2010, 14) on jaotellut tietojärjestelmiä myös toiseen suureen ryhmään, johtamisen tietojärjestelmiksi. Sovelluksen tulisi palvella myös johtamista ja yrityksen tiedonhallintaa, joten sen piti sisältää ominaisuuksia johtamisen tietojärjestelmistä. Näihin johtamisen tietojärjestelmien ominaisuuksiin kuuluu muun muassa töiden delegointi, jota työkalenterisovelluksen avulla haluttiin parantaa. Johtamisen tietojärjestelmät on yleensä räätälöitävä yrityksen tarpeisiin, mikä aiheuttaa yritykselle suuria kustannuksia (Tiirikainen 2010, 17). Yritys halusi välttää suuret kustannukset käyttämällä hyväkseen mahdollisuutta tehdä sovellus opinnäytetyönä.

3 KÄYTETYT TEKNIIKAT JA TYÖKALUT

Sovelluksen pääasialliseksi ohjelmointikieleksi valittiin PHP ja tietokannaksi MySQL, koska ohjelman haluttiin toimivan internetissä ja toimeksiantajalla on käytössään palvelin, josta löytyy tuki sekä PHP:lle että MySQL:lle. Valintaan vaikuttivat myös opinäytetyön tekijän aiempi kokemus PHP-ohjelmoinnista sekä sen hyvä soveltuvuus web-ohjelmointiin. Käyttäjän ja sivuston välistä vuorovaikutusta parantamaan valittiin Javascript-skriptikieli sekä sitä käyttävä jQuery-kirjasto.

Sovelluksen toteutuksen alussa testattiin lukuisia ohjelmointiympäristöjä sekä tekstieditoreja, joilla itse ohjelmointi suoritettaisiin. Ohjelmointia varten tarvittavan ohjelman vaatimuksena oli, että siinä olisi tuki PHP:lle, jQuerylle sekä Javascriptille. Ohjelman piti olla myös maksuton. Lopulta ohjelmaksi valittiin Netbeans, koska sen käyttäminen oli helppoa ja siitä löytyi tarpeellinen tuki vaadituille ohjelmointikielille. Tiedostojen siirto-ohjelmaksi toimeksiantajan palvelimelle valittiin WinSCP, koska siitä löytyi tuki SFTP-protokollan avulla tehtävään tiedonsiirtoon.

3.1 PHP

PHP (Hypertext Preprocessor) on palvelinpuolen ohjelmointiin tarkoitettu skriptikieli. Palvelimelle asennettava PHP-ohjelmisto tulkaa PHP-ohjelman, ja ohjelman luoma tulos palautuu asiakkaalle esitettäväksi. PHP-ohjelma on .php-tunnuksella erottuva tekstimuotoinen lähdekooditiedosto. PHP on avointa lähdekoodia ja sitä käytetään usein Apache-palvelimen, MySQL-tietokannan sekä Linux-käyttöjärjestelmän kanssa. PHP-koodi sisällytetään aina alku- ja lopputagien sisälle. (Kolehmainen 2006, 3–4.) PHP:tä käytetään yli 25 miljoonalla verkkosivulla (BuiltWith).

3.2 JavaScript

Javascript on Netscapen vuonna 1995 julkaisema skriptikieli, jota käytetään käyttäjän ja sivuston vuorovaikutuksen lisäämiseen. Javascript on tulkattava kieli, ja se vaatii toimiakseen Javascript-tulkin käyttäjän sovelluksessa. (Peltomäki & Nykänen 2006, 90, 97.) Javascriptiä käytetään maailmanlaajuisesti miljoonilla verkkosivuilla ja palvelinsovelluksissa (Mozilla Developer Network 1). Javascriptiä käytetään sisällytettynä

muihin sovelluksiin tai tuotteisiin, sillä sen käyttö ainoana ohjelmointikielenä ei ole järkevää (Mozilla Developer Network 2).

3.3 Netbeans IDE

Netbeans IDE (Integrated Development Environment) on alun perin Java-ohjelmointiin suunniteltu ohjelmointiympäristö. Netbeans oli alun perin Java-ohjelmointikielillä toteutettu opiskelijaprojekti. Nykyään Netbeansissä on tuki muun muassa HTML- ja CSS-kuvauskielille sekä PHP- ja Javascript-ohjelmointikielille. Ohjelmointiympäristöstä löytyy myös tuki Javascriptin jQuery-kirjastolle. Netbeansin PHP-tuki sisältää esimerkiksi koodin automaattisen täyttämistoiminnon ja PHP:n sisältämistä funktioista näytetään, mitä kyseinen funktio tekee. (Netbeans.)

3.4 WampServer

WampServer on Windows-käyttöjärjestelmällä toimiva avoimen lähdekoodin web-kehitysympäristö. WampServer sisältää muun muassa PHP-tulkin, Apache-palvelimen, MySQL:n sekä phpMyAdmin-ohjelman. Sillä voidaan luoda verkkosovelluksia, joissa käytetään Apachea, PHP:tä sekä MySQL-tietokantaa. (WampServer.)

Tässä sovelluksessa WampServeriä käytettiin paikallisena testipalvelimena PHP-sivujen näyttämiseen sekä sen sisältämän phpMyAdmin-ohjelman avulla luotiin sovelluksen MySQL-tietokanta.

3.5 WinSCP

WinSCP on paikallisen ja etätietokoneen väliseen tiedonsiirtoon tarkoitettu avoimen lähdekoodin sovellus. Sovelluksen avulla siirrettiin Työkaleri-sovellukseen kuuluvat tiedostot SFTP-protokollan avulla Uudenmaan tietotiimin palvelimelle. Sen avulla pystyttiin myös muokkaamaan palvelimella olevia tiedostoja. Tiedostot avattiin palvelimelta Notepad++-sovellukseen, jolla tehtiin pieniä muutoksia koodiin.

4 OPINNÄYTETYÖPROSESSI

Työ toteutettiin keväällä 2012 etätyönä. Sovelluksen toteuttamisen etätyönä mahdollistivat ip-osoitteen avulla otettava etäyhteys Uudenmaan tietotiimin palvelimeen sekä ilmaiset ohjelmat, joilla sovellus voitiin ohjelmoida. Projektiin kuului neljä työvaihetta, jotka olivat määrittely, suunnittelu, toteutus sekä testaus. Määrittelyvaiheessa määriteltiin sovellukseen tulevat toiminnot sekä vaatimukset. Suunnitteluvaiheessa tehtiin tarvittavat suunnitelmat projektin läpivientiä varten. Toteutusvaiheessa ohjelmoitiin sovellus suunnitelman mukaan. Testausvaiheessa etsittiin sovelluksessa esiintyvät virheet ja korjattiin ne. Tässä luvussa kerrotaan sovelluksen määrittelystä, suunnittelusta ja toteutusvaiheen kulusta. Testauksesta kerrotaan luvussa 6.

4.1 Määrittely

Määrittelyvaiheessa sovittiin toimeksiantajan kanssa sovellukseen tulevat toiminnot sekä keskusteltiin sovelluksen muista vaatimuksista. Näitä vaatimuksia olivat käyttöliittymän selkeys, sovelluksen helppokäyttöisyys ja toimivuus. Lisäksi sovelluksesta haluttiin helposti muokattava, jotta sillä olisi jälleenmyyntiarvoa. Sovituista toiminnoista yksityiskohtaisimmat kirjattiin muistiin. Myöhemmin sovellukseen tulevista toiminnoista tehtiin Microsoft Excelin avulla taulukko, jossa näkyi niiden kuvaus sekä valmiusasteet. Tämän taulukon päivittäminen jäi kuitenkin lopulta tekemättä, koska siitä ei ollut oleellista hyötyä sen jälkeen, kun ohjelmassa alkoi olla valmiita toimintoja ja tiedettiin tarkkaan, mitä sovelluksesta puuttuu.

