

# Tuntikirjaus-ohjelmisto

Tietokannan ja web-käyttöliittymän suunnittelu ja toteutus sekä mobiilisovellus

**Laura Rosenberg**

Opinnäytetyö

---



Koulutusala Tekniikan ja liikenteen ala	
Koulutusohjelma Hyvinvointiteknologian koulutusohjelma	
Työn tekijä(t) Laura Rosenberg	
Työn nimi Tuntikirjausohjelmiston tietokannan ja web-käyttöliittymän suunnittelu ja toteutus sekä mobiilisovellus	
Päiväys 29.5.2012	Sivumäärä/Liitteet 62
Ohjaaja(t) lehtori Jussi Koistinen, lehtori Sami Lahti	
Toimeksiantaja/Yhteistyökumppani(t) SoftRain Blobs Oy	
Tiivistelmä  <p>Tämän opinnäytetyön tavoitteena oli tuottaa SoftRain Blobs Oy:lle web-sovellus tuntien kirjaamiseen, jota käytetään PC:n selaimella. Työssä tutkittiin markkinoilla olevia tuntikirjausohjelmistoja ja niiden soveltuvuutta yrityksen tarpeisiin.</p> <p>Opinnäytetyössä keskityttiin tuntikirjausohjelmiston tietokantojen suunnitteluun ja web-käyttöliittymän suunnitteluun. Ohjelmiston toteutukseen käytetty alusta oli Spring, johon lisättiin Spring Roo + GWT. Lisäksi suunniteltiin ja toteutettiin tuntikirjaukseen mobiilisovellus Windows Phonelle.</p> <p>Opinnäytetyön tuloksena yritys sai tuntikirjaussovelluksen ja kokemusta web-sovellusten kehittämisestä.</p>	
Avainsanat Spring, roo, Gwt, windows phone, tuntikirjaus	

**SAVONIA UNIVERSITY OF APPLIED SCIENCES  
THESIS**

**Abstract**

Field of Study Technology, Communication and Transport			
Degree Programme Degree Programme in Welfare Engineering			
Author(s) Laura Rosenberg			
Title of Thesis Database and Web Interface Design in Hour Recording Software and its Mobile Application			
Date	29 May 2012	Pages/Appendices	62
Supervisor(s) Mr Jussi Koistinen, Lecturer; Mr Sami Lahti, Lecturer			
Client Organisation/Partners SoftRain Blobs Ltd			
Abstract  <p>The purpose of this thesis was to produce web application software for SoftRain Blobs Ltd to be used for recording hours. This thesis also examined the hour recording softwares available on the market and their suitability for the company's needs.</p> <p>The focus in this thesis was on the database design and the web-interface design. The implementation was made by using Spring Framework with Spring Roo and GWT. In addition, a mobile application for Windows Phone was made to be part of the web application software for hour recording.</p> <p>With the result of the thesis the company got the web application software for hour recording and experience in web application programming in order to widen their knowledge.</p>			
Keywords Spring, Roo, Gwt, Windows Phone, web application			

## ALKUSANAT

Haluan kiittää SoftRain Blobs Oy:tä opinnäytetyön aiheesta. Kiitokset työkaverilleni Kimmo Toivaselle vastaamisesta kaikkiin kysymyksiin, joita opinnäytetyön aikana esitin. Kiitokset kuuluvat myös opettaja Jussi Koistiselle, jonka ansiosta tämä työ valmistui ajoissa. Lisäksi haluan kiittää perhettäni siitä, että olette jaksaneet tukea minua tämän opinnäytetyöprosessin aikana.

Varkaudessa 29.5.2012

Laura Rosenberg

## SISÄLTÖ

1 JOHDANTO.....	8
2 TUNTIKIRJAUSOHJELMISTOTUTKIMUS.....	9
2.4 Esimerkkejä markkinoilla olevista järjestelmistä.....	10
2.4.1 Value Frame PSA Pro.....	10
2.4.2 Visma Avendo Aika.....	10
2.4.3 SimoApp.....	11
2.4.4 Tuntikirja.fi.....	11
2.4.5 Ebsolut MOTTI.....	11
2.4.6 Mobilog Mukana™.....	11
2.4.7 Älypuheliin ladattavat tuntikirjaussovellukset.....	12
2.5 Yhteenveto tutkimuksesta.....	12
2 TUNTIKIRJAUSOHJELMISTON SUUNNITTELU.....	14
2.1 Suunnittelun vaatimukset ja määrittely.....	14
2.2 Web-käyttöliittymän suunnittelu tuntikirjausohjelmistoon.....	15
2.2.1 Palvelun avaus ja kirjautuminen.....	16
2.2.2 Pääsivu.....	17
2.2.3 Käyttäjän lisäys ja poisto, käyttöoikeuden vaihtaminen.....	18
2.2.4 Salasanan vaihtaminen.....	20
2.2.5 Projektien lisäys, muutos ja poisto.....	21
2.2.6 Tehtävien lisäys, muutos ja poisto.....	22
2.2.7 Töiden lisäys, muutos ja poisto.....	23
2.2.8 Työntekijöiden liittäminen työtehtäviin.....	24
2.3 Tietokantasuunnittelu tuntikirjausohjelmistoon.....	25
3 OMAN TUNTIKIRJAUSOVELLUKSEN TOTEUTUS.....	32
3.1 Alustan valinta.....	32
3.2 Ympäristön pystytys.....	34
3.3 GWT web -sovelluksen rakenne ja toiminnallisuus.....	35
3.3.1 Model.....	35
3.3.2 View.....	37
3.3.3 Presenter.....	38
3.5 Spring Security .....	39
3.6 Näyttöjen rakenne ja muodostuminen.....	41
3.6.1 SpringRoön avulla toteutettu perusnäky.....	41
3.6.2 Oman näkymän luonti.....	42

3.7 Toteutetut ominaisuudet.....	44
3.7.1 Kirjautuminen.....	44
3.7.2 Päänäyttö.....	45
3.7.3 Työntekijän toiminnallisuus.....	46
3.7.4 Esimiehen tai projektipäällikön toiminnallisuus.....	48
3.7.5 Ylläpitäjän toiminnallisuus.....	50
3.7.6 Lisäominaisuudet.....	52
4.2 Sovellusalueen valinta.....	53
4.1 Suunnittelun vaatimukset ja määrittely.....	53
4.4 Toteutus.....	55
4.4.1 Kirjautuminen.....	55
4.4.2 Päänäyttö.....	57
4.4.3 Tuntikirjaus.....	58
5 YHTEENVETO.....	61
LÄHTEET.....	63

## 1 JOHDANTO

SoftRain Blobs Oy on Varkaudessa vuonna 2007 perustettu yritys, joka työllistää seitsemän ohjelmistosuunnittelun ammattilaista. Yritys keskittyy erityisesti mobiili-, sulautettujen ja selainpohjaisten ohjelmistojen suunnitteluun. Vuoden 2011 alussa SoftRain aloitti toiminnan omana itsenäisenä yksikkönään irtauduttuaan Protacon-konsernista, joten toimintatapoja ja ohjelmistoja jouduttiin miettimään uudelleen. Yksi näistä ohjelmistoista liittyi tuntikirjaukseen ja sen toteuttamiseen. Tuntikirjaussovelluksen avulla yritys haluaa laajentaa toimintaansa web-sovellusten suunnitteluun ja saada kokemusta uusista mobiilialustoista.

