

Tomi Nousiainen

# Töiden raportointi- ja laskutusohjelma BASE

Web-sovellus

Metropolia Ammattikorkeakoulu  
Insinööri (AMK)  
Tietotekniikan koulutusohjelma  
Insinöörityö  
8.5.2011

## ALKULAUSE

Tämä insinööri työ tehtiin yritykselle STD Systems Oy. Kiitän toimitusjohtaja Mikko Töykkälää työn aiheesta sekä tuesta työn toteutuksen aikana. Haluaisin kiittää myös Yritystilin toimitusjohtaja Pekka Mutasta tuloksellisesta yhteistyöstä projektin tiimoilta sekä opastuksesta työn aikana. Kiitos myös työn ohjaajalleni Jarkko Vuorelle.

Erityiskiitos vanhemmilleni Esalle ja Hannelelle kärsivällisyydestä ja tuesta opintojeni aikana.

Tekijä Otsikko	Tomi Nousiainen Töiden raportointi- ja laskutusohjelma BASE
Sivumäärä Aika	35 sivua 8.5.2011
Tutkinto	insinööri (AMK)
Koulutusohjelma	tietotekniikka
Suuntautumisvaihtoehto	ohjelmistotekniikka
Ohjaajat	yliopettaja Jarkko Vuori toimitusjohtaja Mikko Töykkälä
<p>Insinööriyössä esitellään BASE-toiminnanohjausjärjestelmä sekä sen määrittelyä, suunnittelua, toteutusta ja käyttöönottoa. Työn tavoitteena oli kehittää helppokäyttöinen järjestelmä työajan raportointiin, joka toimisi laskutusperusteena tehdyistä töistä.</p> <p>Työssä esitellyn määrittelyn perusteella rakennettiin web-pohjainen portaali, joka toimii työntekijöiden raportoinnin portaalina, laskuttajan työkaluna sekä tarjoaa asiakkaalle sähköisen laskutuksen seurantatyökalut. Järjestelmä tehtiin Yritystili Finland Oy:lle. Markkinoilla olevista järjestelmistä ei löytynyt täysin tarpeeseen sopivaa ja halutunlaista ohjelmistoa, joten päädyttiin vielä tutkimaan avoimen lähdekoodin järjestelmiä. Kun oli käytetty Sillajin avoimen lähdekoodin sovellusta vuoden verran, päädyttiin järjestelmän tilaamiseen, koska Sillajin toiminnot eivät vain enää riittäneet.</p> <p>Järjestelmä on rakennettu Linux-pohjaiseen palvelinkoneeseen, jossa HTTP-palvelinohjelmana toimii Apache. Palvelinpuolen komentosarjakielenä käytetään PHP:tä ja tiedot tallennetaan MySQL-tietokantaan. Käyttäjätunnistukseen käytetään LDAP-tekniikkaa.</p> <p>Järjestelmä valmistui aikataulustaan myöhässä, mikä johtui kehitysvaiheessa lisätyistä ominaisuuksista. Onnistuneen pilottivaiheen jälkeen järjestelmä otettiin käyttöön Yritystilin 25:n asiakkaan kanssa. Laskutukseen käytettävä aika on vähentynyt kolmasosaan eikä yhtään järjestelmään kirjattua työtä enää ole jäänyt laskuttamatta. Asiakkaat ovat myös olleet tyytyväisiä pystyessään tarkastamaan laskujensa perusteet järjestelmästä tietäen, etteivät maksa mistään ylimääräisestä.</p> <p>Tulevaisuudessa järjestelmää kehitetään asiakaspalautteen myötä sekä sen liityntöjä laajennetaan mm. reskontraohjelmiin, jolloin laskutus voidaan synkronoida suoraan järjestelmästä käsin. Myös asiakkaille tehdään mahdollisuus kirjoittaa laskuja järjestelmän kautta, jolloin ne kirjautuvat suoraan oman tilitoimistonsa reskontraohjelman kautta kirjanpitoon.</p>	
Avainsanat	raportointi, laskutus, web-sovellus, PHP

Author Title	Tomi Nousiainen BASE ERP-system
Number of Pages Date	35 pages 8 May 2011
Degree	Information Technology
Degree Programme	Bachelor of Engineering
Specialisation option	Software Engineering
Instructor(s)	Jarkko Vuori, Principal Lecturer Mikko Töykkälä, CEO
<p>This thesis presents a BASE ERP system and its specification, design, implementation and deployment. The aim was to develop an easy-to-use system of working hours for reporting and billing activities based on work carried out.</p> <p>Based on the definition, WEB-based portal was built. The portal serves as reporting portal for workers, as a tool for the biller and provides electronic billing tracking tools for customers. The portal was created for Yritystili Finland Oy. There was no commercially available systems what would have met the requirements, so we opted for investigating trough open-source systems. After using Sillaj for a year, Yritystili decided to order a new fully customized system as Sillaj's functions were no longer sufficient.</p> <p>The system was built into a Linux-based server running Apache as HTTP server software is Apache. PHP is used as a server-side scripting language and all data is saved on a MySQL-database. User authentication is done by LDAP technology.</p> <p>The system was completed behind schedule, due to the development of additional features. After a successful pilot phase, the system was introduced to Yritystili's 25 customers. The system has dramatically decreased the time needed for processing the bills, while no failure in processing has occurred. Customers have also been pleased to be able to check their invoices, without incurring extra cost.</p> <p>In the future the system will be developed by customer feedback and its interfaces will be extended to ledger programs. With this extension it is possible to synchronize billing with the system. There is also a plan for allowing customers to write out bills for their own customers through the system.</p>	
Keywords	web, reporting, PHP

## Sisällys

1	Johdanto	1
2	BASE toiminnanohjausjärjestelmä	3
2.1	Järjestelmän tarve	3
2.2	Projektin aloitus	3
3	Määrittely	5
3.1	Lähtökohdat	5
3.2	Tietoturva	5
3.3	Yhteensopivuus	6
3.4	Työntekijän toiminnot	7
3.5	Töiden hakeminen järjestelmästä	7
3.6	Pääkäyttäjän ja laskuttajan toiminnot	8
3.7	Asiakkaan toiminnot	13
3.8	Raporttien tulostustoiminnot	14
4	Suunnittelu	15
4.1	Verso-kehysrakenne	15
4.2	Tietokanta	16
5	Toteutus	18
5.1	web-sovellustekniikat	18
5.2	PHP	19
5.3	MySQL	19
5.4	LDAP	20
5.5	Mootools	20
5.6	HTTP-palvelin	21
5.7	Palvelinkone	22
5.8	SortableTable	23
5.9	BaseCore-luokka	24
5.10	Lomakkeet	27
5.11	Tulostustoiminnot	29
5.12	Pilottivaihe	31
6	Yhteenveto	33
	Lähteet	34

## Lyhenteet

Ajax	<i>Asynchronous Javascript and XML.</i> Tekniikka, jolla tietoa voidaan siirtää selaimen ja palvelimen välillä ilman uutta sivunlatausta.
CSS	<i>Cascading Style Sheet.</i> WWW-dokumenteille kehitetty tyyliohjeiden laji.
DOM	<i>Document Object Model.</i> HTML-sivun rakenteen esitystapa.
GPL	<i>General Public Lisence.</i> Maailman yleisin käytetty lisenssityyppi ilmaiselle ohjelmistolle
HTML	<i>Hyper Text Marking Language.</i> Verkkosivujen tekoon käytetty kuvauskieli.
HTTP	<i>Hyper Text Transfer Protocol.</i> Web-palvelimissa käytetty siirtoprotokolla.
JSON	<i>Javascript Object Notation.</i> Yksinkertainen tiedonsiirtomuoto.
LDAP	<i>Lightweight Directory Access Protocol.</i> Käyttäjätunnistukseen tarkoitettu kevyt verkkoprotokolla.
PHP	<i>PHP: Hypertext Preprocessor.</i> Yksi palvelimissa käytetyistä komentosarjakielistä.
SFTP	<i>Secure File Transfer Protocol.</i> Salattu tiedonsiirtoprotokolla.
SSL	<i>Secure Sockets Layer.</i> Eräs salausprotokolla.
TLS	<i>Transport Layer Security.</i> Eräs salausprotokolla.
XML	<i>Extensible Markup Language.</i> Merkintäkieli.

## 1 Johdanto

Tässä insinööriyössä esitellään vaiheittain määrittelystä toteutuksen kautta loppukäyttäjän palautteeseen web-käyttöliittymäpohjainen töiden raportointi- ja laskutussovellus BASE. Sovellukseen syötetyistä työraporteista voidaan koostaa sähköisiä laskunliitteitä asiakkaille laskutus-id:n avulla. Sovelluksen avulla saadaan talteen asiakkaille tehty työhistoria sekä digitaalinen arkisto laskutuksen perusteille.

Sovellus on kehitetty sellaisia toimialoja ajatellen, joilla asiakas ei maksa välttämättä fyysisestä tuotteesta vaan palvelusta. Palvelu voi olla tilitoimistotyypistä toimintaa, jossa palvelun hinta riippuu työhön käytetyistä tunneista tai muusta vaihtelevasta määreestä, jota ei voida ennalta arvioida. Sovelluksen avulla asiakkaalle tuodaan mahdollisuus selvittää laskun summan peruste kirjautumalla palveluun ja aukaisemalla laskun liitteessä ollut laskutus-id. Asiakas voi myös kesken laskutuskauden seurata meneillään olevia töitä sekä mahdollisesti jo valmistuneita töitä.

Työnantajan näkökulmasta saavutetaan reaaliaikainen seuranta töiden kulusta sekä niihin kuluvasta ajasta. Laskutukseen käytettävä aika myös vähentyy, mikä antaa mahdollisuuden käyttää aikaa muuhun tuottavampaan työhön.

STD Systems Oy

STD Systems Oy on vuonna 2005 perustettu tietotekniikan alan yritys, jossa olen ollut vuoden 2006 alusta alkaen käytännössä ensimmäisenä työntekijänä. Yritys työllistää tällä hetkellä kymmenen henkilöä. STD Systems kehittää tietoteknisiä järjestelmiä logistiikan hallinnan helpottamiseen. Logististen järjestelmien täysimittaisen käytön hyödyntämisen takia STD Systems kehittää erilaisia web-pohjaisia palveluita, joten teknikat, kuten PHP, Javascript ja MySQL, ovat varsin tuttuja. STD:n tuotteita ovat muun muassa APE-bussimittari, Versotrack-seurantajärjestelmä sekä APE-Schoolbook.

## Yritystili Finland Oy

Yritystili Finland on Espoon Suomenojalla toimiva vuonna 2006 perustettu täyden palvelun tilitoimisto. Yritystili käyttää pääasiassa Nova- sekä Tieto tisman -kirjanpitojärjestelmiä. Toiminnassa on erityisen tärkeää asiakastyytyväisyys, jota tiedustellaan asiakkaalta tasaisin aikavälein. Yritystilin omassa käytössä on tässä työssä esitelty töiden raportointi- ja laskutusohjelma BASE, johon toimitusjohtaja Pekka Mutanen on ollut erittäin tyytyväinen.