4.2 Suunnittelu

Projektin suunnitteluvaiheessa suunnittelun lähtökohtana olivat toimeksiantajan sovellukselle antamat vaatimukset. Sovellusta suunniteltiin käyttöliittymästä tehtyjen mallikuvien avulla, jotta oltaisiin samaa mieltä sen rakenteesta toimeksiantajan kanssa. Nämä mallikuvat toteutettiin aluksi Microsoftin Visual Studiolla, koska sillä pystyi helposti lisäämään käyttöliittymässä yleisesti esiintyviä komponentteja, kuten nappeja. Nopean toteutuksen vuoksi ohjelmalla oli helppo visualisoida mahdollista käyttöliittymää. Tämän jälkeen mallikuvat toteutettiin vielä HTML:n ja CSS:n avulla, koska sovellukseen tulevat sivut näkyisivät näiden tekniikoiden avulla. Ensin päätettiin to-

teuttaa mallikuvan kaltainen kalenteri, johon voidaan jollakin tapaa lisätä työtehtäviä sekä erillinen viikkokalenteri, josta nähdään viikoittaiset työt. Tämän jälkeen tehtiin projektisuunnitelma koko opinnäytetyöprojektistä. Suunnitelma sisälsi arvion projektin aikataulusta ja kuvauksen työkalenterista. Projektisuunnitelmassa kerrottiin myös projektiin liittyvät riskit sekä nimettiin projektiin liittyvät henkilöt. Ohjelmointia suunniteltiin ennen ohjelmointivaihetta sekä sen aikana. Tässä suunnittelussa otettiin huomioon erityisesti vaatimus sovelluksen helpposta muokattavuudesta. Ennen ohjelmoinnin aloittamista suunniteltiin ja määriteltiin osa tarvittavista sivuista sekä luokista.

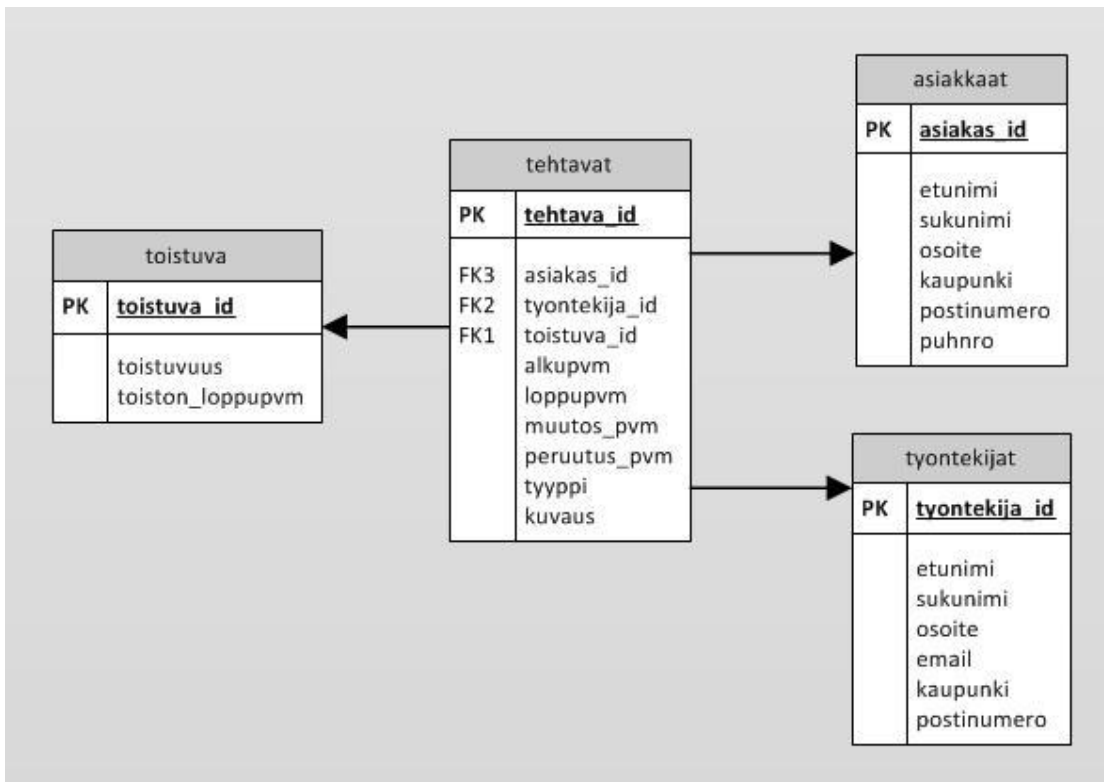
4.3 Toteutus

Projektin toteutusvaihe alkoi työkalenterin pohjan rakentamisella sekä viikkokalenterin tekemisellä. Kalenterit toteutettiin suunnitteluvaiheessa tehtyjen mallikuvien perusteella. Näiden jälkeen toteutettiin asiakkaiden sekä työntekijöiden hallinnat, jotta pystyttiin tekemään työkalenterissa oleva työtehtävien hallinta. Nämä sivut esiteltiin niiden valmistuttua toimeksiantajalle ja sovittiin niihin tehtävistä korjauksista sekä sovellukseen tehtävistä lisäyksistä. Uusiin lisäyksiin kuuluivat palkanlaskenta- ja laskutus-sivut sekä jo aiempaan versioon sovittu toistuvien tehtävien toteuttaminen. Lisäyksien ja korjauksien ohjelmoinnin jälkeen paranneltiin sovelluksen ulkoasua ja siirryttiin testausvaiheeseen.

5 TIETOKANTA

Sovelluksen tietokantaohjelmistoksi valittiin MySQL, koska Uudenmaan tietotiimillä oli oma Unix-palvelin varustettuna Apache-palvelinohjelmalla. Tietokantavalintaan vaikuttivat myös sen maksuton käyttö sekä yhteensopivuus PHP-ohjelmointikielen kanssa. Tietokantataulut tehtiin aluksi phpMyAdmin-ohjelmalla paikallisesti, minkä jälkeen ne siirrettiin Uudenmaan tietotiimin palvelimelle.

Tietokantatauluina sovelluksessa ovat asiakkaat, työntekijät, toistuvat tehtävät sekä työtehtävät. Tietokantakaavio näkyy kuvassa 1. Asiakkaat- sekä työntekijät-taulut kopioitiin toimeksiantajan toisesta sovelluksesta. Työntekijät-taulusta tehtiin myös toinen samanlainen taulu, jossa oli lisäksi tunnus- ja salasanakentät. Tämä taulu tehtiin, koska toimeksiantajan työntekijät-taulu kopioitiin tietyin väliajoin toisesta tietokannasta, ja näin ollen siihen tehdyt muutokset poistuisivat. Toimeksiantajan tauluja ei kuitenkaan saatu käyttöön heti sovelluksen alkuvaiheessa, joten ensin toteutettiin väliaikaiset työntekijä- ja asiakkaat-taulut, jotta sovelluksen toimintaa voitiin testata.