Tuntikirjausohjelmiston suunnittelu ja toteutus jaetaan kahden yrityksen työntekijän (Rosenberg, Toivanen) kesken siten, että Toivanen keskittyy opinnäytetyössään sovellusalan valintaan ja toteutukseen Spring Rooilla sekä tekee mobiilisovelluksen Androidille.

Tähän työhön sisällytetään käyttöliittymän ja tietokannan suunnittelu, tuntikirjaussovelluksen toteutus Spring Roo + GWT -ympäristössä sekä mobiilisovellus Windows Phonelle. Lisäksi tutkitaan markkinoilla olevia tuntikirjausohjelmistoja ja niiden soveltuvuutta yrityksen tarpeisiin. Tutkimuksen perusteella voidaan määritellä ne tuntikirjaussovelluksen ominaisuudet, joita yrityksen käyttöön halutaan.



## 2 TUNTIKIRJAUSOHJELMISTOTUTKIMUS

Markkinoilla olevia tuntikirjausohjelmistoja tutkitaan internetistä saatavien tietojen ja omien kokemusten perusteella. Tässä työssä kartoitetaan valmiiden ohjelmien ominaisuuksia ja yritetään löytää yrityksen tarpeita vastaava ohjelma. Tutkittavat tuntikirjausohjelmistot voidaan jakaa karkeasti kahteen ryhmään, toiminnanohjausjärjestelmiin sekä tuntikirjausohjelmistoihin.

Toiminnanohjausjärjestelmä eli ERP (Enterprise Resource Planning) on määritelmä erilaisia toimintoja sisältävästä yrityksen tietojärjestelmästä. Tällaiset järjestelmät ovat laajoja ohjelmistokokonaisuuksia, joiden tarkoitus on kattaa suuri osa yrityksen tietojenkäsittelytarpeesta. (Wikipedia 2012.)

Järjestelmään voi sisältyä esimerkiksi palkanlaskentaan, varastonhallintaan, tuotannon hallintaan sekä töiden hallintaan liittyviä toimintoja. Järjestelmän osiot koostuvat erillisistä moduuleista, joita voidaan ostaa ja ottaa käyttöön vaiheittain. Toiminnanohjausjärjestelmä tuo mukanaan yritykselle muutakin kuin tietojärjestelmän, yleensä kysymys on myös yrityksen toimintatapojen ja liiketoimintaprosessien kokonaisvaltaisesta muutoksesta. Järjestelmän käyttöönotto vie yritykseltä paljon aikaa ja resursseja.

Markkinoilla on saatavana runsaasti erilaisia toiminnanohjausjärjestelmiä. Yleensä näitä järjestelmiä käyttävät suuret yritykset, mutta myös pienempien yritysten tarpeisiin kehitettyjä järjestelmiä on saatavilla. Luvussa 2.4.1 kuvattu ValueFrame on esimerkki toiminnanohjausjärjestelmästä.

Tuntikirjausohjelmistot sisältävät vain palkanlaskentaan liittyvät toiminnot. Ohjelma on tyypillisesti linkki työntekijöiden, palkanlaskennan ja yritysjohtoon välillä. Se voi sisältää kirjanpito- ja reskontraominaisuuksia. Luvussa 2.4.2 kuvattu Visma Avendo Aika on esimerkiksi tällainen tuntikirjausohjelmisto.

Teknisesti toiminnanohjausjärjestelmät ja tuntikirjausohjelmistot toteutetaan nykyään poikkeuksetta selainpohjaisina järjestelminä tai ohjelmistoina. Selainpohjaisena järjestelmänä toteutetut ohjelmistot toimivat joko yrityksen omalla palvelimella tai palve-

luntarjoajan palvelimella. Tällaisia palveluja kutsutaan SaaS (Software as a Service) -sovelluksiksi. Se tarkoittaa, että käyttäjän ei tarvitse hankkia sovellusta itselleen, vaan sitä voidaan käyttää internetselaimen välityksellä. Tällaisen palvelun käyttöön-otosta ei aiheudu yritykselle investointikustannuksia. Ohjelman käytöstä maksetaan sovittua maksua esimerkiksi kuukausittain ja se voidaan hinnoitella esimerkiksi käyttäjämäärän perusteella.

Liikkuvan työn ja älypuhelimien lisääntymisen myötä mobiilisovellusten merkitys on kasvanut entisestään. Toiminnanohjausjärjestelmät ja tuntikirjaussovellukset sisältävät usein rajapinnan tai mukautetut näkymät mobiiliohjelmistoihin. Mobiiliohjelmisto tässä dokumentissa tarkoittaa älypuhelimelle suunniteltua sovellusta. Mobiilisovellukset ovat yleensä suppeita sisältäen vain tarvittavat ominaisuudet.

## 2.4 Esimerkkejä markkinoilla olevista järjestelmistä

Seuraavassa on kerätty muutamia markkinoilla olevia ohjelmistoja, jotka saattaisivat sopia yrityksen käyttöön.

### 2.4.1 Value Frame PSA Pro

ValueFrame PSA Pro on toiminnanohjausjärjestelmä, joka soveltuu niin pienille kuin suurillekin yrityksille. Järjestelmä räätälöidään toimialoittain ja sen voi ottaa käyttöön pala kerrallaan, omien tarpeiden mukaan. ValueFrame sisältää toiminnanohjauksen lisäksi myös projektinhallinnan ja asiakkuudenhallinnan. (ValueFrame 2012.)

### 2.4.2 Visma Avendo Aika

Visma Avendo Aika on tuntikirjausohjelma pienyritysten käyttöön. Ohjelma auttaa seuraamaan tehtäviä, projekteja sekä sisäistä ajankäyttöä. Visma Avendo Aika sopii kaikille yrityksille, joilla on laskutettavia projekteja tai ajankäytön seurantaa. Ohjelmassa on asetukset myös kirjanpito- ja tilintarkastustoimistoille. Useat ohjelman osat ovat muokattavissa yrityksen omiin tarpeisiin sopiviksi. Ohjelmaa voidaan käyttää verkossa ja siinä voi olla useampi käyttäjä yhtä aikaa. (Visma Avendo Oy 2011.)

### 2.4.3 SimoApp

SimoApp on tuntikirjaussovellus, joka on suunniteltu yrityksen operatiivisten toimintojen hallintaan. Rekisteröinnin yhteydessä yritykselle luodaan tietojärjestelmä yhdelle tai useammalle pääkäyttäjälle. Pääkäyttäjä voi avata omilla tunnuksilla olevan käyttöliittymän yhdelle tai useammalle työntekijälle sekä avata ja sulkea projekteja, joihin töitä kirjataan. Työntekijät voivat tehdä kirjauksia niille projekteille, jotka pääkäyttäjä on heille avannut. Pääkäyttäjä hyväksyy kirjaukset, jolloin tieto hyväksymisestä välittyy myös työntekijälle. Tietojärjestelmästä voi generoida laskutustiedot, matka- ja työaikatiedot. SimoApp-järjestelmää pystyy käyttämään kaikkialla maailmassa ympäri vuorokauden. (Noyra Oy 2012.)