## **2 BASE toiminnanohjausjärjestelmä**

### **2.1 Järjestelmän tarve**

Järjestelmän alkuperäinen tarve ulottuu aina Yritystilin perustamiseen saakka vuoteen 2006, jolloin tiedettiin yrityksen liiketoiminnan perustuvan tarkkaan laskutettavaan työhön asiakkaalta. Väliaikaiseksi ratkaisuksi otettiin käyttöön perinteinen Excel-taulukko, johon merkittiin asiakkaittain tehdyt tunnit, sekä joitain viitteitä siitä mitä oli tehty. Laskuja tehdessä on kaikkien työntekijöiden Excelit avattu ja niistä on käsin koostettu tunnit asiakkaille, mikä vie aikaa keneltä tahansa aivan riittävän kauan.

Yritystilin toimitusjohtajan Pekka Mutasen toimiessa välillä konsulttina STD Systems Oy:ssä tuli vuonna 2006 puheeksi kyseinen ongelma ja hän tiedusteli, olisiko STD:llä mitään ohjelmistoa kyseiseen tarkoitukseen. STD:llä ei ollut tarvittavaa ohjelmistoa, mutta yhdessä selattiin netistä sopivia vapaan lähdekoodin järjestelmiä ja testailtiin muutamaa, joista lähimmäksi osui Sillaj-niminen ajanhallintaan tehty ohjelma [18]. Sillaj asennettiin palvelimelle ja sen käyttöliittymälle tehtiin suomennos. Sillaj otettiin Yritystilille käyttöön vuonna 2007.

Ajan kuluessa Yritystili halusi muutoksia ja käytännöllisempiä ominaisuuksia Sillajiin, mutta se ei olisi onnistunut helposti. Muutoksien teko olisi vaatinut pitkää opiskelemista, miten tietokannat ja muu rakenne oli toteutettu. Muutostyöt päätettiin hylätä ja lopulta alkaa toteuttaa omaa järjestelmää, joka olisi rakennettu täysin omilla ehdoilla ja asiakkaitakin ajatellen.

### **2.2 Projektin aloitus**

Projekti aloitettiin pitämällä aloituspalaveri STD:n tiloissa Espoossa. Aloituspalaverissa sovittiin projektin tavoitteista ja aikataulusta. Aloituspalaveri oli erilainen muihin projekteihin verrattuna, joissa yleisesti ottaen kaikilla projekteilla on kiire valmistua niin nopeasti kuin mahdollista ja lopullinen tuote vastaa asiakkaan senhetkiseen tarpeeseen tai ongelmaan ja tähän halutaan tilattu tuote tai palvelu ratkaisuksi. Palaverissa kävi ilmi, että lopullinen tuote ja kaikki sen toiminnot eivät olleet täysin yksiselitteisesti esitettävissä, vaan ainoastaan tarve pystyttiin selvittämään ongelman kautta.

Projektia lähdettiin vetämään vaiheittaisesti osa-alue kerrallaan ja ominaisuuksia lisättäisiin tarpeen mukaan. Projekti aloitettiin helmikuussa 2010 ja tuotantokäyttöön oli tarkoitus päästä viimeistään kesäkuun 2010 alussa. Projektille muodostui kolme erilaista tavoitemääritelmää, jotka kuitenkin liittyvät vahvasti yhteen. Järjestelmän tuli toimia tilitoimiston toiminnanohjausjärjestelmänä, työreskontrana sekä laskutusten seuranta-portaalina. Määritelmät eivät olleet yksiselitteisiä eivätkä helposti tavoitettavia, mutta onneksi Yritystilillä oli antaa määrittelyjä miten erilaiset toiminnot tulivat toimia.

Tärkeimpänä vaatimuksena oli järjestelmän tavoitettavuus. Järjestelmään piti päästä käsiksi mistä tahansa ilman erikseen asennettavia ohjelmistoja. Web-käyttöliittymä oli siis käytännössä ainoa järkevä mahdollinen vaihtoehto. Yritystili halusi oman yksilöllisen verkkoaluenimen, josta järjestelmään pääsisi käsiksi. STD tarjosi yt.std.fi -aliverkkoa omasta aliverkostaan, joka sopi hyvin Yritystilille.

### 3 Määrittely

#### 3.1 Lähtökohdat

Järjestelmän määrittelyssä syntyi seuraavanlainen yleiskuvaus:

”Töidenhallinta- ja raportointijärjestelmä BASEn tarkoituksena on toimia tehtyjen huolto ym. töiden työaikakirjanpitoa. Sähköiset työraportit toimivat laskutusperusteena asiakkaalle. Kaikki tehdyt huolto- ja räätälöintityöt saadaan helposti kirjattua järjestelmään ja näin saadaan laskutettua. Jokaisesta tehdystä työstä tehdään raportti, jonka sisältö löytyy tietokannasta raportin tekijän määrittelemillä hakusanoilla. Näin järjestelmä voi toimia minkä tahansa yrityksen asentajien, koodaajien, helpparin tai kenen tahansa muun tietokantana, josta voidaan mm. hakea ratkaisuja erilaisiin jo mahdollisesti aikaisemmin ratkaistuihin ongelmiin. Järjestelmän käyttäjätasosta riippuen erilaisten toimintojen ja sisällön laajuutta voidaan rajata.”

Sovelluksesta haluttiin alusta- sekä paikkariippumaton, jolloin ainoana vaihtoehtona pidettiin web-pohjaista palvelua. Palveluun pääsee kirjautumaan kaikilta tietokoneilta, jossa olisi nykyaikainen internet-selain sekä internet-yhteys.

#### 3.2 Tietoturva

Järjestelmän käytöstä haluttiin mahdollisimman turvallista kaikilla osa-alueilla. Yhteysprotokollana käytetään salattua HTTPS-protokollaa, jossa tiedot salataan ennen lähettämistä joko SSL- tai uudemman TLS-protokollan avulla. Std.fi -palvelimella on \*.std.fi sertifikaatti, jolla tunnistaudutaan oikeaksi palvelimeksi.

Järjestelmään kirjautumiseen vaaditaan ns. vahva salasana, eli sen pitää olla vähintään kuusi merkkiä pitkä, ja siinä täytyy olla yksi numero ja yksi iso kirjain. Tärkeä osa turvallisuutta ovat myös käyttäjäoikeudet. Töiden poisto onnistuu esimerkiksi vain ylläpitäjän käyttäjätasolla, joten työntekijät eivät pääse poistamaan vanhingossa omia tai muiden töitä. Mikäli käyttäjältä katoaa sähköt omasta tietokoneestaan kesken raportin kirjoituksen, on kesken jäänyt työ bittiavaruudessa ja se täytyy syöttää uudelleen. Palvelin on sijoitettu tunnettuun palvelinsaliin palveluntarjoajalle, jossa on kahdennetut sähköt sekä kahdennettu nettiliittymä. Mikäli palvelin menee jostain syystä sekaisin, se voidaan käynnistää etänä uudelleen. MySQL sekä LDAP-kannoista otetaan vuorokauden välein varmuuskopiot ja ne siirretään eri palvelimelle. Mikäli MySQL-tietokanta vioittuu voidaan se ladata varmuuskopiosta hyvin nopeasti takaisin. Koko palvelimen tuhoutuessa järjestelmän palauttamiseen menee luonnollisesti jo enemmän aikaa.

### 3.3 Yhteensopivuus

Järjestelmän rakennuksen vuoden 2010 alkuvaiheessa tehtiin nopea kartoitus yleisesti käytössä olevista selaimista ja todettiin, että on järkevintä tukea nykyaikaisimpia selaimia, kuten Internet Explorer 8:aa, Mozillaa Firefox 3:a, Google Chromea, Safaria ja Operaa. Suurimman käyttäjäkunnan tyydyttämiseksi järjestelmään yhteensopiviksi selaimiksi päätettiin valita Firefox 3- sekä Internet Explorer 8 -selaimet.

Määrittelyvaiheessa pidettiin mielessä, että jonkinlainen rajapinta pitää tulevaisuudessa olla vähintään laskutusohjelmiin. Ei myöskään suljettu kokonaan vaihtoehtoa, jossa osa töistä tuotaisiin jostain muusta järjestelmästä suoraan BASEen ja osa raporteista syötettäisiin BASEssa. Tämä "joku muu" -sovellus voisi olla esimerkiksi kännykässä käytettävä sovellus tai muu pitkään käytössä ollut, yrityksen toimintoihin integroitunut ERP-järjestelmä. Rajapintana voisi toimia suoraan jonkin järjestelmän sivun kutsuminen ennalta sovitulla GET/POST-parametreilla tai muu XML-pohjainen järjestely.

#### Käyttäjätasot ja roolit

Käyttäjätasoissa haluttiin käyttää rooleja erilaisina käyttöoikeuksien ja näkymien perustoina. Alusta asti oli selvillä neljä eritasoista roolia: asiakas, työntekijä, työnantaja sekä asiakas. Roolit olivat tässä tapauksessa käyttäjätasoja kuvaavia termejä. Asiakkaille haluttiin omanlaisensa näkymä, jossa pääsy olisi vain heidän yritykselleen tehtyihin töihin ja laskutustietoihin. Työntekijöille haluttiin antaa oikeus selata vain omia tekemiään töitä, lisätä raportteja kaikille mahdollisille asiakasyrityksille sekä antaa pääsy viestintään. Työnantajan oikeudet haluttiin monipuolisemmiksi. Työnantajan pitäisi päästä muokkaamaan työntekijän raportteja kokonaisvaltaisesti sekä poistamaan vielä laskuttamattomia raportteja. Roolit ovat järjestelmätasolla muokattavissa tarpeen mukaan, jolloin tietyt toiminnot voidaan lisätä tai poistaa jonkin roolin käyttäjäoikeuksista. Käyttäjätasot on suunniteltu nousevan hierarkian perusteella, eli mitä suurempi johtaja, sitä enemmän toimintoja on käytettävissä. Tämä suunnittelu estää joustavan käyttäjäkohtaisen oikeuksien muokattavuuden yksilökohtaisella tasolla, joka luetaan järjestelmän tasolla toistaiseksi miinukseksi.

### 3.4 Työntekijän toiminnot

#### Työraportin laatiminen

Järjestelmän yksi keskeisistä toiminnoista olisi työraportin syöttäminen järjestelmään. Työntekijän kirjautuessa omilla tunnuksillaan järjestelmään tulisi raportin syöttölomake olla aina ensimmäisenä näkyvillä, josta työn syöttäminen olisi yksinkertaista, helppoa ja nopeaa. Raporttiin vaadittiin taulukossa 1 esiintyvät tietueet.

Taulukko 1. Työraportin tietueet.