KUVA 1. Tietokantakaavio

Asiakkaat-taulusta sovelluksessa käytetään asiakkaiden nimi-, osoite- sekä puhelinnumerotietoja. Työntekijät-tauluun tallennetaan työntekijän osoitetiedot sekä tunnus ja salasana, joilla kirjaudutaan sovellukseen sisään. Salasana tallennetaan tietokantaan tiivisteenä, joka on tehty MD5-algoritmilla. Ennen MD5-tiivisteiden tekemistä salasana suolataan eli siihen lisätään itse määrittelemä merkkijono. Suolauksella pyritään vaikeuttamaan MD5-salauksen purkamista, jos tietokantatiedot päätyisivät väärin käsiin.

Tehtävät-tauluun tallennetaan työtehtävän tyyppi, joka kertoo, onko kyseessä työtehtävä vai vapaapäivä. Työtehtävästä tallennetaan tauluun myös kuvaus sekä neljä eri päivämääräkenttää. Päivämäärätietokenttinä ovat tehtävän alku- ja loppupäivämäärät sekä muutos- ja peruutuspäivämäärät. Yksittäisellä työtehtävällä on myös neljä eri tunniste-kenttää: asiakas_id, tyontekija_id, toistuva_id sekä tehtava_id. Nämä tunniste-kentät kertovat, mitkä eri taulujen tietueista liittyvät tehtävät-tauluun. Esimerkiksi asiakas_id-kenttä on sekä tehtävät- että asiakkaat-tauluissa, ja se kertoo, kuka asiakas liittyy kyseiseen tehtävään. Toistuvat-tauluun tallennetaan toistuva_id, toistuvuus sekä toiston loppupäivämäärä. Toistuvuus kertoo, miten usein tehtävä toistuu eli onko se yhden, kahden vai neljän viikon välein toistuva. Toiston loppupäivämäärä kertoo, milloin tehtävän toistuvuus loppuu.

6 TYÖKALENTERI

Tässä luvussa esitellään opinnäytetyönä tehdyn työkalenterin ominaisuuksia ja sitä, miten ne on toteutettu. Työkalenteri on töiden suunnitteluun ja työntekijöiden seurantaan tarkoitettu web-sovellus, johon kuuluu seitsemän eri verkkosivua. Esimies pystyy sovelluksen avulla seuraamaan ja luomaan työntekijöiden työtehtäviä sekä katsomaan tietoja asiakkaista sekä heille tehdyistä töistä laskuttamista varten. Sovelluksen avulla voidaan myös seurata työntekijöiden tekemiä työtunteja ja vapaapäiviä, jotta työntekijöille voidaan helposti laskea kuukausipalkka. Esimies voi myös hallita työntekijöiden tietoja. Työntekijät näkevät sovelluksen viikkokalenteri-sivulla omat työtehtävänsä sekä niihin liittyvät tiedot.

Sovelluksen kaikissa päivämäärän valinta -syöttökentissä on käytetty jQuery UI:n Datepicker-lisäosaa. Datepicker-lisäosa on päivämäärän valintaa helpottava kalenteri, josta voidaan kalenterissa olevaa päivävystä klikkaamalla valita syöttökenttään haluttu päivämäärä. Tehtävän sekä työntekijän lisäyksessä on käytetty jQuery UI:n Dialog-lisäosaa. Dialog on iframe-ikkuna, joka voidaan määrittää avautumaan esimerkiksi jotakin kohtaa klikattaessa.

6.1 Sisäänkirjautuminen ja navigointi

Työkalenterissa on pakollinen sisäänkirjautuminen, josta sisäänkirjautuneet työntekijät ohjataan viikkokalenteri-sivulle ja esimiehet työkalenteri-sivulle. Sisäänkirjautuminen tapahtuu sovelluksen sisäänkirjautumis-sivulla, jolla oleviin tunnus- ja salasana-kenttiin syötetään oma tunnus sekä salasana. Painamalla sivulla olevaa Lähetä-nappia tiedot lähetetään tarkistussivulle, jolla tarkistetaan ovatko tunnus ja salasana tietokannassa. Sisäänkirjautuessa tallennetaan evästeeseen tieto siitä, että käyttäjä on kirjautunut onnistuneesti. Kaikilla sovelluksen sivuilla tarkistetaan, onko käyttäjä kirjautunut sisälle eli onko tieto kirjautumisesta tallennettu evästeeseen. Kirjautumattomat käyttäjät ohjataan sisäänkirjautumis-sivulle.

Navigointi ohjelmassa tapahtuu sivujen vasemmassa laidassa olevien linkkien avulla. Navigointipalkkiin voidaan lisätä uusien sivujen linkkejä syöttämällä uuden sivun osoite sekä linkin nimi ohjelman asetustiedostossa olevaan taulukkoon, josta sovellus luo linkit navigointipalkkiin. Navigointilinkit näytetään allekkain sivun vasemmassa laidas-

sa, koska Työkaleri-sivulla haluttiin näyttää mahdollisimman monta tehtävää allekkain, ja vaakasuorassa olevat linkit olisivat vieneet liikaa tilaa sivulla olevalta työkalenterilta.

6.2 Viikkokalenteri

Viikkokalenteri on työntekijöille tarkoitettu sivu, josta he näkevät omat työtehtävänsä. Sivulla oleva kalenteri näyttää nimensä mukaisesti yhden viikon kerrallaan, koska se on tällöin sopivan kokoinen tulostettavaksi. Kalenterin selailu tapahtuu edellinen viikko ja seuraava viikko -linkkien avulla. Linkkiä klikattaessa ohjelma lähettää GET-parametreilla samalle sivulle seitsemän päivän päästä olevan päiväyksen, josta PHP:n date-funktion avulla lasketaan kyseisen viikon maanantain päivämäärä.

Kalenterissa listataan kirjautuneena olevan työntekijän kaikki työtehtävät ja vapaat haetulta viikolta (kuva 2). Kalenterissa näkyvät kaikki arkena olevat Suomen viralliset vapaapäivät kyseiseltä viikolta. Listassa näytetään työtehtävän tietoina asiakkaan nimi ja yhteystiedot sekä tehtävälle mahdollisesti annettu kuvaus. Kalenterisivulla on myös tulostimen kuva, jota klikkaamalla piilotetaan sivulla olevat navigointilinkit, ja viikkokalenteri on siten valmis tulostettavaksi.

<< Huhti-Toukokuu 2012 Viikko 18 >>

Maanantai 30.4	Tiistai 1.5	Keskiviikko 2.5	Torstai 3.5	Perjantai 4.5	Lauantai 5.5	Sunnuntai 6.5
<p>6:30-10:15 Asiakas Aapo 040 000 0000 Hyvinkää</p> <p>11:00-13:00 Asiakas1 Testi Pyörönkaari 666 Kuopio</p> <p>14:00-17:00 Ryökäle Risto 066 555 4444 Korvatie 15 Korvatunturi</p>	Vappu	<p>7:30-11:00 Asiakas Aapo 040 000 0000 Hyvinkää</p> <p>11:15-13:35 Ryökäle Risto 066 555 4444 Korvatie 15 Korvatunturi</p> <p>14:00-17:00 Asiakas1 Testi Pyörönkaari 666 Kuopio</p>	<p>6:00-10:00 Ryökäle Risto 066 555 4444 Korvatie 15 Korvatunturi</p> <p>10:00-14:00 Asiakas1 Testi Pyörönkaari 666 Kuopio</p>	Palkaton Vapaa		

KUVA 2. Viikkokalenteri-sivu

6.3 Työkaleri-sivu

Työkaleri-sivulla nähdään kaikkien työntekijöiden työtehtävät ja vapaapäivät haetulta aikaväliltä. Työlistan tehtävät haetaan MySQL-tietokannasta, ja ne tulostetaan näytölle PHP:llä tehdyn kalenteri-luokan avulla. Aikavälin hakeminen tapahtuu joko syöttämällä päivämäärät sivun ylärivillä oleviin tekstikenttiin tai klikkaamalla tekstikenttää, josta avautuu jQuery UI:n päivämäärän valintakalenteri ja painamalla tämän jälkeen Hae-nappia (kuva 3). Oletuksena kalenterissa näytetään tämän hetkisen kuukauden työlistä. Kuten kuvasta nähdään, listan viimeisellä rivillä näkyvät myös työtehtävien yhteenlasketut työtunnit. Työvuorolistassa näytetään punaisella värillä Suomen viralliset vapaapäivät sekä sunnuntait.