### 2.4.4 Tuntikirja.fi

Tuntikirja.fi mahdollistaa tuntien kirjaamisen järjestelmään ja palkanlaskennan näiden tuntien perusteella. Käyttäjä kirjautuu järjestelmään henkilökohtaisten tunnusten avulla. Ohjelmiston käyttöliittymä sisältää mobiilioptimoidut näkymät, joten tuntien kirjaus onnistuu myös iPhonella ja Symbian- tai Android-puhelimen selaimella. Palvelun avulla työnantaja voi seurata ja ylläpitää työntekijöidensä tuntikirjauksia ja palkanmaksua. Palvelulla on käyttäjäperusteinen laskutus, ja se toimii kaikkialla maailmassa ympäri vuorokauden. (Net Weight Oy 2012.)

### 2.4.5 Ebsolut MOTTI

Ebsolut MOTTI on matkapuhelinsovellus tuntikirjausten tekemiseen ajasta ja paikasta riippumatta. Sovelluksen toimintoihin kuuluu tuntien kirjaaminen ja niiden hyväksyminen. Sovellus integroituu tarvittaessa toiminnanohjausjärjestelmään, jolloin viestintä tapahtuu Webservice rajapinnan läpi XML- viestein. (Ebsolut Oy 2012.)

### 2.4.6 Mobilog Mukana™

Mobilog Mukana™ on tarkoitettu liikkuvaa työtä tekevien henkilöiden töiden ja työajan seuraamiseen. Mobilog integroituu yrityksen ydinjärjestelmiin, sillä matkapuhelimessa luodut tapahtumat ohjataan laskutus- ja palkanmaksujärjestelmiin. Tuntikirjaukset voi-

daan kohdistaa projekteille, asiakkaille, työkohteille tai kustannuspaikoille. (Logica 2012.)

#### 2.4.7 Älypuheliin ladattavat tuntikirjaussovellukset

Puhelimiin ladattavia sovelluksia katsottiin kolmen suosituimman markkinoilla olevan laitteen latauspaikoista, joita ovat Google Play Androidille, AppStore iPhonelle ja WindowsPhoneMarket Windows Phonelle. Tuntikirjaus- ja työajanseurantasovelluksia tutkittaville laitteille löytyi useita ja niitä kehitetään koko ajan lisää. Näistä löydetyistä sovelluksista osa oli maksullisia ja osa ilmaisia. Yhteenvetona voisi sanoa, että nämä tuntikirjaussovellukset ovat itsenäisesti puhelimesta toimivia sovelluksia. Suurin osa näistä sovelluksista toimii kellokortin tapaan, jossa nappia painamalla annetaan työn aloitus- ja lopetusaika. Sovellus laskee annettujen aikojen perusteella kuluneen työajan ja laskutus asiakkaalta voidaan suorittaa. Nämä sovellukset soveltuvat hyvin liikkuvaa työtä tekeville, esimerkiksi asentajille, kodinhoitajille, remonttimiehille ja sellaisille, joiden työ ei ole projektiluonteista.

#### 2.5 Yhteenveto tutkimuksesta

Tutkimuksen lopputuloksena saatiin paljon tietoa markkinoilla olevista tuntikirjausohjelmistoista. Yrityksen ollessa vielä osana Protacon-konsernia, oli käytössä toiminnanohjausjärjestelmä (ValueFrame), jossa tuntikirjaus on osana. Yleisesti toiminnanohjausjärjestelmät koettiin liian massiivisiksi nykyisen yrityksen tarpeisiin. Yrityksen käyttöön riittää tässä vaiheessa tuntikirjaustoiminnallisuus, sillä projektinhallinta ei ole vielä niin suurta (2-3 projektia kerrallaan). Älypuheliin ladattavat tuntikirjaussovellukset ovat sellaisenaan taas liian suppeita yrityksen käyttöön ja ne toimivat suurelta osin kellokorttimaisesti. Luvussa 2.4 esitellyt tuntikirjausjärjestelmät Visma Avendo Aika, SimoApp ja tuntikirja.fi olisivat soveltuneet yrityksen tarpeisiin, sillä ne sisältävät paljon sellaisia ominaisuuksia, joita tuntikirjaussovellukselta haetaan. Jos tarvetta päättöyön tekemiselle ei olisi ollut, olisimme hyvinkin saattaneet valita niiden joukosta yhden yrityksen käyttöön.

Yrityksellä on vuosien kokemus mobiilisovelluksista ja koska esimerkiksi Symbian-kehitys alkaa olla taakse jäänyttä aikaa, nähtiin mahdollisuus laajentaa omaa osaamista web -sovellusten maailmaan oman tuntikirjaussovelluksen avulla. Samalla

myös olemassa olevaa mobiilisovellus osaamista voisi integroida mukaan. Toisaalta toteuttamalla oma tuntikirjaussovellus, voidaan hyödyntää koulussa opittuja asioita ja näin tuoda uutta tietoa myös yrityksen muille työntekijöille.

Markkinoilla olevien ohjelmistojen perusteella lähdettiin suunnittelemaan ominaisuuksia omaan ohjelmistoon, jota sitten joskus voisi tarjota palveluna muillekin yrityksille. Toteutus omaksi tuotteeksi asti on kuitenkin pitkä prosessi, eikä tuotteistaminen ole ensisijainen tavoite, mutta sen mahdollisuus otettiin suunnittelussa huomioon. Tutkimuksen yhteydessä ei sellaista ohjelmaa löytynyt, jossa tuntikirjaussovellus olisi tehty selainpohjaisena ja siihen lisäominaisuutena olisi saatavana mobiililaitteeseen sopiva asennuspaketti tuntien kirjaamista varten. Lisäksi halusimme mahdollisuuden tallentaa tuntikirjaukset mobiililaitteen tietokantaan, jos internet yhteyttä ei ole käytössä tai sen käyttäminen on jossain päin maailmaa liian kallista. Mobiililaitteen selaimella tuntikirjauksia pystyi tekemään monenkin valmistajan sovelluksissa.

## 2 TUNTIKIRJAUSOHJELMISTON SUUNNITTELU

Uuden tuntikirjausohjelmiston suunnittelussa ja toteutuksessa oli tavoitteena paitsi saada opintoihin vaadittava päättötyö tehtyä, niin laajentaa osaamista web-ohjelmoinnin maailmassa ja opetella uusia työtapoja. Päättötyönä toteutetaan yrityksen omaan käyttöön soveltuva tuntikirjausohjelmisto web-sovelluksena, joka vähentää yrityksen sisäistä työtä laskutuksen ja palkkojen maksun eri vaiheissa. Lisäksi toteutetaan kaksi mobiilisovellusta, jotka liittyvät tuntikirjaussovellukseen niille suunnitellun rajapinnan kautta. Ohjelmiston suunnittelu ja toteutus jaetaan kahden yrityksen työntekijän (Rosenberg, Toivanen) kesken siten, että Toivanen työssään (Toivanen 2012.) keskittyy sovellusalustan valintaan ja toteutukseen Spring Rooilla sekä tekee mobiilisovelluksen Androidille. Tähän työhön sisällytetään käyttöliittymän ja tietokannan suunnittelu (luku 2), toteutus Spring Roo + GWT (luku 3) sekä mobiilisovellus Windows Phonelle (luku 4).