<b>Tietueen nimi</b>	<b>Tarkoitus</b>
<b>Päivämäärä</b>	Päivämäärä milloin työ on tehty
<b>Nimi</b>	Työn nimi
<b>Otsikko</b>	Alaotsake
<b>Selvitys</b>	Raporttiteksti
<b>Yritys</b>	Yritys mille työ tehdään
<b>Laskutettava yritys</b>	Yritys ketä työstä laskutetaan
<b>Tilaaja</b>	Työn tilaajan nimi
<b>Projekti</b>	Valittavissa mikäli projekteja luotuna
<b>Työn luonne</b>	esim. ohjelmointi, suunnittelu, ajatustyö
<b>Tehdyt tunnit</b>	Työhön oikeasti käytetty aika
<b>Laskutettavat tunnit</b>	Työstä laskutettava aika
<b>Tilatieto</b>	(kesken/valmis) jos kesken voidaan lisätä "lisärivejä" myöhemmin

Työraportin syöttämisestä tuloksena syntyvä tietue tietokantaan voi olla valmis sellaisenaan tai siihen voidaan lisätä lisärivejä. Lisärivejä työhön lisättäessä kasvavat työkonaisuuden tunnit jokaisen lisärivin myötä. Työtä, jossa on monia työraporttirivejä, voi olla esimerkiksi jokin pidempiaikainen työ, jota tehdään monena päivänä tai osissa pitkän kuukautta tai pidempään.

### 3.5 Töiden hakeminen järjestelmästä

Hakutoimintoja pidettiin tärkeänä toimintona ja sen määrittelyyn käytettiin yhdessä asiakkaan kanssa paljon aikaa. Ensimmäisessä löyhässä määrittelyssä valittiin ainoastaan asiakas, jolta töitä halutaan hakea sekä aikaväli. Tämä kuitenkin todettiin hyvin nopeasti puutteelliseksi ja epäkäytännölliseksi useimmissa tapauksissa.

Toiseen määrittelyyn käytettiin kokonaisen palaverin verran aikaa ja tutkittiin yhdessä muita verkossa toimivia hakutoimintoja vertailukohtina. Lopulta päädyttiin samantyyppiseen ratkaisuun, kuin Suomalainen Nettivene.com sekä monet sen alaiset sivut ovat

käyttäneet hakutoiminnoissaan. Käytännössä tekniset elementit haluttiin olevan <select multi="multi"> -kenttiä, joiden sisältö riippuisi käyttäjätasoista. Käytännössä näitä kenttiä olisi vierekäin kaksi. Vasemmanpuoleisessa kentässä on vaihtoehdot, joista valinnat kopioituvat oikeanpuoleiseen. Kentiksi haluttiin tarpeelliseksi nähtyjä vaihtoehtoja: asiakas, laskutettava asiakas, projekti, tekijä sekä työn tyyppi. Näiden lisäksi piti pystyä hakemaan tietyssä tilassa olevia töitä: laskutetut, laskuttamattomat, valmiit sekä keskeneräiset työt. Näillä valinnoilla pystyttäisiin hakemaan joustavasti töitä. Kuvassa 1 on ensimmäinen mallinnus hakutoiminnosta.

Hae aikaväliltä  
 2011-03-21 00:00 - 2011-03-23 23:59

Laskuttamattomat

Laskutetut

Kesken

Valmiit

Asiakas

Asiakas1 Asiakas3  
 Asiakas2  
 Asiakas3  
 Asiakas4

Kuva 1. Suunnitelma hakutoiminnosta.

### 3.6 Pääkäyttäjän ja laskuttajan toiminnot

#### Töiden muokkaus

Keskeneräisten töiden muokkaus haluttiin olevan mahdollista myös työntekijätasolla, mutta eritoten laskutuksessa haluttiin vielä valmiidenkin töiden muokkauksen olevan mahdollista. Töitä haluttiin voida muokata kokonaisvaltaisesti, mikä tapahtuu töiden haun jälkeen työn kohdalla olevan muokkaustoiminnon kautta. Tuntien muokkaus onnistuu napsauttamalla tuntimäärää ja muuttamalla tuntimäärän haluamukseen sivulle avautuvassa ponnahdusnäkylässä, kuten kuvassa 2 on esitetty. Töiden kokonaisvaltaisessa muokkauksessa pystyy muokkaamaan kaikkea muuta tietoa poislukien asiakastietoja. Muokkauksesta jää aina jälki, milloin työtä on muokattu, ja se näkyy työhaulistauksessa.

The image shows a web application interface with a modal dialog box open. The dialog is titled "Muokkaa laskutettuja tunteja" (Edit billed hours). It contains two input fields: "Laskutetut tunnit" (Billed hours) and "Tehdyt tunnit" (Actual hours), both of which have the number "1" entered. Below the input fields are two buttons: "Tallenna" (Save) and "Peruuta" (Cancel). The background shows a table with columns for "Valmis ID", "Tilaaja", "Tehty", "Lisätty", "Muokattu", and "Poista:". The table rows are partially visible, showing "Valmis ID=764" and "Valmis ID=70".

Kuva 2. Työtuntien muokkaus.

## Laskutus

Laskutusta mietittäessä haluttiin, että työt voi merkitä laskutetuiksi, minkä jälkeen laskutettavat työt tulisi niputtaa jollain loogisella tunnuksella toisiinsa, jotta niiden haku asiakkaan nimellä olisi mahdollista. Lopulta päädyttiin generoimaan jokaiselle laskutettavalle työkokonaisuudelle oma ID. ID:n nimeksi tuli APE-ID. APE-ID on viisimerkkinen merkkisarja, joka generoidaan jokaiseen laskuun erikseen. Tällä ID:llä voidaan hakea ja listata laskuja jälkikäteen, mikä helpottaa huomattavasti laskutustoiminnassa.

Laskutuksen seuranta varten haluttiin ehdottomasti taulukko, jossa olisi listattuna allekkain kaikki asiakkaat ja sarakkeina laskuttamattomat, laskutetut ja keskeneräiset työt. Taulukkoa paranneltiin myöhemmin ottamalla käyttöön aikavälin ulkopuolelle jäävät työt, jolloin esimerkiksi haun päivämäärillä 1.1.2010 – 31.1.2010 ulkopuolelle jäävät työt näkyvät taulukon sarakkeissa "laskuttamattomat aikavälin ulkopuolella" sekä "laskutetut aikavälin ulkopuolella", kuten kuvassa 3 huomataan. Tällä toiminnolla varmistetaan se, että jos jokin työ jää syystä tai toisesta laskuttamatta, sitä ei unohdeta laskuttaa myöhemmin, koska se ilmestyy aikavälin ulkopuolelle jäävään sarakkeeseen.

Hae aikaväliltä  
18.04.2011 - 02.05.2011

Aikavälillä 18.04 - 02.05 Kaikki yhteensä

Asiakas	Laskutettu	Laskuttamatta	Kesken	Laskutettu	Laskuttamatta	Kesken
Firma A [+]	0	2	0	55	12	4
Firma B [+]	0	0	0	38	11	1
Firma C [+]	0	0	0	0	1	0
Firma D [+]	0	0	0	0	15	2
Firma E [+]	0	0	0	0	1	0
Firma F [+]	0	0	0	0	31	4
6 riviä	0	0	0	0	0	0

Kuva 3. Laskutustaulukko.

Käytännössä laskutus tapahtuu hakemalla työhaulla halutulta asiakkaalta joltain tietyltä aikaväliltä laskuttamattomat työt tai napsauttamalla "laskutettavat työt" -taulukosta "laskuttamattomat työt" -sarakkeessa olevaa numeroa. Haun jälkeen työt ovat listattuna ruudulla, jonka jälkeen laskuun haluttavat työt valitaan merkitsemällä ruksi vasemmassa laidassa olevaan <checkbox> -elementtiin. Kun tarvittavat työt on valittu, haetaan sivun alalaidassa nappi "laskuta työt", jonka jälkeen BASE generoi APE-ID:n sekä laskun loppusumman laskua varten. Tämän jälkeen summa sekä APE-ID kirjoitetaan laskun viestikenttään ja lasku voidaan lähettää asiakkaalle.

#### Asiakkaiden ja työntekijöiden lisääminen

Asiakkaiden ja työntekijöiden lisääminen tapahtuu ylläpitomoduulin kautta. Asiakkaat välilehdellä on listattuna kaikki jo järjestelmään syötetyt asiakkaat ja listauksen alla on lomake uuden asiakkaan lisäystä varten. Asiakasta varten kerätään taulukon 2 mukaisia tietoja.

Taulukko 2. Asiakastietueet.

<b>Tietueen nimi</b>	<b>Tarkoitus</b>
<b>yrietyksen nimi</b>	
<b>asiakasnumero</b>	
<b>y-tunnus</b>	
<b>yhteyshenkilö</b>	
<b>YH puhelin</b>	
<b>YH email</b>	
<b>Lähiosoite</b>	
<b>postinumero</b>	
<b>kaupunki</b>	
<b>puhelin2</b>	
<b>laskutus lähiosoite</b>	
<b>laskutus postinume- ro</b>	
<b>laskutus kaupunki</b>	
<b>fax</b>	
<b>www-osoite</b>	
<b>email2</b>	
<b>toimiala</b>	
<b>yh1,yh2,yh3,yh4,yh5</b>	Mahdolliset ennalta määrätty yhteyshenkilöt omasta yrityksestä asiakkaalle. Näkyvät asiakkaan etusivulla puhelinnumeroineen.
<b>vakiotilaaaja</b>	Asiakasyrityksen tunnus, joka on vakiona kaikissa töissä tilaajana ellei muuta määritely
<b>vapaa kenttä</b>	Vapaan kentän sisältö tulee esiin kun asiakasta laskutetaan. Kentässä voidaan pitää esimerkiksi asiakaskohtaisia hintoja tms. Ei näy asiakkaalle.

Asiakkaan lisäämisen jälkeen tulee lisätä kyseiselle asiakkaalle käyttätunnuksia. Tunnuksia voidaan tehdä niin monta kuin halutaan. Jokin asiakkaalle tehdyistä tunnuksista liitetään asiakkaan vakiotilaaajaksi, jolloin hänen nimensä liitetään tilaajaksi kaikkiin kyseiselle asiakkaalle tehtyihin raportteihin, ellei muuta erikseen määritely.

Työntekijän lisäys tapahtuu Käyttäjät-välilehdeltä. Käyttäjän lisäämisessä valitaan taulukossa 3 esitetyt tietueet.

Taulukko 3. Käyttäjän tietueet.