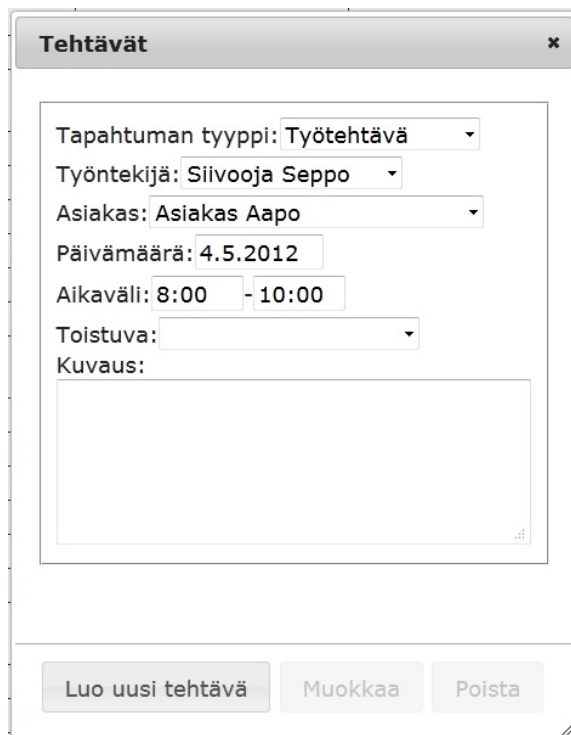
<< Alkupäivämäärä:
Loppupäivämäärä: >>

	Kehveli Keijo	Rähmänen Reijo	Siivooja Seppo
Su 6.5			
Ma 7.5	Asiakas1 Testi 8:00-15:00	Ryökäle Risto 11:00-16:00	
Ti 8.5	Vuosiloma	Asiakas Aapo 16:00-17:00	
Ke 9.5	Palkaton vapaa	Asiakas Aapo 8:30-11:00	
To 10.5	Asiakas Aapo 8:00-10:00	Ryökäle Risto 8:00-10:00	
	Ryökäle Risto 12:00-15:00	Vuosiloma	
Pe 11.5	Asiakas Aapo 7:30-10:00	Vuosiloma	
La 12.5			
Su 13.5			
Ma 14.5			<input type="button" value="Lisää"/>
Ti 15.5	Ryökäle Risto 8:00-12:00		
Ke 16.5	Asiakas1 Testi 8:00-15:00	Ryökäle Risto 8:00-10:00	
To 17.5			
Pe 18.5	Asiakas Aapo 7:30-10:00		
La 19.5			
Tunnit	28:00	12:30	0:00

KUVA 3. Työkaleri-sivu

Kaikkien työtehtävän tietojen näyttäminen yhdessä solussa olisi vienyt liikaa tilaa, joten yksittäisestä työtehtävästä näytetään vain kellonaika sekä asiakkaan nimi. Työtehtävää klikkaamalla nähdään kuitenkin työtehtävän muut tiedot, kuten kuvaus ja toistuvuus.

Työtehtävien lisäys tapahtuu klikkaamalla työlistan soluissa olevaa tyhjää tilaa. Hiiren kursori on tyhjän tilan kohdalla ristin muotoinen ja pidettäessä kursoria paikallaan noin sekunnin ajan, sen kohdalle tulee teksti Lisää. Klikkauksen jälkeen avautuu ruudun keskelle jQuery UI:n dialog-lomake (kuva 4). Lomakkeesta voidaan valita tapahtuman tyyppi, jossa vaihtoehtoina ovat työtehtävä, palkaton vapaa, sairausloma ja vuosiloma. Työntekijä-alasvetovalikossa on oletuksena sen työntekijän nimi, jonka sarakkeesta solua on klikattu. Työntekijä voidaan kuitenkin vielä muuttaa, jolloin onnistuneesti lisätty tehtävä siirtyy valitun työntekijän sarakkeeseen. Asiakas-alasvetovalikosta voidaan valita asiakkaan nimi, jolle työtehtävä suoritetaan. Päivämääräkentässä on oletuksena klikatun solun rivin päivämäärä, joka voidaan haluttaessa vaihtaa. Aikavälilenttiin merkitään työn alkuaika sekä loppuaika. Alkamisajan oletuksena on kello 8:00 ja loppuajan oletuksena on kello 10:00. Toistuva-alasvetovalikosta voidaan valita haluttaessa työn toistuvuusväli. Tehtävän voi laittaa toistuvaksi yhden, kahden tai neljän viikon välein. Valittaessa toistuvuus tulee toistuva-alasvetovalikon alapuolelle näkyviin päivämääräkenttä, johon laitetaan toistuvuuden loppupäivämäärä. Toistuvuuden loppupäivämäärä-kentässä on oletuksena neljän viikon kuluttua oleva päivämäärä. Kuvaus-tekstikenttään voidaan kirjoittaa työn kuvaus tai muuta tarpeellista tietoa tehtävästä.



The image shows a dialog box titled "Tehtävät" with a close button (x) in the top right corner. The dialog contains several input fields and buttons:

- Tapahtuman tyyppi: Työtehtävä (dropdown menu)
- Työntekijä: Siivooja Seppo (dropdown menu)
- Asiakas: Asiakas Aapo (dropdown menu)
- Päivämäärä: 4.5.2012 (text input)
- Aikaväli: 8:00 - 10:00 (text input)
- Toistuva: (dropdown menu)
- Kuvaus: (text area)

At the bottom of the dialog, there are three buttons: "Luo uusi tehtävä", "Muokkaa", and "Poista".

KUVA 4. Tehtävän lisäys

Dialog-lomake toimii html-formina ja lähettää Luo uusi tehtävä -painiketta klikkaamalla valitut tiedot Ajax-tekniikan avulla toiselle PHP-sivulle. Siellä tarkistetaan, onko työntekijällä muita tehtäviä samaan aikaan ja tallennetaan tehtävä tietokantaan, jos päällekkäisiä tapahtumia ei ole. Jos työntekijällä on samaan aikaan muita tehtäviä, tulostetaan virheilmoitus, jossa näytetään päällekkäisten tehtävien päivämäärät. Työtehtävän tietokantaan lisäyksen jälkeen sivu ladataan uudestaan, jotta lisätyt tehtävät näkyvät kalenterissa. Luo uusi tehtävä -painiketta klikattaessa tarkistetaan myös Javascriptillä tehdyn Regular Expressionin avulla, onko päivämäärät ja aika syötetty oikeassa muodossa. Päivämäärän oikea muoto on pp.kk.vvvv ja ajan tt:mm. Mikäli aika tai päivämäärä on väärässä muodossa, muuttuu kentän väri lomakkeessa punaiseksi. Tällöin Luo uusi tehtävä -painikkeesta ei tapahdu mitään, ennen kuin muoto on korjattu. Työtehtäviä muokataan ja poistetaan klikkaamalla työlistassa jo olevaa tehtävää, josta avautuu sama dialog-lomake kuin työtehtävän lisäyksessä, mutta siinä on jo valmiiksi tehtävän tiedot. Silloin Luo uusi tehtävä -painiketta ei voida klikata, mutta Muokkaa ja Poista -painikkeet ovat käytössä. Toistuvista tehtävistä voidaan muokata toistuvuuden alkupäivämäärää, aikoja, loppupäivämäärää sekä toistuvuusväliä. Toistuvia tehtäviä voidaan myös muokata yksitellen, jos esimerkiksi työ kestääkin jollakin kerralla odotettua pidempään.