### 2.1 Suunnittelun vaatimukset ja määrittely

Tuntikirjausohjelmisto on web-palvelimessa toimiva sovellus, jossa on web-käyttöliittymä. Ohjelmisto tarjoaa rajapinnan mobiilisovelluksia varten. Sovellus toimii Java-palvelinalustalla (Tomcat, Jetty tms.) ja vaatii toimiakseen tietokannan (MySQL) sekä verkkoyhteyden. Ohjelmisto mahdollistaa työntekijöille tuntikirjauksen internet selaimella ja mobiililaitteella.

Ohjelmiston käyttöä rajoitetaan erilaisten käyttäjäroolien mukaan. Ensimmäisessä vaiheessa käyttäjäroolit määritellään työntekijälle, ylläpitäjälle ja esimiehelle tai projektipäällikölle sekä myöhemmin myös tilitoimistolle. Työntekijä kirjautuu palveluun omalla käyttäjätunnuksellaan ja voi kirjata työtunteja sellaisille projekteille, joihin hänet on merkitty työn tekijäksi. Ylläpitäjä lisää työntekijät ja muut käyttäjät järjestelmään. Projektin esimies tai projektipäällikkö asettaa työntekijän projektin tekijäksi. Tilitoimisto voi ottaa järjestelmästä raportin kunkin työntekijän palkkakauden tuntien määräst. Ensimmäisessä vaiheessa ohjelmistosta toteutetaan tuntikirjaus osuus ja myöhemmin siihen lisätään mahdollisuus myöskin kulu- ja matkalaskujen tekemiseen. Ensimmäisen vaiheen toteutus liittyy tähän päättötyöhön. Tuntikirjausohjelmisto suunnitellaan pienten ja keskisuurten yritysten (max. 250hlö) tarpeisiin sopivaksi, sillä

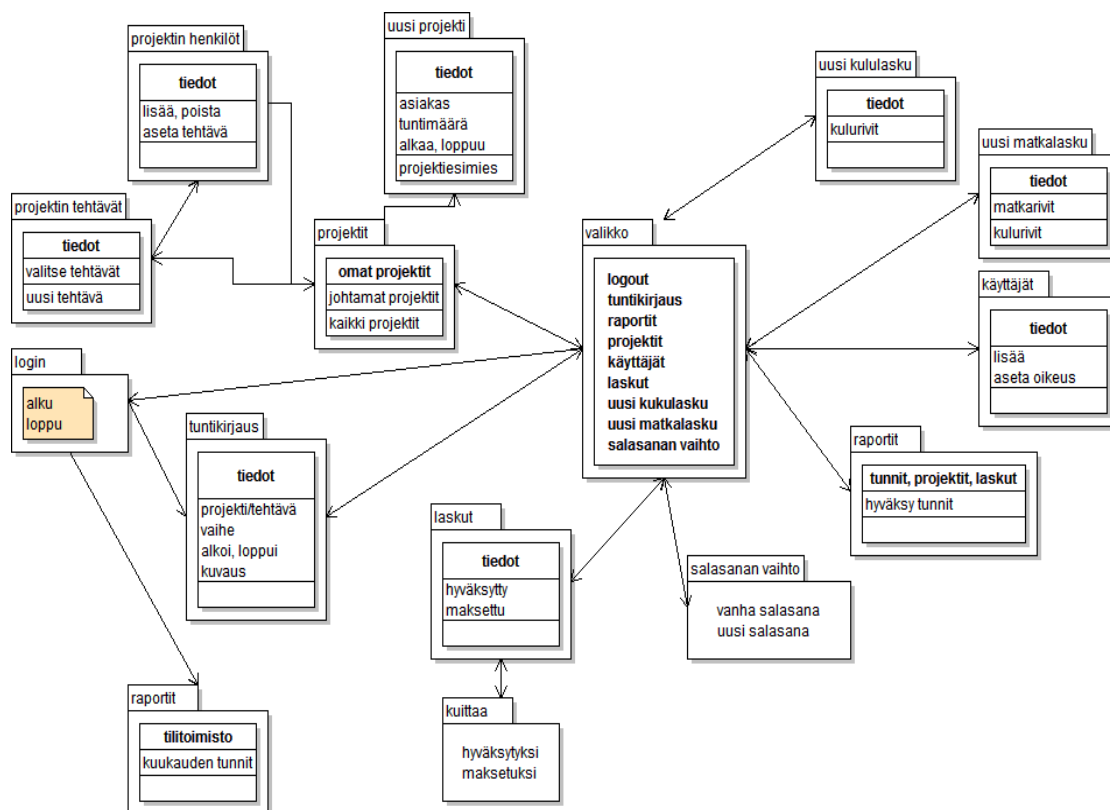
esimerkiksi serveri kapasiteetin määrä rajoittaa suunnittelua, eikä sitä ole yrityksellä suunnitelmissa kasvattaa suurille yrityksille sopivaksi.

## 2.2 Web-käyttöliittymän suunnittelu tuntikirjausohjelmistoon

Käyttöliittymän suunnittelussa on otettava huomioon, että ohjelmistoa käyttää työntekijöiden lisäksi esimiehet, projektipäälliköt ja tilitoimiston työntekijät. Näille kaikille käyttäjärooleille on luotava heidän käyttöoikeuksiaan vastaavat näkymät sovellukseen. Näiden lisäksi järjestelmällä on ylläpitäjä, joka luo käyttäjiä ja antaa käyttäjille käyttöoikeuksia.

Työntekijöiden on pystyttävä käyttämään järjestelmää ilman erillistä koulutusta. Esi- miesten ja ylläpitäjän on voitava suorittaa tavallisia tehtäviä lyhyen käyttöohjeen perusteella. Ylläpidon muita toimintoja, esimerkiksi järjestelmän käyttöönotto, varten tulee olla tarkempi ohje. Järjestelmää käytetään monta kertaa päivässä ja kukin käyttökerta vaatii useita sivunlatauksia ja tietokantahakuja, mikä on otettava käyttöliittymäsuunnittelussa huomioon.

Käyttöliittymä suunnittelussa keskitytään siihen mitä komponentteja tietyn asian toimintaan tarvitaan. Tämän kappaleen sisältämät näytönkuvat ovat suunnittelunaikaisia kuvia, joiden pohjalta toteutusta lähdetään tekemään, eivät kuvia varsinaisesta sovelluksesta. Alla olevassa kuvassa (kuvio 1) on suunnitelma erilaisista näytöistä, joita tuntikirjausohjelmiston toteuttamiseen on suunniteltu tarvittavan.



KUVIO 1. Suunnitelma erilaisista näytöistä tuntikirjausohjelmistossa

### 2.2.1 Palvelun avaus ja kirjautuminen

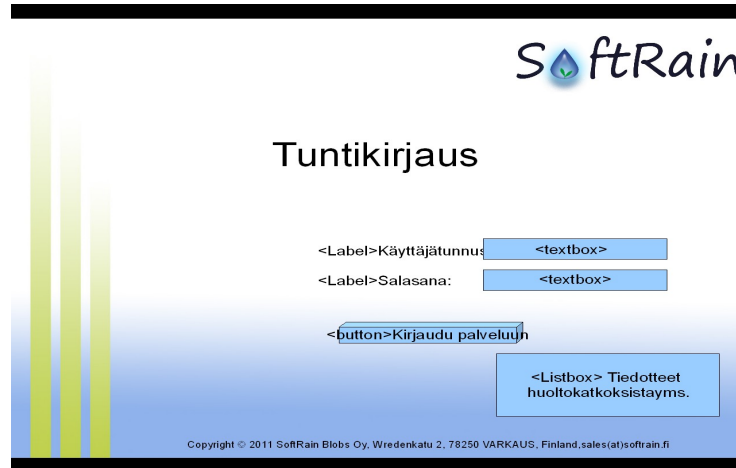
Kun käyttäjä avaa internetselaimeen palvelun URL (Uniform Resource Locator) -osoitteen, saadaan aloitusnäky, josta käyttäjä pystyy kirjautumaan palveluun. Aloitusnäkyssä (kuvio 2) näytetään taulukossa 1 kuvatut ominaisuudet.