Tietueen nimi	Tarkoitus
<b>käyttäjätunnus</b>	tunnus millä kirjautudutaan järjestelmään
<b>salasana</b>	salasana
<b>etunimi</b>	
<b>sukunimi</b>	
<b>matkapuhelin</b>	
<b>sähköposti</b>	
<b>lisätieto</b>	vain admintasolle näkyyvä lisätieto
<b>käyttäjätaso</b>	asiakas,työntekijä,laskuttaja,yritysadmin
<b>asiakasliityntä</b>	liitetäänkö tunnus johonkin asiakkaaseen

Käyttäjän lisäyksen yhteydessä tulee määritellä käyttäjän oikeudet valitsemalla oikea käyttäjätaso. Käyttäjätasot ovat ns. roolitettuja, eli käyttäjän lisääjän ei tarvitse miettiä erikseen, mitä sivuja tai toimintoja jokin käyttäjä voisi tarvita tai saada nähdä. Käyttäjien oikeuksia ei voi muokata käyttäjäkohtaisesti. Käyttäjätasoksi voi valita yhden seuraavista:

- Asiakas
- Työntekijä
- Laskuttaja
- Järjestelmänylläpitäjä.

#### Viestintä

BASEen haluttiin myös viestintätoiminto. Ensimmäisessä vaiheessa ei nähty tarpeelliseksi käyttäjien välisiä viestejä, vaan ainoastaan asiakkaan ja toimittavan yrityksen väliset viestit. Viestintä tapahtuu erillisestä moduulista, jossa ei ole muita toimintoja asiakkaalle kuin viestin lähetys ja saapuneiden lukeminen. Yritysylläpitäjä voi viestinnän kautta antaa omille työntekijöille työmääräyksiä jonoon, jotka sitten työntekijät käyvät kuittaamassa itselleen viestintämoduulin kautta. Asiakkaan lähettämä viesti voi olla myös työmääräys, joten kaikki viestit voidaan ottaa työksi. Työksi ottaminen kopioi viestistä oleelliset tiedot. Järjestelmä luo uuden työn ja nimeää sen työn ottajan käyttäjätunnukselle. Tällaisen työn tilaksi tulee aina "kesken" sekä työtunneiksi 0 h.

## Lisätoiminnot

Laskutusta varten suunniteltiin jälkikäteen muutamia lisätoimintoja, jotka nopeuttavat laskutusta. Koska laskut normaalisti lähetetään ikkunallisissa kirjekuorissa, oli tarve saada prosessin nopeuttamiseksi asiakkaat välilehdelle toiminto, jolla voidaan tulostaa kirjekuoren sisälle sopiva puoliksi taiteltu A4-kokoinen liuska asiakkaan yhteystiedoilla. Tätä varten haluttiin nappi, josta saisi kaikkien asiakkaiden yhteystiedot samalla kertaa tulostettua.

## Tiedostot

Ylläpitäjätason käyttäjä voi halutessaan liittää asiakaskohtaisesti tarpeelliseksi näkemiin dokumentteja, jotka asiakkaat voivat omasta käyttöliittymästään helposti avata tai tallentaa omalle tietokoneelleen. Kyseiset dokumentit voivat olla esimerkiksi asiakaskohtaisia hinnastoja tai muita tärkeitä dokumentteja. Toistaiseksi ei nähty tarvetta antaa asiakkaan lisätä tiedostoja itse. Tiedostot löytyvät asiakkaan omalta etusivulta alimmaisena.

### 3.7 Asiakkaan toiminnot

#### Etusivu

Järjestelmään sisään kirjatessaan asiakas ohjataan etusivulle, jossa häntä tervehditään omalla käyttäjänimellä, mitä seuraa viimeisin voimassa oleva tiedote. Etusivulta haluttiin löytyvän myös omien yhteyshenkilöiden tiedot, omat tiedot sekä tiedostot. Ensivaikutelman haluttiin olevan kirjautumisen jälkeen mahdollisimman henkilökohtainen ja tarjoavan riittävästi ajankohtaista tietoa, joka saisi käyttäjän kiinnostumaan enemmän järjestelmästä.

## Laskujen hakeminen

Järjestelmän yksi keskeisimmistä toiminnoista on tarjota asiakkaalle reaaliaikaista töiden seuranta. Kirjatumalla järjestelmään näkee aina mitkä työt ovat kesken ja mistä töistä on seuraavaksi tulossa lasku. Asiakas pystyy tällä tiedolla arvioimaan, kuinka paljon kustannuksia tulee seuraavassa laskussa. Laskun saapuessa postilaatikkoon luetaan laskusta laskutus-ID ja kirjoitetaan se laskutusten yhteenveto-välilehdellä hakukenttään. Haun jälkeen asiakkaan ruudulla on listattuna täysin samat työt, kuin laskuttaja on laskuttanut. Tällä tavalla vältetään mahdolliset epäselvyydet laskujen koostumuksista.

### 3.8 Raporttien tulostustoiminnot

Kaikki työlistaukset haluttiin saada tulostettua myös tarvittaessa paperille sekä tallennettua jollakin tavalla liitetiedostoksi esimerkiksi sähköpostia varten. Tulostus voitaisiin jättää esimerkiksi selaimen huoleksi ja tarjota vain ylimääräinen ikkuna, jossa ei näy sivun ulkoasun muita elementtejä, kuten valikoita. Liitetiedostoksi harkittiin ensin OpenOfficen spreadsheet-muotoista vientitiedostoa, joka hylättiin, koska siinä ei saatu samannäköistä työlistausta aikaiseksi kuin alkuperäinen näkymä sivulla on. Nopea tutkimus internetistä osoitti, että on olemassa jotakin avoimen lähdekoodin kirjastoja, joilla saa muodostettua PDF-tiedostoja. PDF on ajatuksena hyvä, koska sen saa tallennettua, se on yleinen tiedostomuoto ja se on tulostettuna aina täysin samannäköinen, kun miltä se näyttää tietokoneen ruudulla katsottuna.

## 4 Suunnittelu

### 4.1 Verso-kehysrakenne

Määrittelyn valmistuttua tultiin vaiheeseen, jossa piti tehdä valintoja, minkälaista tietä on aikomus edetä kohti yksinkertaista ja vauhdikasta toteutusvaihetta. Toteutuksen kannalta olisi helpointa valita sovellukselle jo olemassa oleva rakenne, jota muokkamalla saisi valikot oikeille kohdilleen. Käytännössä vaihtoehdot olivat aloittaa käyttöliittymän tekeminen tyhjästä, valita jokin STD Systemsissä valmiiksi käytössä oleva kehysrakenne tai valikoida jokin avoimeen lähdekoodiin pohjautuva rakenneratkaisu, kuten Joomla tai ZEND [16,17]. Käyttöliittymän rakenteelliseksi pohjaksi valikoitui lopulta STD Systemsin viimeaikaisin kehysrakenne, joka oli saanut kehitysvaiheessa nimekseen Verso-kehysrakenne.

Verso tarjoaa moduulipohjaisen valikkorakenteen. Valikkorakenne on täysin dynaaminen ja mukautuu jokaiselle käyttäjälle erikseen automaattisesti käyttöoikeuksien sekä näkävyyksien mukaan. Versossa on käytetty käyttöoikeuksiin sekä valikkorakenteen luontiin LDAP-tekniikkaa. Verso ei ole varsinaisesti julkaisujärjestelmä, vaan rakenteellinen pohja, jonka päälle järjestelmä voidaan projektikohtaisesti rakentaa ja tarpeen tullen laajentaa. Verso tarjoaa sisäänrakennetun käyttäjähallinnan ja tarjoaa täten siihen lisättäville moduuleille erilaisia tunnistautumispalveluita, jotka muuten pitäisi rakentaa jokaiselle moduulille erikseen.

Verso oli ideaalinen valinta moduulipohjaisuutensa takia. Jokainen käyttäjätaso saadaan omaksi moduulikseen, jolloin esimerkiksi työntekijätason näkymään ei edes tule hallintaosiota ollenkaan, mutta ylläpitäjätasolle voidaan lisätä esimerkiksi laskutustoinnit töiden hallintamoduuliin. Kuvassa 4 sisäänkirjautuneena on ylläpitäjätason käyttäjä, jolla on oikeudet kaikkiin moduuleihin ja kaikkiin moduulien sivuihin.

Yritystili.fi  
BASE Työraportointi

Työraportti Työhaku Töiden selaus Omat keskeneräiset Laskutus yhteenveto Viimeisimmät työt Omat työt

Erkki Esimerkki

Lisää suosikkeihin

Etusivu

Viestintä

Ylläpito

Työt

Lisää uusi työraportti

Pvm  
16.04.2011

Nimi

Otsikko

Selvitys tehdystä työstä

Yritys  
Valitse yritys

Tilaaaja  
Valitse Tilaaaja

Laskutettava yritys  
Valitse yritys

Valitse työn luonne  
Valitse työn luonne

Tehdyt

Tilatieto  
 Kesken  
 Valmis  
 Tallenna projektiksi

Laskutettavat

TALLENNA \* = vaadittavat kentät

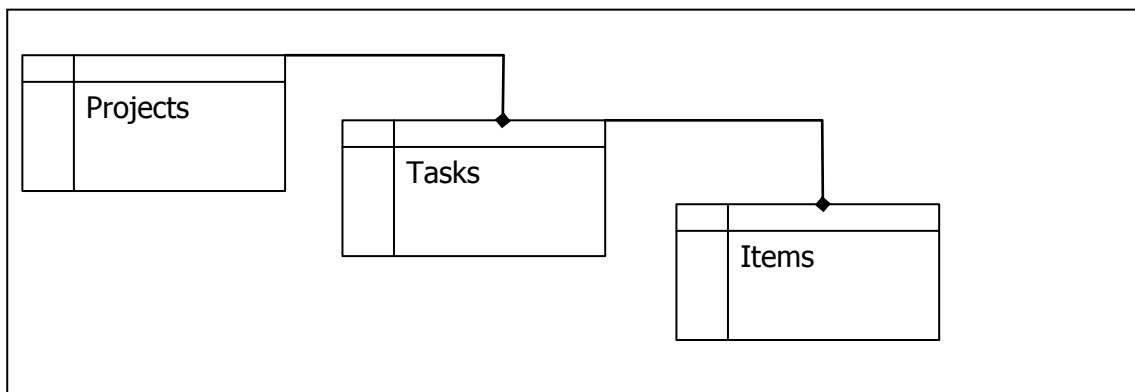
Kuva 4. Verso-kehysrakenteen moduulijako sekä työn syöttölomake.

## 4.2 Tietokanta

STD Systems Oy:n kaikissa projekteissa käytetään MySQL-tietokantaa, joten oli luonnollista pitäytyä tutussa ja turvallisessa ympäristössä. Tietokannan suunnittelu lähti peruskysymyksistä, kuten kuinka isoja tietomääriä tauluihin on mahdollisesti tulossa sekä pitääkö viite-eheydestä huolehtia tarkemmin kuin yleensä. Mikäli vastaus jompaankumpaan edellä mainittuun kysymykseen on positiivinen, tulee MySQL-tietokannan koneeksi (engine) valita InnoDB. Kannan suunnitteluvaiheessa todettiin tietomäärien mahdollisesti kasvavan huimiin määriin työraporttien lisääntyessä, joten lopulta päädyttiin InnoDB:n käyttöön.