Työtehtävien lisäystä, muokkausta ja poistoa varten on toteutettu oma luokkansa, jossa tarkistetaan työtehtävää lisättäessä tai muokattaessa päivämäärien päällekkäisyys. Toistuvista tapahtumista tarkistetaan päivämäärien päällekkäisyydet jokaiselta tehtävältä toistuvan tapahtuman loppuun saakka.

Päivämäärän käsittelyssä sovelluksessa käytettiin PHP:n strtotime- ja date-funktioita. Strtotime-funktiolla pystytään muuttamaan tekstimuodossa oleva päivämäärä Unix-ajaksi, joka kertoo, kuinka monta sekuntia on kulunut koordinoitun yleisajan (UTC) vuoden 1970 tammikuun 1. päivän keskiyöstä (php.net). Funktiolla voidaan myös muuttaa muuttujassa olevan Unix-ajan päivämäärää. Date-funktiolla taas pystytään tulostamaan päivämäärän osia, kuten esimerkiksi pelkästään vuosi.

6.4 Työntekijöiden hallinta ja Asiakkaat-sivu

Esimies voi lisätä, muokata tai poistaa työntekijöiden tietoja sekä vaihtaa kunkin työntekijän salasanan Työntekijät-sivulla. Siirryttäessä tälle sivulle avautuu salasanoja lukuun ottamatta lista kaikista työntekijöistä ja heidän tiedoistaan. Työntekijöiden tie-

toja ovat muun muassa osoitetiedot, sähköposti sekä puhelinnumerot. Työntekijät voidaan myös määritellä aktiiviseksi tai ei-aktiiviseksi. Vain aktiiviset työntekijät näytetään työkalenterissa. Uusi työntekijä lisätään listan lopussa olevan Lisää työntekijä -painikkeen avulla. Painikkeesta avautuu jQuery UI:n dialog-lomake, johon voidaan syöttää uuden työntekijän tiedot. Muokkaaminen ja poistaminen tapahtuvat klikkaamalla listasta halutun työntekijän riviä, josta avautuu samanlainen dialog-lomake kuin lisäyksessä, mutta siinä on työntekijän tiedot jo valmiiksi. Salasana vaihdetaan kirjoittamalla uusi salasana sekä Salasana- että Salasana uudestaan -tekstikenttiin.

Ohjelmaan tehtiin aluksi myös asiakkaiden hallinta, joka oli samanlainen työntekijöiden hallinnan kanssa. Sovelluksessa käytetyn tietokannan asiakkaat-taulu kuitenkin kopioidaan Oracle-tietokannassa olevasta taulusta tietyin väliajoin, joten kaikki Työkalenteri-sovelluksessa tehdyt muutokset asiakas-tauluun poistuisivat aina kopiointin yhteydessä. Niinpä asiakkaiden hallinta ei ollut tarpeellinen eikä myöskään käyttökelpoinen toimeksiantajalle. Sovellukseen sisällytettiin kuitenkin asiakkaat-sivu, josta esimies voi katsoa asiakkaiden tietoja. Asiakkaat-sivulla valitaan haluttu asiakas alasvetovalikosta, minkä jälkeen sovellus tulostaa näytölle asiakkaan tiedot. Asiakkaista nähdään esimerkiksi osoitetiedot sekä puhelinnumerot.

6.5 Omat tiedot -sivu

Omat tiedot -sivulla näytetään kirjautuneen käyttäjän tiedot tekstikentissä. Tekstikenttien tietoja voidaan muokata ja ne tallentuvat, kun painetaan sivulla olevaa Tallenna-nappia. Sivulla voidaan myös vaihtaa käyttäjän oma salasana sekä muuttaa kirjautumistunnusta.

6.6 Palkanlaskenta

Palkanlaskenta-sivun tarkoituksena on helpottaa työntekijöiden palkanlaskentaa näyttämällä tietoja tehdyistä töistä. Sivulla nähdään listassa valitun työntekijän tehdyt työtehtävät, sairaustunnit ja vapaapäivät valitulta aikaväliltä (kuva 5). Listassa on myös laskettu kaikista työtehtävistä kertyvät työtunnit ja asiakaskäyntien määrä. Palkanlaskenta-sivulla on myös näkyvissä valitun aikavälin sisältämät arkipyhäpäivät, jotta ne voidaan ottaa tarpeen mukaan huomioon palkanlaskennassa. Tehtävälistan vieressä

on myös lista näytetyllä aikavälillä olleista arkipyhäpäivistä, koska toiminto oli yritykselle tarpeellinen.

Työntekijä valitaan alavetovalikosta, josta löytyvät kaikki aktiiviseksi merkatut työntekijät. Päivämäärän valinta tapahtuu alkupäivämäärä- ja loppupäivämääräkenttien avulla. Hae-nappia painamalla ohjelma lähettää Ajax-tekniikan avulla valitut tiedot PHP-sivulle, jolla tulostetaan tehtävien tiedot sekä lasketaan työtunnit ja tehtyjen tehtävien määrä.

Ilman päivämääriä haetaan tämän kuun käynnit

Työntekijä: ▼

Alkupäivämäärä:

Loppupäivämäärä:

Päivämäärä	Tyyppi	Tunnit	Toistuvuus
2.5.2012	Tehtävä	3:30 tuntia	-
2.5.2012	Tehtävä	2:20 tuntia	-
2.5.2012	Tehtävä	3 tuntia	-
3.5.2012	Tehtävä	4 tuntia	-
3.5.2012	Tehtävä	4 tuntia	-
4.5.2012	Palkaton vapaa	-	-
7.5.2012	Tehtävä	6:02 tuntia	-
8.5.2012	Vuosiloma	-	-
9.5.2012	Palkaton vapaa	-	-
10.5.2012	Tehtävä	1:15 tuntia	-
10.5.2012	Tehtävä	3 tuntia	-
15.5.2012	Tehtävä	4 tuntia	-
16.5.2012	Tehtävä	7 tuntia	-
21.5.2012	Tehtävä	3 tuntia	kuukausittain
22.5.2012	Sairausloma	3:36 tuntia	-
29.5.2012	Tehtävä	2 tuntia	-

Pyhäpäivät:

Vappu 1.5
Helatorstai 17.5

Käyntejä yhteensä: 12
Palkattomat vapaat: 2 päivää
Vuosilomapäivät: 1 päivä
Pyhäpäivät: 2 päivää
Sairaspoissaolot: 3:36 tuntia
Tunnit yhteensä: 43:07 tuntia

KUVA 5. Palkanlaskenta-sivu

6.7 Laskutus

Laskutus-sivun tarkoituksena on helpottaa asiakkaiden laskuttamista näyttämällä tietoja yksittäiselle asiakkaalle tehdyistä tehtävistä. Sivulla näytetään kaikki valitulla aikavälillä asiakkaalle tehdyt työtehtävät listassa. Kaikilla näytetyillä työtehtävillä on tietokenttinä päivämäärä, työtunnit sekä tehtävän toistuvuus. Liian myöhään siirretyistä tehtävistä näytetään lisäksi muutospäivämäärä, jotta voidaan ottaa laskutuksessa huomioon siirtomaksu. Liian myöhään peruutetuista tehtävistä näytetään tehtävän peruutuspäivämäärä. Jokaisesta tehtävästä näytetään lisäksi kuvaus. Sivulla näytetään listassa näkyvistä työtehtävistä kertyvät työtunnit ja montako kertaa asiakkaalle on tehty töitä. Peruutetuista tehtävistä ei lasketa työtunteja eikä sitä lueta tehtyihin töihin. Tehtävälistan vieressä on myös samanlainen lista näytetyllä aikavälillä olleista arkipyhäpäivistä, kuin palkanlaskentasivulla.