TAULUKKO 1. Ominaisuudet palvelun avaukselle ja kirjautumiselle

Ominaisuus	Kuvaus
Palvelun tiedot	Sovelluksen nimi
Palveluntarjoajan tiedot	Yrityksen nimi ja logo
Käyttäjätunnus	Kenttä käyttäjätunnuksen syöttämiseen
Salasana	Kenttä salasanan syöttämiseen
Kirjautuminen	Painike palveluun kirjautumiseen
Muita tietoja	Ajankohtaista tietoa esim. huoltokatkoksista



Aloituspäätöksessä voi olla myös session tunnus, jota voidaan käyttää hyväksi myöhemmissä vaiheissa. Palvelua käytetään SSL-yhteydellä mikäli mahdollista. SSL (Secure Sockets Layer) on salausprotokolla, jolla voidaan suojata internetsovellusten tietoliikenne IP-verkkojen yli.



KUVIO 2. Suunnitelma aloitusnäköistä

## 2.2.2 Pääsivu

Pääsivulla, joka näytetään käyttäjälle heti sisäänkirjautumisen jälkeen, suoritetaan varsinainen tuntikirjaus. Pääsivulta voi olla linkkejä myös muihin toimintoihin, kuten myöhemmässä vaiheessa kulu- ja matkalaskujen kirjaukseen sekä raportointiin. Palvelu antaa selaimen näkymän (kuvio 3), jossa on taulukon 2 ominaisuudet.

TAULUKKO 2. Pääsivun ominaisuudet

Ominaisuus	Kuvaus
Kirjauksen tiedot	Kenttä, johon kirjaus kohdistuu
Aloitusaika	Kenttä aloitusajalle (päivämäärä, aika)
Lopetusaika	Kenttä lopetusajalle (päivämäärä, aika)
Ylityöt	Kenttä kestolle ylityöt
Työn kesto	Painike tai linkki keston laskemiseen aloitus- ja lopetusajan erotuksena
Selite	Kenttä työn kuvaukselle
Kuittaus	Kenttä kuittaukselle (projektin johto tai esimies tarkistaa kirjaukset ja kuittaa ne)
Tallennus	Painike tai linkki kirjauksen tallentamiseksi

SoftRain  
Vaihda salasana Kirjaudu ulos

**Pääsivu/tuntikirjaus** Tänään on tiistai 2. heinäkuuta 2002

Tuntikirjaus Matkalasku Kulukorvaus Raportit

<Label>Tervetuloa palveluun <käyttäjän nimi>

<Label>Päivämäärä: <textbox> Työ alkoi: <textbox> Työ päättyi: <textbox>

<Label>Projektin nimi/tehtävä: <listbox> Ruokatauko: <textbox>

<Label>Selite: <textbox>

ma	ti	ke	to	pe	la	su

Tallenna

Copyright © 2011 SoftRain Blobs Oy, Wredenkatu 2, 78250 VARKAUS, Finland.sales(at)softtrain.f

KUVIO 3. Suunnitelma tuntikirjauksen pääsivusta

### 2.2.3 Käyttäjän lisäys ja poisto, käyttöoikeuden vaihtaminen

Järjestelmän pääkäyttäjä lisää käyttäjiä palveluun. Selaimen näkymässä (kuvio 4) annetaan kullekin käyttäjälle taulukossa 3 määritellyt tiedot.

TAULUKKO 3. Ominaisuudet käyttäjän lisäykselle

Ominaisuus	Kuvaus
Käyttäjätunnus	Esimerkiksi sähköpostiosoite
Oletus salasana	Palvelu ehdottaa oletussalasanan
Käyttäjän oikeudet	USER, ADMIN, LOCALADMIN, voi asettaa/muuttaa myöhemminkin
Käyttäjän sukunimi	Kenttä sukunimelle
Käyttäjän etunimi	Kenttä etunimelle
Käyttäjän aktiivisuus	On töissä
Käyttäjän henkilötiedot	Kenttä henkilötiedoille, jos niitä tarvitaan

**SoftRain**  
Kirjaudu ulos

## Käyttäjän lisäys ja poisto

<Label>Käyttäjätunnus:

<Label>Oletussalasana:

<Label>Käyttäjän oikeudet:

<Label>Etunimi:

<Label>Sukunimi:

<Label>Työpuhelin numero:

<Label> Käyttäjän aktiivisuus:

[Lisää henkilötiedot<linkki>](#)

[Takaisin edelliselle sivulle](#)

Copyright © 2011 SoftRain Blobs Oy, Wredenkatu 2, 78250 VARKAUS, Finland. sales(at)softtrain.fi

KUVIO 4. Suunnitelma käyttäjän lisäyksestä ja poistosta

Jos käyttäjällä ei ole kirjauksia, käyttäjän voi poistaa. Muussa tapauksessa käyttäjää ei voi poistaa ennen kuin kirjatut poistetaan. Tällöin käyttäjätunnus voidaan määrittellä passiiviseksi. Passiivisella käyttäjätunnuksella ei voi kirjautua järjestelmään eikä passiivista käyttäjää voi liittää projekteihin tai tehtäviin. Passiivisen käyttäjän kirjatut pysyvät tietokannassa mahdollista myöhempää tarkastelua varten.

Mahdollisia käyttäjärooleja sovelluksessa ovat ylläpitäjä, projektipäällikkö tai esimies, työntekijä ja tilitoimisto. Taulukossa 4 on kuvattu näille rooleille mahdolliset käyttöoikeudet.

TAULUKKO 4. Käyttäjäroolit ja käyttöoikeudet

Käyttäjärooli	Käyttöoikeudet
Ylläpitäjä	Voi lisätä, muuttaa ja poistaa käyttäjiä.
	Voi lisätä, muuttaa ja poistaa projekteja.
	Voi lisätä, muuttaa ja poistaa tehtäviä.
	Voi lisätä, muuttaa ja poistaa vaiheita.
	Voi tehdä raporteja kaikesta.
	On myös työntekijä.
Projektipäällikkö/esimies	Voi lisätä, muuttaa ja poistaa omien projektien tekijöitä.
	Voi lisätä, muuttaa ja poistaa omien projektien tehtäviä.
	Voi hyväksyä (sulkea) tuntikirjauksia, kululaskuja ja

	matkalaskuja.
	Voi tehdä raportteja omista projekteistaan.
	On myös työntekijä.
Työntekijä	Voi lisätä, muuttaa ja poistaa omia avoimia tuntikirjauksia.
	Voi lisätä, muuttaa ja hyväksyä omia avoimia laskuja.
	Voi tehdä raportteja omista kirjauksistaan ja laskuistaan.
Tilitoimisto	Saa raportin kunkin työntekijän kuukausittaisesta työajasta ( normaali ja ylityö, liukuma? ).