Tietokannan rakenteen puolesta Verso-framework pakotti lähtökohtaisesti tiettyjä tauluja käyttöön, joiden ympärille suunniteltiin toimiva kokonaisuus. Rakennetta suunniteltaessa piti ottaa huomioon kaikki määrittelyssä esiintyvät tekijät sekä jättää varaa vielä

tuntemattomille tekijöille. Raportoinnin osalta rakenteesta tuli lopulta melko yksinkertainen pääpainon kohdistuessa tehtävä- sekä nimike-tauluihin. Rakenne on esitelty näiden taulujen osalta kuvassa 5. Merkistöksi valittiin STD Systems Oy:n muissakin tietokannoissa hyväksi ja yhteensopivaksi todettu UTF8 sekä lajitteluperusteeksi utf8\_swedish\_ci. UTF8-merkistöllä pystytään esittämään kaikki Unicode-merkistön merkit, sekä se on taaksepäin yhteensopiva ASCII-merkistön kanssa [1].



Kuva 5. Raporttien taulusuhteet.

## 5 Toteutus

### 5.1 Web-sovellustekniikat

Perinteisesti ajatellen Web-sovellukset ovat kokonaisuudeksi saatettu kokoelma erilaisia tekniikoita. Sovelluksen ytimenä toimii HTTP-palvelinohjelmisto, jonka kautta saadaan yhteys asiakkaan selaimelta sovellukseen. Yleisimmin käytetyt HTTP-palvelinohjelmistot ovat Apache ja Microsoftin IIS (Internet Information Services). Kaikki www-sivut näkyvät selaimelle HTML-kuvauskielenä [10], sekä lisäksi erilaisina asiakaspuolen tulkattavina komentosarjoina. Erilaisia tulkattavia komentosarjakieliä on Javascript, Active-x, VBScript, Jscript ja Flash Actionscript [9]. Lomakkeiden sekä erilaisten URI-parametrien käsittelyyn voidaan nykypäivänä käyttää useita erilaisia palvelinpään komentosarjakieliä kuten PHP, Microsoftin .NET ASP (Active Server Pages), Perl, Python tai Javaa. Valitan näiden välillä on yleensä kiinni ohjelmoidan kokemuksesta kyseiseen tekniikkaan sekä mahdolliset rajapinnat muihin sovelluksiin, joissa saatetaan asettaa vaatimuksia palvelinpään skriptikielelle. [6.]

Sivujen ulkoasua kuvataan usein CSS (Cascading Style Sheet) -kielen [10] avulla, jolla luodaan säännöstö, joka ehdottaa selaimelle miten sisältö tulisi näyttää asiakkaalle[7]. Koska sovelluksen tulee tarvittaessa tallentaa suuria määriä tietoa tarvitaan tietokanta. Suosituin tietokanta on MySQL, joka on ilmainen. Isoissa yrityksissä käytetään usein myös maksullista Oraclen [8] tietokantaa, joka on suunnattu enemmän alan ammattilaisille ja valtaville tietomäärille. Käyttöoikeuksien hallintaa varten joissain järjestelmissä on käytössä LDAP (Lightweight Directory Access Protocol), jossa tiedot ovat tallennettu puumaiseen rakenteeseen avain-arvopareina.

Erilaisten tekniikoiden valinta eri projektin osa-alueille oli perusteltua ja lopulta niistä koostui onnistunut kokonaisuus. Palvelinohjelmaksi valittiin ilmaisuuden, vakaan toiminnan ja monipuolisen moduulivalikoimansa takia Apachen HTTP-palvelinohjelma. Apache toimii hyvin niin Linux- kuin Windows-ympäristössä, mutta kustannus palvelimen Windows-käyttöjärjestelmän osalta olisi noussut huomasti ilman minkäänlaista lisähyötyä. Kustannussyistä johtuen palvelimen käyttöjärjestelmäksi valittiin Linuxin

Ubuntu-jakeluversio. Ubuntun valinnan jälkeen palvelimen komentosarjakielistä varteenotettavin vaihtoehto oli PHP, koska ei ollut mitään vaatimuksia, jotka olisivat vaatineet Pythonin, Perlin tai Javan käyttöä. Projektin aikataulua myös silmälläpitäen olisi ollut haasteellista valita ennestään täysin tuntematon komentosarjakieli. PHP-moduuli Apacheen sekä itse PHP-kirjastot löytyvät suoraan Ubuntun paketinhallinnasta. Asiakaspään komentosarjakeleksi valittiin Javascript yksinkertaisesti Mootools-kirjaston käyttöä ajatellen. Mootools tarjoaa yksinkertaisen ja monen selaimen kanssa yhteensopivan kirjaston, jolla Ajax-pyyntöt ja sivujen dynaamiset muokkaukset onnistuvat. Ver-so-frameworkista johtuen paketissa on mukana myös LDAP.

## 5.2 PHP

PHP tulee sanoista **PHP: Hypertext Preprocessor**. PHP on laajalti käytetty yleiskäyttöinen vapaan lähdekoodin komentosarjakieli. PHP soveltuu erityisen hyvin web-kehitykseen, ja se voidaan sisällyttää olemassaolevaan HTML-koodiin. PHP on tulkattava kieli ja se ajetaan palvelinpäässä, jonka takia asiakaspään selaimella ei voi saada mitenkään selville varsinaista PHP-koodia. PHP:tä tukevat laajalti niin Apache, Microsoft Internet Information Server, Personal Web Server, Netscape ja iPlanet servers, Oreilly Website Pro server, Caudium, Xitami, OmniHTTPd ja monet muut palvelinohjelmistot. PHP:tä ei käytetä ainoastaan web-palvelimissa vaan sitä voidaan ajaa myös sellaiseenaan suorittamaan esimerkiksi päivittäisiä rutiineja Linux-ympäristössä. [2.]

## 5.3 MySQL

GPL-lisensointiin perustuvat MySQL on maailman suosituin avoimeen lähdekoodin perustuva tietokanta. Vuonna 1996 julkistetun ensimmäisen version takana ovat suomalainen Michael "Monty" Widenius sekä ruotsalainen David Axmark. Nykyään MySQL:n uusin versio on 5.5, johon sisältyy viite-ehyteen liittyvät toiminnot (foreign-key) sekä transaktiotuki. MySQL:n valinta tietokannaksi oli perusteltua ilmaisuutensa sekä aikaisemman kokemuksen takia. Muita vaihtoehtoja olisi ollut Oracle tai PostgreSQL. [3.]

## 5.4 LDAP

LDAP (Lightweight Directory Access Protocol) on kevyt verkkoprotokolla, joka on alun perin suunniteltu korvaamaan monimutkainen X.500-hakemistopalvelu. Tiedot talletetaan hakemistopuuhun avain-arvopareina. LDAP ei ole täysiverinen tietokanta ja siksi sitä käytetään enimmäkseen harvakseltaan muuttuvien tietojen, kuten käyttäjätietojen säilyttämiseen ja hakemiseen.

Syy LDAP:n käytölle projektissa on Verso-frameworkkiin syväle sisällytetty käyttäjätunnistusemekaniikka, jonka taustalla käytetään LDAP-protokollaa. Projektin alussa suunniteltiin Verson LDAP tietueiden laajentamista tarpeiden mukaan, mikä kuitenkin keskeytettiin varsin aikaisessa vaiheessa tietueiden lisäämisen hitauden ja monimutkaisuuden takia. Projektin edetessä suurin osa aikaisemmin LDAP:ssa olleista tietueista oli siirretty MySQL-tietokantaan LDAP:n ja MySQL:n yhteiskäytön hankaluuden vuoksi ja ainoastaan rungon toimivuuden vuoksi pakolliset tietueet ovat jääneet LDAP-taustaan.

## 5.5 Mootools

Javascript tarjoaa mahdollisuuden muuttaa sivun sisältöä dynaamisesti ilman sivun uudelleenlatausta, mutta käytännössä helpon ja nopean siitä tekee avoimeen lähdekoodiin perustuva Mootools-kirjasto. Mootools yksinkertaistaa ja nopeuttaa HTML-dokumenttien muokkaamista sekä Ajax-pyyntöjen käsittelyä. Mootools-kirjaston suurin vahvuus on sen yhteensopivuus eri selainten kanssa, minkä ansiosta ei tarvitse tehdä erilaisia versioita eri selaimia varten.

BASE-työraportointijärjestelmässä kaikki Ajax-pyyntöt ja niiden käsittelyt on toteutettu Mootools-kirjastolla. Ajax-pyyntöt ovat selainohjelman taustalla tapahtuvaa asynkronista tiedonsiirtoa palvelimen kanssa, jottei koko verkkosivua tarvitse ladata uudelleen [20]. Myös erilaisia käyttövyöhykettä parantavia visuaalisia tehosteita sekä DOM-puun muokkauksia on luotu käyttäen Mootools-kirjastoa.

## 5.6 HTTP-palvelin

Kaikenlaisten dokumenttiin siirtoon tarkoitettu HTTP (HyperText Transfer Protocol) -protokolla on yleisesti käytetty yhteyshäytntö. Yleisesti protokollaa käytetään web-palvelimissa, mutta sitä voidaan käyttää muuallakin. Normaali web-liikenne ohjautuu porttiin 80, mutta se voidaan myös HTTP-palvelimen asetuksista muuttaa mihin tahansa muuhun porttiin. Selaimen ja palvelimen välinen keskustelu tapahtuu tekstimuotoisesti, jota myös ihminen voi ymmärrettävästi lukea niin tarvittaessa. Keskustelussa selain lähettää pyyntöjä HTTP-palvelimelle ja tulkitsee vastauksia ja näyttää ne käyttäjälle. (19, s. 204-205.)

### Apache

Ensimmäinen virallinen julkaisu nykyäänkin täysin ilmaisesta ja maailman suosituimmasta HTTP-palvelinohjelmasta Apache oli huhtikuussa 1995 julkaistu versio 0.6.2. NCSA:ssa (National Center for Supercomputing Applications) työskentelevän pieni ryhmä perusti vuonna 1999 Apache Software Foundationin, jonka tarkoitus oli tuoda organisaatiollinen, laillinen ja rahallinen tuki Apache HTTP -palvelinohjelmistolle. ASF onnistuikin asiassa varsin hyvin ja on tuonut vankan ja turvallisen ympäristön tulevaisuuden kehitykselle. [11.]