Sivun toiminta on samanlainen kuin palkanlaskenta-sivulla, eli valitaan alaspäivälistä asiakas, sekä päivämääräkentistä alkupäivämäärä ja loppupäivämäärä (kuva 6). Hae-nappia painamalla ohjelma lähettää Ajax-tekniikan avulla valitut tiedot PHP-sivulle, jolla tulostetaan tehtävien tiedot sekä lasketaan työtunnit ja tehtyjen tehtävien määrä.

Ilman päivämääriä haetaan tämän kuun käynnit

Asiakas:

Alkupäivämäärä:

Loppupäivämäärä:

Päivämäärä	Tunnit	Toistuvuus	Muutettu	Peruutettu	Kuvaus
2.5.2012	2 tuntia	viikottain			
2.5.2012	2:20 tuntia	-			
3.5.2012	4 tuntia	-			
7.5.2012	5 tuntia	-			
9.5.2012	2 tuntia	viikottain			
10.5.2012	3 tuntia	-			
11.5.2012	2 tuntia	-			
15.5.2012	4 tuntia	-			
16.5.2012	2 tuntia	viikottain			
23.5.2012	2 tuntia	viikottain			
30.5.2012	2 tuntia	viikottain			

Käynnejä yhteensä: 11

Tunnit yhteensä: 30:20 tuntia

Pyhäpäivät:

Vappu 1.5
Helatorstai 17.5

KUVA 6. Laskutus-sivu

7 TESTAUS

Testauksen tavoitteena oli löytää sovelluksesta mahdolliset virheet sekä tarkistaa, että sovelluksen vaatimukset täyttyivät. Ohjelman testauksessa käytettiin sovelletusti v-mallia, joka on perinteinen testauksessa käytetty malli. Testaus sisälsi kolme eri vaihetta, jotka olivat yksikkötestaus, integraatiotestaus sekä järjestelmätestaus. Nämä vaiheet ovat samoja kuin v-mallissa olevat testausvaiheet. Sovelluksen testausvaiheet eivät kuitenkaan täysin vastanneet v-mallissa olevia käytäntöjä. V-mallissa on käyttämieni vaiheiden lisäksi vielä hyväksymistestaus, mutta sitä ei tässä opinnäytetyössä ehditty tekemään valmiiksi. (Mathur & Malik 2010, 30.) Yksikkötestausta suoritettiin osana ohjelmointivaihetta ja siinä testattiin funktioiden sekä metodien toimivuutta. Integraatiotestausta tehtiin myös ohjelmointivaiheessa, mutta aina yksikkötestauksen jälkeen. Siinä testattiin suurempia kokonaisuuksia sekä niiden yhteensopivuutta. Viimeisenä suoritettiin järjestelmätestaus, missä testattiin Työkaleri-sivun suorituskykyä, koska se oli sovelluksen raskain osa. Suorituskyvyn testaus suoritettiin etäpalvelimella, koska paikallisella palvelimella ei viivettä oikeastaan ole. Testausvaiheissa esiintyneet virheet korjattiin aina niiden löydyttyä, koska testauksessa ei käytetty testaussuunnitelmaa eikä virheitä kirjattu ylös.

Testauksessa sekä virheiden korjaamisessa käytettiin apuna selaimien lisäosia sekä ohjelmointiympäristön ja ohjelmointikielen tarjoamia virheraportointeja. Netbeans helpotti sovelluksen ohjelmointia ja testausta ilmoittamalla esimerkiksi puolipisteiden puuttumisesta johtuvat virheet. Sovellusta testattiin kokonaisuutena ensin paikallisella palvelimella ennen sen siirtoa Uudenmaan tietotiimin palvelimelle. Paikallisen testauksen jälkeen sovellus siirrettiin etäpalvelimelle, ja tarkastettiin, että sovelluksen toimintaan ei tullut muutoksia. Sivustoa testattiin eri selaimilla, jotta sovelluksen ulkoasu ja toiminnot olisivat yhdenmukaisia. Testattaviksi selaimiksi valittiin kolme yleisimmin käytettyä selainta, jotka ovat Mozilla Firefox, Google Chrome sekä Microsoft Internet Explorer. Javascriptillä ja jQueryllä tehtyjen toimintojen testauksessa käytettiin apuna Mozilla Firefox-selaimen liitännäistä nimeltään Firebug sekä Google Chromen Javascript-debuggeria. PHP:n testauksessa käytettiin apuna PHP:n omia virheraportointeja sekä tulostus-funktioita.

Sovellusta testattiin pääasiassa syöttämällä eri arvoja tekstikenttiin sekä kokeilemalla kaikkia käyttäjälle mahdollisia valintoja esimerkiksi alasvetovalikoista. Luokkien toimivuutta testattiin esimerkiksi tulostamalla sille kuuluvissa metodeissa olevia muuttujia.

Testausta varten luotiin testiasiakkaita sekä testityöntekijöitä. Toimintoja testattiin esimerkiksi syöttämällä arvoja uuden tehtävän luonti- ja muokkauslomakkeisiin. Toistuvien tehtävien lisäämistä testattiin esimerkiksi luomalla työtehtäviä työkalenteriin ja sen jälkeen lisättiin toistuva tehtävä, joka tehtiin päällekkäiseksi luotujen tehtävien kanssa.

8 TYÖN ARVIOINTI

Työn tavoitteena oli suunnitella ja tehdä Kodin Apujoukoille toimiva web-kalenterisovellus, joka on helppokäyttöinen ja selkeä. Helppokäyttöisyyttä sovellukseen saatiin esimerkiksi toteuttamalla tehtävien hallinta työkalenterin päälle tulevan dialog-lomakkeen avulla. Tällä lomakkeella saavutettiin se, että kalenterisivua ei tarvinnut vaihtaa, kun haluttiin muokata tehtäviä. Näin sivusta saatiin myös interaktiivisempi ja nykyaikaisempi sekä käyttäjän kannalta paljon selkeämpi. Mielestäni sovelluksesta saatiin kokonaisuudessa erittäin helppokäyttöinen, koska suunnitteluvaiheessa tähän kiinnitettiin erityisen paljon huomiota.

Työkalenterista haluttiin helposti muokattava, jotta sillä olisi jälleenmyyntiarvoa. Tämä tavoite ei onnistunut täysin, sillä toimeksiantajan haluama räätälöinti teki siitä hankalan toteuttaa. Ohjelma on muokattavissa muidenkin yritysten käyttöön, mutta se vaatisi jonkin verran vaivannäköä. Sovelluksesta voidaan kuitenkin muuttaa suhteellisen helposti joitakin asioita, esimerkiksi kalenterissa näkyviä osia. Ulkonäköä voidaan myös muuttaa yksinkertaisesti, koska kaikki tyylit löytyvät css-tiedostoista.