#### 2.2.4 Salasanan vaihtaminen

Palveluun kirjautunut käyttäjä voi vaihtaa salasanansa. Palvelu antaa selaimen näkymän (kuvio 5), jossa on taulukossa 5 kuvatut ominaisuudet.

TAULUKKO 5. Ominaisuudet salasanan vaihtamiseksi

Ominaisuus	Kuvaus
Käyttäjän nimi	Kenttä käyttäjän nimelle
Entinen salasana	Kenttä entisen salasan syöttämiseen.
Uusi salasana	Kenttä uuden salasan syöttämiseen
Salasan vahvistaminen	Kenttä uuden salasan vahvistamiseen
Salasan vahvuus	Palkki tai sanallinen osoitus uuden salasan vahvuudesta.
Salasan vaihtaminen	Painike tai linkki salasan vaihtamiseksi.
Paluu oletusnäkyään	Painike tai linkki, josta palataan pääsivulle

SoftRain  
Kirjaudu ulos

## Salasanan vaihtaminen

Syötä salasanat <käyttäjän nimi>

Nykyinen salasana:

Uusi salasana:

Uuden salasanan vahvistus:

[Takaisin edelliselle sivulle](#)

Copyright © 2011 SoftRain Blobs Oy, Wredenkatu 2, 78250 VARKAUS, Finland, sales(at)softrain.f

KUVIO 5. Salasanan vaihtamisen suunnitelma

Jos uuden salasanan ja vahvistuksen yhdenmukaisuus voidaan tarkistaa selaimessa, tarkistetaan ensin, ovatko uusi salasana ja vahvistus samanlaiset. Jos salasanat ovat samanlaiset, lähetetään palveluun tiiviste entisestä ja uudesta salasanasta. Jos entisen salasanan tiiviste vastaa tallennettua tiivistettä ja uusi salasana on sama vahvistuksen kanssa, tallennetaan uuden salasanan tiiviste käyttäjän salasanaksi. Muussa tapauksessa annetaan selaimen sama näkymä sekä ilmoitus virheellisesti syötetyistä tiedoista. Jos uusi salasana ja vahvistus ovat erilaiset, annetaan käyttäjälle ilmoitus virheellisesti syötetyistä tiedoista eikä palvelimeen oteta yhteyttä.

Jos uuden salasanan ja vahvistuksen yhdenmukaisuutta ei voida tarkistaa selaimessa, lähetetään palveluun tiiviste kaikista salasanoina. Jos entisen salasanan tiiviste vastaa tallennettua tiivistettä ja uusi salasana on sama kuin vahvistus, tallennetaan uuden salasanan tiiviste käyttäjän salasanaksi. Muussa tapauksessa annetaan selaimen sama näkymä sekä ilmoitus virheellisesti syötetyistä tiedoista.

### 2.2.5 Projektien lisäys, muutos ja poisto

Projekti tarkoittaa tilattua työtä, jolla on asiakas tai yrityksen sisäistä työtä, jota ei las-  
kuteta eteenpäin. Jos palveluun on kirjautunut pääkäyttäjä tai muu käyttäjä, jolla on  
oikeus, lisätä, poistaa tai muuttaa projekteja, palvelu antaa selaimen näkymän (ku-  
vio 6), jossa on taulukossa 6 kuvatut ominaisuudet.

TAULUKKO 6. Ominaisuudet projektien lisäämiselle

Ominaisuus	Kuvaus
Projektin tunniste	Kenttä projektin tunnisteelle
Projektin nimi	Kenttä projektin nimelle tai kuvaukselle.
Asiakkaan nimi	Kenttä asiakkaan nimelle
Projektin esimiehen tiedot	Kenttä projektin esimiehen nimelle
Projektin kesto	Kenttä projektin kestolle (sovittu laskutusmäärä)
Projektin alku- ja loppuajat	Kentät projektin aloitus- ja lopetus ajalle
Projektin tallennus	Painike tai linkki projektin tietojen tallentamiseksi.
Paluu oletusnäkömään	Painike tai linkki, josta palataan pääsivulle.

KUVIO 6. Suunnitelma projektin lisäyksestä, muuttamisesta ja poistamisesta

### 2.2.6 Tehtävien lisäys, muutos ja poisto

Tehtävät voivat olla yleisluontoisia tehtävänimikkeitä, joita suoritetaan projekteissa, tai projektia varten räätälöityjä tehtäviä. Jos palveluun on kirjautunut pääkäyttäjä tai muu käyttäjä, jolla on oikeus lisätä, poistaa ja muuttaa projektin tehtäviä, palvelu antaa selaimeen näkymän (kuvio 7) seuraavilla taulukon 7 ominaisuuksilla.

TAULUKKO 7. Ominaisuudet tehtävien lisäykselle

Ominaisuus	Kuvaus
Tehtävän tunniste	Kenttä tehtävän tunnisteelle
Tehtävän nimi	Kenttä tehtävän nimelle
Tehtävän kuvaus	Kenttä tehtävän kuvaukselle
Tehtävän vaihe	Kenttä tehtävän vaiheille (esim. Pilkkulla erotettuna)



KUVIO 7. Suunnitelma tehtävien lisäyksestä, muuttamisesta ja poistamisesta

### 2.2.7 Töiden lisäys, muutos ja poisto

Työ liittyy yhteen projektin ja tehtävän. Jos palveluun on kirjautunut pääkäyttäjä tai muu käyttäjä, jolla on oikeus lisätä, poistaa ja muuttaa töitä, palvelu antaa selaimen näkymän (kuvio 8), jossa on taulukossa 8 kuvatut ominaisuudet.

TAULUKKO 8. Ominaisuudet töiden lisäykselle

Ominaisuus	Kuvaus
Tilaus	Kenttä tilaukselle, mihin työ liittyy.
Tehtävän nimi	Kenttä tehtävälle, mihin työ liittyy.
Työn kuvaus	Kenttä työn kuvaukselle

KUVIO 8. Suunnitelma töiden lisäyksestä, muuttamisesta ja poistosta

### 2.2.8 Työntekijöiden liittäminen työtehtäviin

Työ ja työntekijä liitetään yhteen työmääräimellä, jotta työ voidaan jakaa useamman työntekijän kesken. Jos palveluun on kirjautunut pääkäyttäjä tai muu käyttäjä, jolla on oikeus lisätä, poistaa ja muuttaa työmääräimiä, palvelu antaa selaimeen näkymän (kuvio 9), jossa on taulukon 9 mukaiset ominaisuudet.