## 5.7 Palvelinkone

Palvelinkone on järjestelmän ydin, jossa pyörii vähintään HTTP-palvelinohjelmisto, mutta hyvin usein myös samaan fyysiseen tietokoneeseen on asennettu tietokantaohjelmisto. Palvelinkone eroaa tavallisesta PC-tietokoneista yleensä niin koon kuin ominaisuuksienkin puolesta. Palvelinkoneet ovat yleensä litteitä, ja ne asennetaan palveluntarjoajan tiloihin ns. "räkkikaappiin". Räkkipaappi on standardisoitu (EIA 310-D, IEC 60297 ja DIN 41494 SC48D) 19-tuuman levyinen laitekaappi, jonne asennetaan palvelinkoneita. Yleinen palvelimen laitetekoko on 1U (korkeus 44,25 mm). Palvelinkoneiden vikasietoisuutta parannetaan kahdennetulla virtalähteellä, kahdennetulla internetyhteydellä, RAID-levyjärjestelmällä ja ECC-virheenkorjaavalla keskusmuistilla. [5.]

Palvelinkone piti valita sinne tarvittavan ohjelmiston ja arvioidun käyttökuorman perusteella. HTML-palvelinohjelmaksi valitun Apachen toiminta onnistuisi niin Windows- kuin Linux-palvelimillakin. PHP eikä MySQL aseta sellaisia vaatimuksia, jotka kallistaisivat päätöstä Windowsin puolelle käyttöjärjestelmän suhteen, joten ei nähty mitään syytä, miksi pitäisi hankkia palvelinkoneeseen Windows-käyttöjärjestelmä. Palvelinkonetta ei haluttu hankkia omaksi, jolloin vuokrapalvelin oli jäljelle jäävä vaihtoehto. Palvelinkoneeksi valittiin uusi vuokrapalvelin, joka sijaitsee nimeltä mainitsemattoman palveluntarjoajan tiloissa. Palvelinkoneeseen valittiin käyttöjärjestelmäksi Ubuntu 10.04 LTS (Long Term Support). Kyseinen versio valittiin, koska se oli uusin Ubuntu-pitkäaikaisella varustettu käyttöjärjestelmä. Palvelimeen ei ole fyysistä pääsyä, vaan yhteys palvelimeen muodostetaan ainoastaan SSH-yhteyden avulla. SSH-yhteyden avulla palvelin konfiguroidaan ja tarvittavat ohjelmat käynnistetään. Tiedostojen siirtoa varten käytetään SFTP-yhteyttä.

## 5.8 SortableTable

SortableTable on nimensä mukaan lajitteltava taulukko. SortableTable on toteutettu PHP-luokkana, mutta se sisältää myös jonkin verran Javascriptiä. SortableTable on STD Systems Oy:n ehkä eniten käytetty ohjelmistokomponentti, ja sitä kehitetään edelleen jatkuvasti tarpeiden mukaan. SortableTablen avulla yksinkertaistetaan erilaisten taulukoiden tekoa sekä samalla kaikista taulukoista saadaan ominaisuuksiltaan samanlaiset. SortableTablen käytöllä säästetään myös aikaa monimutkaistenkin taulukoiden teossa eikä koodia tarvitse monistaa turhaan useille eri sivuille. SortableTablen perusominaisuuksiin kuuluu taulukon järjestäminen sarakkeen sisällön mukaan laskevasti tai nousevasti, tarpeettomien kenttien piilottaminen sekä kenttien suodatus sisällön mukaan. Suodatus voidaan erikseen kytkeä sellaisiin kenttiin, joissa sitä erikseen tarvitaan.

SortableTablea käytetään järjestelmässä kaikkialla muualla, paitsi töiden listauksessa, johon perinteinen taulukkomalli ei soveltunut. Vastaavanlaista komponenttia ei ole saatavilla avoimen lähdekoodin järjestelmästä, joten SortableTable oli todella projektia nopeuttava ohjelmakomponentti. Kuvassa 6 on esimerkki, miltä töiden listaaminen sivulla olisi näyttänyt SortableTablella.

Yritys	ID	Projekti	Työnimi	Otsikko	Tekijä	Selvitys	Tehty	Lisätty	Muokattu	Tila	Lasketettu	T.Tunnit	L.Tunnit	
STD Systems Oy	1094	Oletus	Intra bugikorjaukset toukokuu 2011	palkkaraportti	Tomi Nousiainen		02.05.2011	02.05.2011	01.01.1970	valmis	<input type="checkbox"/>	1	1	
STD Systems Oy	1092	Oletus	Intra bugikorjaukset toukokuu 2011	Paikannus	Tomi Nousiainen		02.05.2011	02.05.2011	01.01.1970	valmis	<input type="checkbox"/>	3	3	
STD Systems Oy	1096	Oletus	Inssiduuni	Viimeistelyä	Tomi Nousiainen		02.05.2011	02.05.2011	01.01.1970	valmis	<input type="checkbox"/>	3	3	
<b>3 riviä</b>		<b>3282</b>											<b>7</b>	<b>7</b>

Valitse kaikki   Laskuta valitut

Kuva 6. Työlistaus SortableTablella.

## 5.9 BaseCore-luokka

Projektin alussa määrittelyn mukaisesti jokaiselle toiminnolle tehtiin oma PHP-sivu, joka toimi vallan mainiosti. Mitä pidemmälle projektia edettiin, huomattiin samojen työlistauksien toteutuvan useassa paikassa, mutta kuitenkin vähän erilaisten toimenpiteiden tuloksena. Työlistaukseen tuli jatkuvasti muutoksia sisällöllisesti sekä tyyllisesti. Jokaisen koodiin tehdyn muutoksen jälkeen samat koodirivit oli kopioitava kaikille mahdollisille sivuille. Muutosten lisääntyessä kopiointityö alkoi viedä paljon aikaa. PHP:ssa on mahdollisuus tehdä omia luokkia kaikkien olio-ohjelmointikielien tapaan, joten tässä vaiheessa päädyttiin tekemään työkaluluokka BASEa varten. Ajatuksena oli kerätä luokkaan toimintoja, joita tarvittaisiin mahdollisimman monessa paikassa. Ensimmäisenä luokkaan haluttiin toteuttaa työlistauksia tulostava funktio. Funktiota oikeilla parametreilla kutsuttuna sitä voitaisiin käyttää kaikissa mahdollisissa listatyypissä näkymissä.

Luokka sai nimekseen BaseCore. BaseCore toteuttaa nykyisellään reilut parikymmentä funktiota, joista viisi on erilaisiin tarpeisiin rakennettuja työlistauksien tulostusfunktioita.

### BaseCore-luokan käyttö

Ensimmäiseksi BaseCore-luokka sisällytetään projektiin PHP:n include-funktiolla:

```
include("base_core.php");
```

Sisällyttämisen jälkeen BaseCore-luokkaa voidaan käyttää aivan normaalisti PHP:n luokkamallin mukaisesti, kuten kuvassa 7 on esitetty.

```

229     $base = new base_core();
230     $base->setSQL(
231         $yid,
232         $_REQUEST['sdate'],
233         $_REQUEST['edate'],
234         null,
235         null,
236         null,
237         $projects,
238         null,
239         array($_REQUEST['customer_id']),
240         null,
241         null,
242         $filters,
243         $_REQUEST['outside']
244     );
245     //echo $base->returnStaticSQL();
246     echo $base->get_NormalWorkList($_REQUEST['adminlevel']);

```

Kuva 7. BaseCore-luokan käyttö.

BaseCore-luokassa haluttiin jo itsessään säästää vaivaa ja aikaa eriyttämällä tietokantaan kohdistuvan kyselyn rakentamisfunktio sekä tulostamisfunktiot. Määrittelyssä esiintyvien näkymien perusteella voitiin huomata, että tulee tarve toteuttaa erityyppisiä työlistauksia riippuen käyttäjätasosta. Jokaista erilaista työlistausta varten tulisi toteuttaa oma funktio, mutta samalla voitaisiin kaikissa näissä funktioissa käyttää samaa funktiota, jolla MySQL-kysely luotaisiin. Tämän vuoksi ennen tulostusfunktioita kutsutaan aina setSQL(..)-funktioita, jolle annetaan parametreina kaikki tarpeellinen tulostusfunktion toimintaa varten.

Funktiot ovat BaseCore-luokassa nimetty sitten, että mitä tahansa palauttavat funktiot alkavat "get"- ja muut toiminnalliset funktiot alkavat "set" -merkkijonoilla. Insinööriyötä kirjoitettaessa BaseCore luokkaan on kertynyt koodia jo reilut 2000 riviä. Voidaankin melkein sanoa, että järjestelmän koko järki on BaseCore-luokassa ja muualla on vain pakollisia rakenteellisia toimintoja, jotta järjestelmästä saadaan käyttökelpoinen.

BaseCore-luokka on säästänyt aikaa ja vaivaa koko projektinkin näkökulmasta ja on harmillista, että sitä ei ymmärretty lähteä toteuttamaan heti projektin alkaessa. Yksi mielenkiintoisimmista haasteista oli yrittää keksiä jokin keino, miten näytölle tulostettu listaus voitaisiin "muistaa" täydellisenä käyttäjän halutessa tulostaa sen paperille täysin samanlaisena kuin se on näytöllä esitetty. Ensimmäisenä vaihtoehtona kokeiltiin, että käyttäjän on täytettävä hakulomake uudelleen ja painettava eri nappia kuin näytölle listauksen halutessaan. Tämä ei kuitenkaan ollut missään määrin käytännöllistä, joten vaihtoehto ei tullut kyseeseen. Vastaus tulostukseen oli todella yksinkertainen ja käyttäjälle täysin näkymätön. Ennen tulostamista BaseCore-luokassa täytyy asettaa SQL-

parametrit kuvan 8 mukaan, jossa ne talletaan palvelimen muistiin käyttäjäkohtaisesti \$\_SESSION-muuttujassa. Käyttäjän napsauttaessa tulostus-nappulaa kutsutaan Ajax-requestin kautta PDF:n luontifunktiota, joka käyttää \$\_SESSION-muuttujassa olevia arvoja. Tulostus esitetty kuvassa 9.

```

99 function setSQL($company_id=NULL,$sdate=NULL,$edate=NULL,
100                $apeid=NULL,$outerSearch=NULL,$wid=NULL,
101                $projects=NULL,$users=NULL,$billed_customers=NULL,$customers=NULL,
102                $worktypes=NULL,$statefilters=NULL,$outside=NULL,$user_id=NULL)
103 {
104
105     $_SESSION['yid']           = $company_id;
106     $_SESSION['sdate']        = $sdate;
107     $_SESSION['edate']        = $edate;
108     $_SESSION['APEid']        = $apeid;
109     $_SESSION['outerSearch']   = $outerSearch;
110     $_SESSION['wid']          = $wid;
111     $_SESSION['projectss']    = $projects;
112     $_SESSION['stdi_users']    = $users;
113     $_SESSION['billed_customers'] = $billed_customers;
114     $_SESSION['customers']    = $customers;
115     $_SESSION['worktypes']    = $worktypes;
116     $_SESSION['filters']      = $statefilters;
117     $_SESSION['outside']      = $outside;
118     $_SESSION['user_id']     = $user_id;
119

```

Kuva 8. SetSQL-funktio.

```

9     $base = new base_core();
10     $base->setSQL(
11         $_SESSION['yid'],
12         $_SESSION['sdate'],
13         $_SESSION['edate'],
14         $_SESSION['APEid'],
15         $_SESSION['outerSearch'],
16         $_SESSION['wid'],
17         $_SESSION['projectss'],
18         $_SESSION['stdi_users'],
19         $_SESSION['billed_customers'],
20         $_SESSION['customers'],
21         $_SESSION['worktypes'],
22         $_SESSION['filters'],
23         $_SESSION['outside'],
24         $_SESSION['user_id']
25     );

```

Kuva 9. \$\_Session-muuttujan käyttö tulostuksessa.