Kokonaisuudessaan opinnäytetyöprojekti onnistui hyvin. Toimeksiantaja oli tyytyväinen ohjelmaan, ja se soveltui hyvin yrityksen tarpeisiin. Tätä edesauttoi toimintojen testaus molempien osapuolten toimesta. Kaikki toimeksiantajan antamat vaatimukset ohjelman toiminnoista saatiin täytettyä. Ohjelmasta saatiin toimiva ja siitä ei löytynyt virheitä, mikä oli myös tavoitteena. Työkalenterin toteutuminen web-pohjaisena ja reaaliaikaisena helpottaa työntekijöiden työtehtävien seuraamista, mikä tehostaa toimintaa.

Opinnäytetyön tarkoituksena olleiden ajankäytön tehostamisen ja työtehtävien hallinnan tulokset eivät ole valmistuneet, koska ohjelmaa ei ole otettu vielä käyttöön. Yrityksen tulisikin sovelluksen käyttöönotossa huomata, ettei se käyttäisi enää vanhoja järjestelmiä uuden rinnalla lopullisen testauksen jälkeen. Uuden työkalun tehokas käyttö onnistuu nimittäin vain, jos vanhat työkalut hylätään yrityksessä (Tiirikainen 2010, 17).

9 POHDINTA

Jos aloittaisin projektin alusta, käyttäisin enemmän aikaa asiakkaiden vaatimusten selvittämiseen. Näin vaatimukset olisivat molemmille osapuolille jo alusta asti selvät. Suunnittelisin myös tarkemmin ohjelmoinnissa käytetyt luokat ja funktiot, jotta samankaltaisen koodin toistoa saataisiin vähennettyä. Tekisin lisäksi kattavamman funktioiden kommentoinnin, jotta niiden toiminta saataisiin selville nopeammin.

Vaikeimpana yksittäisenä osana ohjelmoinnissa oli mielestäni toistuvia työtehtäviä muokattaessa tehtävät tarkistukset, jotka aluksi yritin toteuttaa yhden funktion sisällä. Kun aloin lisätä tarkastuksia toistuvia tehtäviä varten huomasin if-lauseita ja samoja tarkastuksia olevan niin paljon, että oli vaikeaa hahmottaa, missä ollaan menossa. Ymmärsin siinä vaiheessa, että eri tarkastukset on viisainta tehdä omissa funktioissaan, joten loin luokan, johon tein tarkastukset. Tämän seurauksena koodin ymmärtäminen ja järjestys olivat huomattavasti helpompia ymmärtää.

Opinnäytetyöprojekti opetti minulle paljon ohjelmistokehityksestä sekä web-ohjelmoinnista. Ohjelmointivaiheessa opin tekemään koodia loogisessa järjestyksessä ja käyttämään luokkia PHP:n avulla. Opin esimerkiksi, miten päivämäärien tarkastukset kannattaa toteuttaa järjestelmällisesti. Suunnittelusta opin muun muassa, että kannattaa kirjata asiat johdonmukaisesti sekä nimetä tiedostot ja muuttujat siten, että ne kertovat jotain itsestään. Opin myös, miten tärkeää on huomioida ohjelmien ja kielten eri versiot. Siirtäessäni sovelluksen ensimmäistä kertaa toimeksiantajan palvelimelle, huomasin, etten ollut kysynyt palvelimella olevan PHP:n versiota. Onnekseni jouduin vain korvaamaan työtuntien laskemiseen käyttämäni luokan, joka oli tullut PHP:n uudemmassa versiossa, eikä asiasta seurannut muita ongelmia.

Sovimme toimeksiantajan kanssa, että teen vielä pieniä korjauksia ohjelmaan. Jatkokehityksenä työkalenteriin voitaisiin tehdä esimerkiksi toiminto, jolla asiakkaille olisi mahdollista lähettää lasku tehdyistä töistä tai tulostaa se. Myös palkkalaskelman tekeminen ohjelman avulla voisi olla jatkokehityksen aiheena.

LÄHTEET

Asleson, R. & Schutta, N. 2007. *Ajax*. Helsinki: Readme.fi.

BuiltWith [verkkosivu]. [viitattu 27.4.2012]. Saatavissa:

<http://trends.builtwith.com/framework/PHP>

Järvenpää, P. & Hänninen, J. 2011. *Paranna liiketoiminnan tuottavuutta tietotekniikalla*. Helsinki: Teknologiateollisuus ry.

Kolehmainen, K. 2006. *PHP & MySQL teoriasta käytäntöön*. Jyväskylä: Readme.fi.

Mathur, S. & Malik, S. 2010. Advancements in the V-Model. *International Journal of Computer Applications*. 2010, nro 12, 29–34. [viitattu 21.5.2012]. Saatavissa:

<http://www.ijcaonline.org/journal/number12/pxc387425.pdf>

Mozilla Developer Network 1 [verkkosivu]. [viitattu 5.5.2012]. Saatavissa:

https://developer.mozilla.org/en/About_JavaScript

Mozilla Developer Network 2 [verkkosivu]. [viitattu 5.5. 2012]. Saatavissa:

https://developer.mozilla.org/en/JavaScript/Guide/JavaScript_Overview#JavaScript_and_the_ECMAScript_Specification

Netbeans [verkkosivu]. [viitattu 25.4.2012]. Saatavissa: <http://netbeans.org>

Peltomäki, J. & Nykänen, O. 2006. *Web-selainohjelmointi*. Jyväskylä: Docendo.

Tiirikainen, V. 2010. *IT ja parempi bisnes*. Helsinki: Talentum Media Oy.

WampServer [verkkosivu]. [viitattu 25.4.2012]. Saatavissa:

<http://www.wampserver.com/en/>

Projektisuunnitelma

Joonas Roivainen

29.5.2012
versio 1.0

VERSIOHISTORIA

Versio	Muutokset	Pvm
0.1	Aloitettu	28.2.2012
0.2	Lisätty aikatauluun valmistumispäiviä sekä korjattu muotoilua.	25.3.2012
0.3	Korjattu oikeinkirjoitus sekä muotoilu. Lisätty aikatauluun loput valmistumispäivät.	23.5.2012
0.4	Korjauksia	27.5.2012
1.0	Julkaisuversio	29.5.2012

SISÄLTÖ

1	JOHDANTO.....	4
1.1	Yleiskuvaus.....	4
1.2	Tuote	4
1.3	Unelmapaketti Ay	4
2	PROJEKTIN ORGANISOINTI.....	5
2.1	Projektin vaiheistus.....	5
2.2	Projektin organisaatio	5
3	PROJEKTIN OHJAAMINEN.....	6
3.1	Tavoitteet ja priorisointi	6
3.2	Riskien hallinta	6
3.3	Seuranta ja ohjaus.....	6
4	TEKNIikka.....	7
4.1	Menetelmät ja työkalut.....	7
4.2	Dokumentointi.....	7
5	VAIHEET JA AIKATAULUT	8

1 JOHDANTO

1.1 Yleiskuvaus

Tämä dokumentti on projektisuunnitelma opinnäytetyönä tehtävälle työvuorokalenterille sekä siitä tehtävälle raportille. Projektisuunnitelmassa kerrotaan opinnäytetyön tavoitteista sekä toteutettavasta sovelluksesta. Suunnitelmassa esitellään myös käytettävät tekniikat sekä mukana olevat henkilöt. Lisäksi dokumentissa arvioidaan opinnäytetyön aikataulu sekä riskit.