TAULUKKO 9. Työntekijöiden liittäminen työtehtäviin

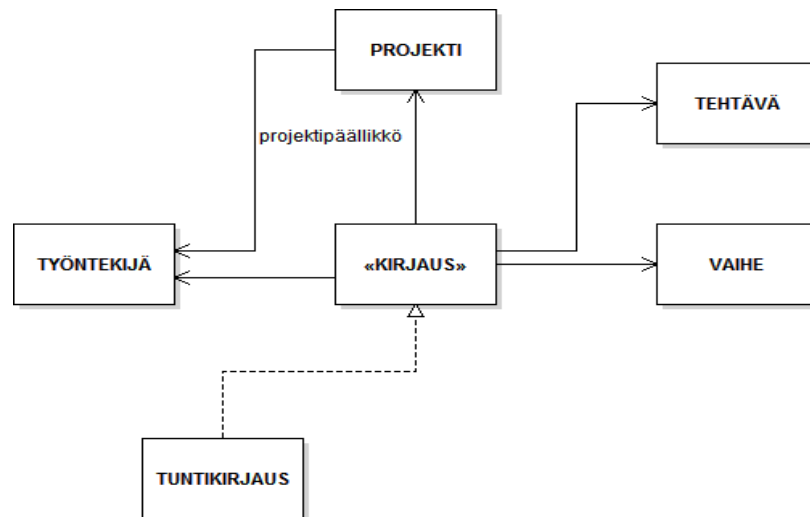
Ominaisuus	Kuvaus
Työ	Kenttä työlle
Käyttäjä	Kenttä käyttäjälle
Kuvaus	Kenttä työn kuvaukselle

KUVIO 9. Suunnitelma työntekijän liittamisestä työtehtäviin



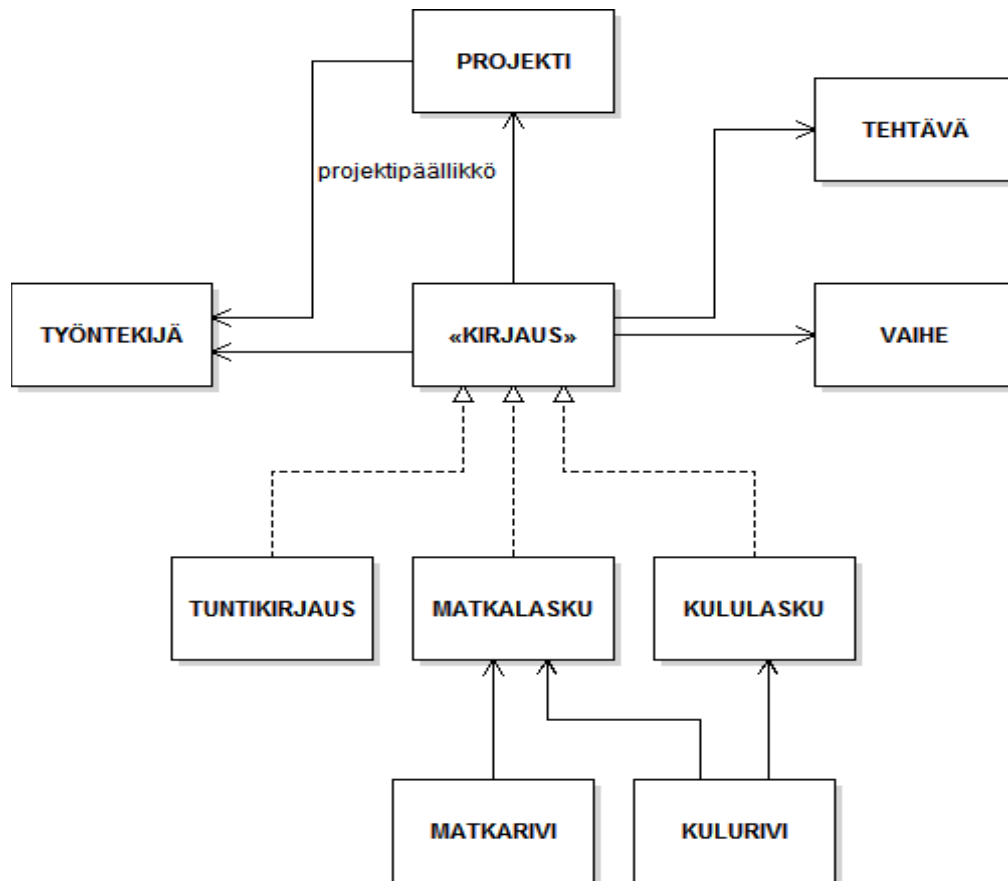
### 2.3 Tietokantasuunnittelu tuntikirjausohjelmistoon

Tietokannan taulut kuvataan UML-luokkanotaatiolla. Ylimmässä osiossa (luokan nimi) on taulun nimi ja seuraavassa (luokan metodit) avain tai avaimet (kuvio 13). Luokissa suositaan luonnollisten tai koostettujen avaimien sijasta surrogaatteja (keinotekoinen, yleensä juokseva järjestysluku). Alimmassa osiossa (luokan muuttujat) ovat taulun ominaisuudet. Notaatiossa nuoli (yhteys tai viite) alkaa siitä taulusta, jossa viiteavain on ja päättyy siihen tauluun, jonka tietueeseen viitataan. Tarvittaessa nuolen yhteydessä kuvataan yhteyden lukumääräsuhde. (Kujala 2003.) Alla olevassa kuvassa (kuvio 10) on tietovaraston rakenne kaikessa yksinkertaisuudessaan. Rakenne noudattaa tähtimallia, jossa kirjaus on faktataulu (tapahtumat) ja muut taulut dimensiotauluja (tapahtumiin verrattuna muuttumattomia tietoja).



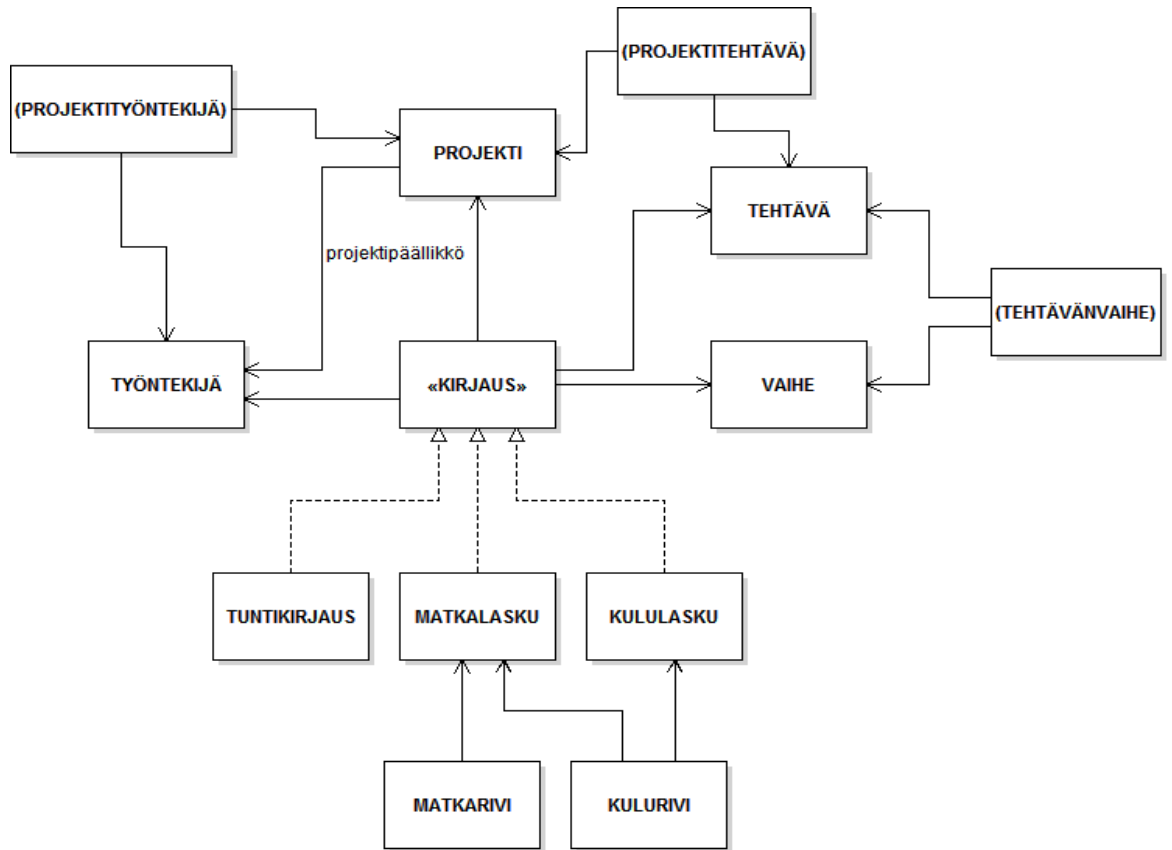
KUVIO 10. Tietokannan luonnos, tietovarasto