## 5.10 Lomakkeet

### Toimintaperiaatteen esittely

BASE-toiminnanohjausjärjestelmässä toiminta perustuu käyttäjän syötteen käsittelyyn. Syötteet voivat olla hakuja järjestelmään erilaisilla parametreilla tai kokonaisia lomakkeita. Lomakkeet voivat olla erilaisia lomake-elementtejä sisältäviä. Perusteeltaan lomakkeiden sisältöä tarkastellaan tärkeimpien elementtien osalta Javascriptillä, ja mikäli tuo tarkastus menee läpi, lähetetään tiedot Ajax-pyyntöllä PHP-sivulle käsittelyä varten. Koska Ajax-pyyntö on asynkroninen, ei käyttäjän selain ole lukittuna lomakkeen lähetyksen jälkeen, vaan Ajax-pyyntö palautuessa Javascriptissä käsitellään PHP-sivulta palautettu tieto ja se lisätään HTML-sivun DOM (Document Object Model) elementtipuuhun. Käyttäjälle ei näy uutta sivunlatausta laisinkaan. Kaikkien mahdollisten sivujen kanssa ei ole tarvinnut käyttää näin ”monimutkaista” järjestelyä, vaan lomakkeen tiedot on voitu vain lähettää samalle sivulle uudestaan käyttäen POST-parametreja. Kuvassa 10 on esitetty yksinkertainen tapa lähettää Ajax-pyyntö käyttäen Mootools-kirjastoa.

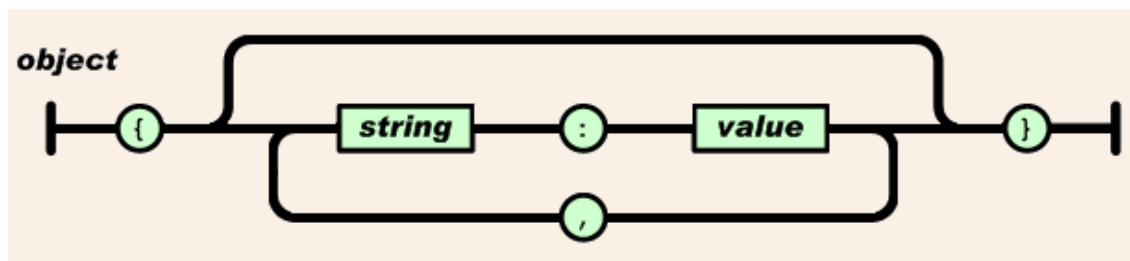
```
231     var customer_id = $('project_customer_id').value;
232     url="ldapAjax/stdBaseAjax.php?action=GetProjectsForCustomer";
233     var req = new Request({url: url, method: 'post', onSuccess: project_id_changed_response });
234     req.send("customer_id="+customer_id);
```

Kuva 10. Ajax-pyyntö Mootools-kirjastolla.

### JSON-tiedosiirtomuoto

Ajax-pyyntöjen yhteydessä usein on tarpeellista käyttää JSON (JavaScript Object Notation) -tiedonsiirtomuotoa. Kaikessa yksinkertaisuudessaan sen avulla saadaan järkevästi erilaista dataa helposti siirrettyä sivulle asti riippumatta enkoodauksesta tai muista merkistöistä riippuvista seikoista. JSON rakennetaan taulukoksi PHP:n avulla, jolloin esimerkiksi tietokannasta haetut kymmenet tai sadat rivit saadaan siististi pakettiin ja voidaan palauttaa yksinkertaisena JSON-oliona aktiivisen HTML-sivun Javascriptille käsiteltäväksi. HTML-sivulla JSON-olio dekodataan, ja sitä voidaan käyttää oliomaisesti ja kaikkiin taulukon jäseniin voidaan viitata niiden avain-arvoparien avainten nimillä.

Kuvassa 11 on esitetty miten JSON-olio rakentuu ja kuvassa 12 on yksinkertainen esimerkki miltä JSON-olio voisi näyttää. [12.]



Kuva 11. JSON-olion rakentuminen.

```

7  [ {
8    "0":"148","project_id":"148",
9    "1":"Oletus","project_name":"Oletus",
10   "2":null,"project_raport":null,
11   "3":"56","project_customer_id":"56",
12   "4":"0","project_billed_customer_id":"0",
13   "5":"59","project_owner_stdi_user_id":"59",
14   "6":"0","project_status_id":"0",
15   "7":null,"project_worktype_id":null,
16   "8":"2010-11-03 11:58:00",
17   "project_startdate":"2010-11-03 11:58:00"
18 } ]

```

Kuva 12. JSON-esimerkki.

## DOM-puun muokkaus

Ajax-pyyntöön palautuessa kutsutaan Javascriptin palautteen käsittelyfunktiota, jolle välitetään parametrina saadut tiedot. Kyseisessä funktiossa puretaan tarpeen mukaan JSON-olio tai käsitellään muuten saadut tiedot ja lisätään tiedot HTML-sivun DOM-puuhun. Mootols-kirjastossa on lukuisia erilaisia menetelmiä, mutta tässä projektissa useimmin käytetty funktio on Inject, jolla jokin HTML-elementti voidaan lisätä sivun olemassa olevaan DOM-puuhun tiettyyn paikkaan. Elementti ensin rakennetaan javascriptin puolella ja sitten lisätään oikeaan kohtaan DOM-puuta. DOM-puuta muokattaessa, selain päivittää välittömästi näkymän asiakkaalle, jolloin ei ylimääräistä sivunlatausta tapahdu. Ponnahdusikkunoiden luonti on tällä menetelmällä varsin yksinkertaista ja tehokasta toteuttaa, kuten kuvassa 13 on esitetty.

```

195 function laskutetutPopUp_response(text,xml)
196 {
197     elem = new Element('div');
198
199     elem.setStyles({
200         border:'1px solid',
201         position: 'absolute',
202         left: (mouseX + 15) + 'px',
203         top: (mouseY + 15) + 'px',
204         'background-color' : 'white',
205         'z-index': 11
206     });
207
208     elem.innerHTML = text;
209
210     elem.inject($('lisays'));
211 }

```

Kuva 13. Inject-funktion käyttö.

## 5.11 Tulostustoiminnot

Määrittelyssä päädyttiin PDF-tiedostomuotoon. PDF:n luontiin käytetyn tekniikan valinta ei ollut yksinkertaista eikä testaaminen ollut maailman nopeinta hommaa. Ensimmäinen vilkaisu internetin valikoimaan osoitti varteenotettaviksi vaihtoehdoiksi DOMPDF-sekä HTML2PDF –kirjastot [13,14]. Kummatkin ovat ilmaisia ja toimivat GNU LGPL-lisenssin alla [15], mikä käytännössä tarkoittaa sitä, että lähdekoodia, jossa käytetään LGPL-lisenssoituja komponentteja, ei tarvitse julkaista.

Ensimmäiseksi testattavaksi osui DOMPDF-kirjasto, joka lopulta osoittautui hankalaksi. DOMPDF vaikutti alkuun varsin lupaavalta. Kirjaston käyttö oli helppoa ja tulosteet yksinkertaisten esimerkkien kanssa onnistuivat hyvin. Työlistauksiin vaihdettaessa jouduttiin HTML-koodia siivoamaan monen monituista kertaa, eivätkä tulosteet olleet halutunlaisia. Yleisin tapaus tuotti tulokseksi yhden A4-kokoisen sivun, jossa kaikki taulukot olivat kohdistuneet yhteen kohtaan paperia päällekkäin. Dompdf:n dokumentaatiota selvitettyä huomattiin siitä puuttuvan tuki työlistaustaulukon tärkeimmille CSS-kuvauksille, joten jouduttiin työstämään yksinkertaisempi taulukko tulostusta varten.

BaseCore-luokkaan tehtiin tulostusta varten oma funktiota, joka muotoilee taulukon käyttäen toista CSS-tiedostoa. Tälläkään menetelmällä ei päästy edes kohtuulliseen tulokseen, joten oli aika kokeilla HTML2-PDF-kirjastoa. Olin joskus aikasemmin päässyt kokeilemaan HTML2-PDF-kirjastoa, joten se vaikutti tutummalta heti alkuunsa. Ensimmäinen testi osoitti kirjaston sopivuuden projektin käyttöön, joskin vähän vielä oli

hommassa hiottavaa sivunvaihtojen osalta. HTML2-PDF-kirjasto tarjoaa monipuoliset toiminnot mitä erilaisimpiin toimintoihin:

- automaattinen sivunvaihto tietyn HTML-elementin jälkeen
- ylä- ja -alatunnisteet
- viivakoodituki
- PDF:n näyttö selaimessa tai tallennus levyille
- sivun pinta-alan käyttö
- marginaalit.

Kuvassa 14 on esitettyä normaali työlistausta, jossa on eriteltynä kaksi erillistä työtä. Ensimmäisessä työssä ei ole kuin yksi raportti, mutta alemmassa on syötetty jälkeensä yksi lisärivi. Työlistausta ei kyseisessä muodossa onnistuttu kääntämään PDF:ksi, vaan sitä varten jouduttiin tekemään kokonaan oma funktio BaseCore-luokkaan.