1.2 Tuote

Projektin aiheena on suunnitella ja tehdä työvuorokalenteri Kodin Apujoukoille sekä tehdä siitä opinnäytetyöraportti. Työvuorokalenteri tulee hyödyntämään Kodin Apujoukkojen asiakastietokantaa. Työvuorokalenteri tulee sisältämään viikkokalenterin, josta työntekijät pystyvät katsomaan työtehtävänsä. Esimies tulee näkemään omassa työkalenterissaan kaikkien työntekijöiden työtehtävät valitulta aikaväliltä. Työkalenteriin voidaan myös lisätä tehtäviä klikkaamalla haluttua päivämääräkenttää. Ohjelma tulee sisältämään myös työntekijöiden hallintasivun sekä asiakassivun, jossa näkyvät asiakkaiden yhteystiedot.

1.3 Unelmapaketti Ay

Unelmapaketti Ay on yritys, joka sisältää tietotekniikkapalveluja tarjoavan Uudenmaan tietotiimin sekä kodinhoitopalveluja tarjoavan Kodin Apujoukot. Opinnäytetyönä tehtävä työvuorokalenteri tehdään yhteistyössä Unelmapaketti Ay:n yrittäjien Jari Laaksosen sekä Eeva Väisäsen kanssa. Työvuorokalenterin on tarkoitus tulla Kodin Apujoukkojen käyttöön.

2 PROJEKTIN ORGANISOINTI

2.1 Projektin vaiheistus

Alla olevassa taulukossa 1 on kuvattu projektin eri vaiheet.

TAULUKKO 1. Projektin vaiheet

1.	Työvuorokalenterin suunnittelu ja määrittely
2.	Projektisuunnitelman tekeminen
3.	Työvuorokalenterin toteutus
4.	Työvuorokalenterin testaus ja viimeistely
5.	Opinnäytetyöraportin suunnittelu
6.	Opinnäytetyöraportin kirjoittaminen
7.	Opinnäytetyöesitelmän suunnittelu
8.	Opinnäytetyön esittely seminaarissa
9.	Opinnäytetyöraportin viimeistely
10.	Opinnäytetyön julkaisu

2.2 Projektin organisaatio

Alla olevassa taulukossa 2 kuvataan projektissa mukana olevat henkilöt. Taulukon tehtävä-kentässä kerrotaan mikä on kyseisen henkilön rooli projektissa.

TAULUKKO 2. Projektin organisaatio

Nimi	Titteli	Sähköposti	Rooli
Sami Lahti	lehtori	sami.lahti@savonia.fi	Opinnäytetyön ohjaaja
Jussi Koistinen	lehtori	jussi.koistinen@savonia.fi	Opinnäytetyön ohjaaja
Eeva Väisänen	yrittäjä		Työn toimeksiantaja
Jari Laaksonen	yrittäjä		Työn toimeksiantaja

3 PROJEKTIN OHJAAMINEN

Tässä luvussa kerrotaan opinnäytetyön tavoitteet sekä priorisointi. Luvussa selvitetään myös projektin riskit ja toiminta niiden sattuessa.

3.1 Tavoitteet ja priorisointi

Tavoitteena on saada toimiva ja helppokäyttöinen työvuorokalenteri Kodin apujoukot yrityksen käyttöön. Toimintojen saaminen valmiiksi priorisoidaan ennen visuaalista ilmettä. Opinnäytetyöraportin tarkoituksena on antaa kattava kuvaus tehdystä työstä, käytetyistä tekniikoista sekä saavutetuista tuloksista. Tavoitteena on saada opinnäytetyöraportin tekstistä mahdollisimman selkeää ja johdonmukaista.

3.2 Riskien hallinta

Alla olevassa taulukossa 3 kuvataan projektin riskit sekä mitä tehdään riskin sattuessa.

TAULUKKO 3. Riskit

Riski	Toiminta riskin sattuessa
Aikataulun pettäminen	Ylityöt
Työkoneen hajoaminen	Koneen vaihto
Tietojen häviäminen	Varmuuskopioiden palauttaminen

3.3 Seuranta ja ohjaus

Toimintojen valmiusastetta seurataan Microsoft Excelin avulla tehtävällä toimintosuunnitelmalla, jota päivitetään noin viikon välein. Viikoittain tehdään lisäksi muistio tehdystä työstä. Jari Laaksosen kanssa sovitaan teknisestä toteutuksesta ja Eeva Väisäsen kanssa sovitaan tulevista toiminnoista sekä käyttöliittymän muotoilusta. Opinnäytetyön ohjaajilta kysytään tarvittaessa neuvoa. Opinnäytetyön etenemisen seuranta tapahtuu myös opinnäytetyöohjaajien puolesta.

4 TEKNIikka

Tässä luvussa kerrotaan projektissa käytettävistä ohjelmointikielistä, työkaluista sekä muista tekniikoista. Luvussa esitellään myös tarvittavat dokumentit.

4.1 Menetelmät ja työkalut

Opinnäytetyönä tehtävän sovelluksen tietokantana on MySQL, ohjelmointikielina PHP sekä Javascript. Javascriptin ohjelmoinnissa käytetään apuna jQuery-kirjastoa sekä jQuery:n avulla tehtyä jQuery UI -lisäosaa. Kuvauskielina käytetään HTML:ää sekä CSS:ää. Ohjelmointivälineenä käytetään Netbeans IDE ohjelmointiympäristöä ja valmiiden sivujen näyttämiseen kotikoneella käytetään WampServeriä. Palvelimelle tiedot siirretään WinSCP-ohjelman avulla.

4.2 Dokumentointi

Projektin dokumentaatioon kuuluvat projektisuunnitelman lisäksi toimintolista, opinnäytetyöraportti ja opinnäytetyöesitelmä. Toimintolistassa kerrotaan kaikista ohjelman toiminnoista nimi, kuvaus sekä valmiusaste. Opinnäytetyöraportissa kerrotaan projektistä kokonaisuudessaan. Opinnäytetyöesitelmässä on tiivistettynä opinnäytetyöraportin sisältö.

5 VAIHEET JA AIKATAULUT

Alla olevassa taulukossa 4 kerrotaan projektin aikataulusta. Siinä arvioidaan projektin eri osien valmistumispäivämääriä sekä kirjataan toteutunut valmistumispäivämäärä.

TAULUKKO 4. Aikataulu

Tehtävä	Arvioitu valmistuminen	Toteutunut valmistuminen
Työvuorokalenterin suunnittelu	25.2.2012	25.2.2012
Viikkokalenteri	5.3.2012	15.3.2012
Työntekijöiden hallinta	9.3.2012	8.3.2012
Asiakkaiden hallinta	10.3.2012	11.3.2012
Työvuorojen hallinta	20.3.2012	20.3.2012
Sisäänkirjautuminen	20.3.2012	20.3.2012
Testaus	3.4.2012	16.4.2012
Työvuorokalenterin viimeistely	8.4.2012	21.4.2012
Opinnäytetyöraportin sisällön suunnittelu	1.4.2012	15.4.2012
Opinnäytetyö esitelmän suunnittelu	22.4.2012	22.4.2012
Opinnäytetyö esitelmä	26.4.2012	26.4.2012
Opinnäytetyöraportti	20.5.2012	23.5.2012