Työn nimi	Inssiduuni	Valmis ID=1096	Tilaaja Mikko Töykkälä	Tehty 2.5 16:28	Lisätty 2.5 16:40	Muokattu	Poista: X	
Projekti	Oletus							
Otsikko	Viimeistelyä							
Yritys	L. yritys	Tekijä	Työn laatu	Selvitys	tuntia	L. tuntia	Lisärivi	Laskuta
STD Systems Oy	STD Systems Oy	Tomi Nousiainen	Ajatustyö	- kuvankaappaukset - yhteenveveto - korjauksia	3	3	Tee lisärivi Muokkaa Kopioi	<input type="checkbox"/>
<b>Tehtävän tunnint yhteensä</b>					<b>3</b>	<b>3</b>	Merkkaa keskeneräiseksi	

Työn nimi	Intra bugikorjaukset toukokuu 2011	Valmis ID=1092	Tilaaja Mikko Töykkälä	Tehty 2.5 16:26	Lisätty 2.5 16:28	Muokattu	Poista: X			
Projekti	Oletus									
Otsikko	Paikannus									
Yritys	L. yritys	Tekijä	Työn laatu	Selvitys	tuntia	L. tuntia	Lisärivi	Laskuta		
STD Systems Oy	STD Systems Oy	Tomi Nousiainen	Ohjelmointi	- korjattu ajoneuvojen tilatietoja - suuntimat tarkastettu	3	3	Tee lisärivi Muokkaa Kopioi	<input type="checkbox"/>		
Lisärivi					ID=1094	Tilaaja Mikko Töykkälä	Tehty 2.5 16:28	Lisätty 2.5 16:28	Muokattu	Poista: X
Otsikko	palkkaraportti									
Yritys	L. yritys	Tekijä	Työn laatu	Selvitys	tuntia	L. tuntia	Lisärivi	Laskuta		
STD Systems Oy	STD Systems Oy	Tomi Nousiainen	Ohjelmointi	- korjattu palkkaraportista pyöristyksen summavirheet	1	1	Tee lisärivi Muokkaa Kopioi	<input type="checkbox"/>		
<b>Tehtävän tunnint yhteensä</b>					<b>4</b>	<b>4</b>	Merkkaa keskeneräiseksi			

Hakutuloksen yhteensä	7	7	
tuntia	7	7	
Ajatustyö	3	3	tuntia
Ohjelmointi	4	4	tuntia

Laskuta valitut tuloste - Pysty tuloste - Vaaka Valitse kaikki

Kuva 14. Työlistausta.

## 5.12 Pilottivaihe

Pilottivaiheella tarkoitetaan sitä vaihetta, jossa tuotetta kokeillaan ensimmäisen kerran mahdollisesti jopa loppukäyttäjän kanssa. Pilottivaiheen aikana tuotetta testaavaa käyttäjää tai käyttäjäkuntaa kuunnellaan aktiivisesti ja korjataan virheitä, joita ei sisäisessä testauksessa olla huomattu. Pilottivaiheen kesto voidaan sopia yhdessä pilottiin osallistujien kesken tai se voi olla ennalta määrätty. Pilottivaiheen jälkeen tarvittavat muutokset toteutetaan, jonka jälkeen seuraa yleensä käyttöönottovaihe tai vaihtoehtoisesti tuotteen julkaisu.

Yritystili valitsi asiakkaistaan yhden vapaaehtoisin, jolle luotiin järjestelmään tunnukset sekä annettiin pohjustava perehdytys järjestelmään. Pilottivaiheen aikana huomattiin useita konkreettisia virheitä, jotka olisivat mahdollisesti aiheuttaneet sekaannusta käytön tuotantovaiheessa. Ensimmäisen laskutuksen jälkeen alettiin jo konkreettisesti huomata järjestelmän tehokkuus laskutukseen käytettävän ajan vähentyessä sekä asiakkaan tietoisuuden lisäämisessä siitä, mitä milloinkin laskutetaan. Pilottiasiakkaalta saatiin välillä palautetta, että jotkin haut eivät välttämättä toimineet aivan oikein suodatuksineen, jolloin näihin osa-alueisiin osattiin keskittyä paremmin. Pilotin loppuvaiheessa järjestelmään oli tehty esimerkiksi seuraavanlaisia muutoksia:

- Töiden tulostus oli muutettu PDF-pohjaiseksi.
- Töiden listauksia oli muokattu selkeämmiksi.
- Työlistaukseen oli lisätty töiden muokkaushistoria-toiminnallisuus.
- Töiden hakutoimintoja oli paranneltu.

## Käyttöönotto

Ennen käyttöönottovaihetta Yritystilissä ruvettiin lisäämään asiakkaita järjestelmään, mikä loi uusia haasteita järjestelmän pääkäyttäjälle. Asiakkaiden lisäys oli sinänsä helppoa, mutta uuden käyttäjätunnuksen luominen asiakkaalle ja sen liittäminen asiakkaaseen vaati liikkumista eri sivujen välillä ja saattoi täysin unohtua kiireessä. Lisäystä varten asiakkaiden lisäyssivulle tehtiin toiminto, jossa tunnuksen voi tehdä asiakkaalle jo asiakasta järjestelmään luotaessa. Kaikille asiakkaille luotiin järjestelmään yritysniemet sekä henkilökohtaiset käyttäjätunnukset. Yritystili myös lähetti asiakkailleen info-kirjeen järjestelmän käyttöönotosta.

## 6 Yhteenveto

Työn tarkoituksena oli määrittellä ja toteuttaa töiden hallinta- ja raportointiohjelma. Vaatimukset määriteltiin ja niistä keskusteltiin yhdessä Yritystilin kanssa. Tekemiseen oli helppo päästä käsiksi erinomainen määrittelyn avulla. LDAP:n ja MySQL:n ongelmallisen vaikean yhteen hiileen puhaltamisen unohtamisen jälkeen tietokanta saatiin toimintakuntoon ja varsinaisia toimintoja päästiin kehittämään. LDAP:n käyttö yhdessä MySQL:n kanssa vaatisi enemmän aikaa, kuin projekti antoi myöten. Työn yksi suurimmista helpotuksista oli Mootools-kirjasto, jonka avulla Ajax-pyyntöt ja HTML-sivujen DOM-puiden muokkaukset onnistuivat pienellä vaivalla.

Työstä ilmeni myös selainten erilainen tulkinta HTML-sivujen sisällöstä, josta johtuen tulostusta ei kannata jättää selaimelle yksinään. Internetistä löytyy maksullisia sekä vapaaseen lähdekoodiin perustuvia PHP:lle tarkoitettuja PDF-kirjastoja, joilla tulosteet kannattaa ehdottomasti rakentaa.

Käyttäjältä töiden raportointiin käytetty aika voi ehkä hieman pidentyä Exceliin naputtamiseen verrattuna, mutta järjestelmää oppii arvostamaan, kun huomaa, että omat työt pysyvät tallessa ja niitä on helppo hakea ja tehdä yhteenvetoa. Töiden laskuttajalta vähentyy laskuttamiseen kuluva aika, sekä luottamus ja tiedonkulku asiakkaan välillä paranee huomattavasti. Järjestelmä tarjoaa myös tiedot palkanlaskentaan, mikäli palkkaus on urakkapohjaista. Kaikki määrittelyssä esitetyt vaatimukset sekä kehityksen aikana esiin tulleet vaatimukset toteutettiin vapaata hakua lukuun ottamatta. Pilottivaiheen jälkeen BASE otettiin onnistuneesti käyttöön, ja projektin voi sanoa onnistuneen. Ainoa tavoite mikä on vielä tavoittamatta, on saada järjestelmä kaikkien asiakkaiden tietoisuuteen, niin että he myös ymmärtävät järjestelmän tarkoituksen ja oppivat hyödyntämään sitä kokonaisvaltaisesti. Tulevaisuudessa BASEen liitetään esimerkiksi rajapinta laskutusohjelmiin, jolloin laskut ja työtunnit (palkat) voidaan ajattaa automaattisesti eteenpäin jatkokäsittelyä varten.

## Lähteet

- 1 UTF-8. 2011. Verkkodokumentti. Wikipedia. <<http://en.wikipedia.org/wiki/UTF-8>>. Luettu 17.4.2011.
- 2 What is PHP?.2011. Verkkodokumentti. Php.net. <<http://fi2.php.net/manual/en/intro-whatism.php>>. Luettu 17.4.2011.
- 3 MySQL. 2011. Verkkodokumentti. Wikipedia. <<http://fi.wikipedia.org/wiki/MySQL>>. Luettu 17.4.2011
- 4 LDAP. 2011. Verkkodokumentti. Wikipedia. <<http://fi.wikipedia.org/wiki/LDAP>>. Luettu 17.4.2011.
- 5 19 tuuman räkki. 2011. Verkkodokumentti. Wikipedia. <<http://fi.wikipedia.org/wiki/U-yksikk%C3%B6>>. Luettu 25.4.2011.
- 6 Harri Laine. 2011. Web-sovelluksen rakenteesta. Verkkodokumentti. <[http://www.cs.helsinki.fi/u/laine/tikas/material/web\\_sovellus.html](http://www.cs.helsinki.fi/u/laine/tikas/material/web_sovellus.html)> . Luettu 25.4.2011.
- 7 CSS. 2011. Verkkodokumentti. Wikipedia. <<http://fi.wikipedia.org/wiki/Css>>. Luettu 25.4.2011.
- 8 Oracle. Verkkodokumentti. Oracle Website. <<http://www.oracle.com/us/products/database/index.html>>. Luettu 25.4.2011.
- 9 Harri Inkinen. 2011. Skriptikielet ja dynaamiset WWW-sivustot. Verkkodokumentti. <[http://matwww.ee.tut.fi/hmopetus/hypmed02/arc/hyper\\_skripti\\_hi\\_201102.pdf](http://matwww.ee.tut.fi/hmopetus/hypmed02/arc/hyper_skripti_hi_201102.pdf)>. Luettu 25.4.2011.
- 10 W3C. 2011. Verkkodokumentti. W3C. <<http://www.w3.org/standards/webdesign/htmlcss>>. Luettu 25.4.2011.
- 11 Apache. 2011. Verkkodokumentti. Apache Foundation. <[http://httpd.apache.org/ABOUT\\_APACHE.html](http://httpd.apache.org/ABOUT_APACHE.html)>. Luettu 25.4.2011.
- 12 JSON. 2011. Verkkodokumentti. JSON.org. <<http://www.json.org/>>. Luettu 30.4.2011.
- 13 HTML to PDF converter. 2011. Verkkodokumentti. dompdf. <<http://code.google.com/p/dompdf/>>. Luettu 1.5.2011.
- 14 html2pdf. 2011. Verkkodokumentti. html2pdf. <<http://html2pdf.fr/en/default>>. Luettu 1.5.2011.
- 15 GNU LGPL. 2011. Verkkodokumentti. Wikipedia. <[http://fi.wikipedia.org/wiki/GNU\\_LGPL](http://fi.wikipedia.org/wiki/GNU_LGPL)>. Luettu 1.5.2011.
- 16 Joomla. 2011. Verkkodokumentti. joomla.org. <<http://www.joomla.org/>>. Luettu 1.4.2011.

- 17 ZEND. 2011. Verkkodokumentti. Zend Technologies Inc. <<http://www.zend.com/en/>>. Luettu 1.4.2011.
- 18 Sillaj. 2011. Verkkodokumentti. Sourceforge. <<http://sillaj.sourceforge.net/>>. Luettu 4.5.2011.
- 19 Korpela Jukka K., Linjama Tero. 2004. XHTML Käsikirja. Docendo Finland Oy.
- 20 Ajax(Ohjelmointi). 2011. Verkkodokumentti. Wikipedia. <[http://fi.wikipedia.org/wiki/Ajax\\_%28ohjelmointi%29](http://fi.wikipedia.org/wiki/Ajax_%28ohjelmointi%29)>. Luettu 8.5.2011.