Kirjaus on rajapinta, jonka erilaiset kirjaukset (tuntikirjaus, kululasku, matkalasku) toteuttavat. Projektin toisen vaiheen toteutuksessa, joka aloitetaan opinnäytetyön osuuden jälkeen, lisätään matkalaskun ja kululaskun kirjausmahdollisuudet. Alla olevassa kuvassa (kuvio 11) nämä laskut on lisätty. Matka- ja kulurivit muuttavat rakenteen luomihiutalemalliksi.



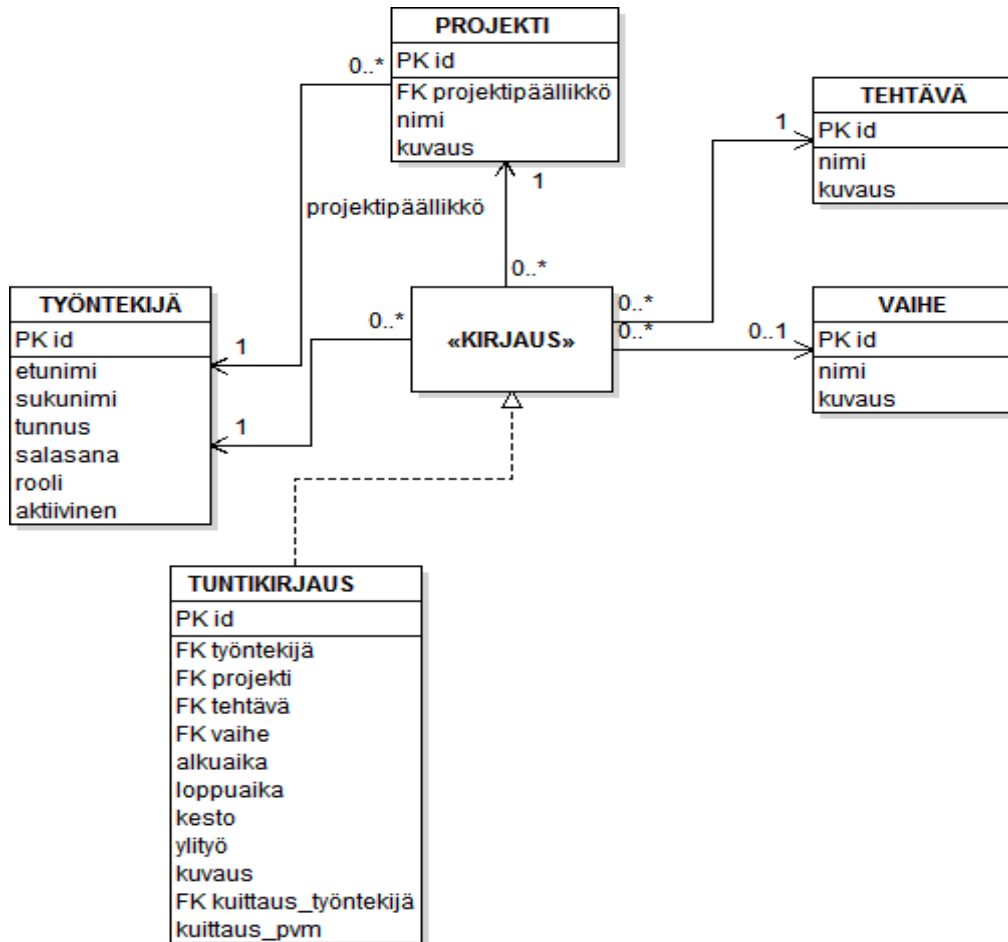
KUVIO 11. Lumihutalemalli, matka- ja kululaskut lisätty

Kun tarvitaan tarkempaa hallintaa siihen, kuka voi kirjata tunteja millekin projektille, lisätään tietokantaan erilliset liitostaulut (kuvio 12). Näillä tauluilla ei ole raporttien tai kirjausten kannalta merkitystä, vaan ne toimivat ainoastaan käyttöliittymässä rajoittaen käyttäjille annettuja vaihtoehtoja.



KUVIO 12. Käyttöliittymän aputaulut

Palvelun kehittyessä voidaan kirjausta tarkentavia tai kohdistavia tietoja lisätä uusien taulujen ja kirjaukseen lisättävien viittausten avulla. Näistä voi tulla lisäksi uusia liitostauluja, mutta niiden tuoma monimutkaisuus jää käyttöliittymän puolelle. Alla olevassa kuvassa (kuvio 13) on tarkemmin kuvattu taulujen rakenne sekä niiden lukumääräsuhteet toisiinsa. Kuvasta voidaan nähdä, että esimerkiksi jokaisella projektilla on aina yksi projektipäällikkö ja että työntekijä voi olla useamman tai ei yhdenkään projektin päällikkö.



KUVIO 13. Ensimmäisen vaiheen tarkempi rakenne

Kirjausta voidaan tarvittaessa denormalisoida sisältämään viitattavien tietueiden nimet (projektin nimi, työntekijän nimi) vain siinä tapauksessa, että tietokantahaut esimerkiksi raporteja luodessa havaitaan liian hitaiksi eikä indeksointia virittämällä päästä hyvään käytettävyyteen.

Kuvion 13 perusteella määriteltiin tietosisällöt työntekijälle, projektille, tehtävälle, vaiheelle ja tuntikirjaukselle.

### TYÖNTEKIJÄ

Työntekijästä voidaan käyttää myös nimitystä järjestelmän käyttäjä. Käyttäjiä ovat työntekijöiden lisäksi myös esimiehet ja tilitoimisto.

ominaisuus	PK	FK	NN	tyyppi	kuvaus
id	X		X	luku	Tietueen tunniste
etunimi				teksti	Työntekijän etunimi
sukunimi				teksti	Työntekijän sukunimi
tunnus				teksti	Käyttäjätunnus, yksilöivä
salasana				teksti	Salasana tiivisteenä
rooli				luku	Työntekijä, esimies, pääkäyttäjä tai tilitoimisto
aktiivinen				totuusarvo	Onko työntekijä palveluksessa vai ei

### PROJEKTI

Projekti voi olla yrityksen sisäinen tai yritykseltä tilattu projekti.

ominaisuus	PK	FK	NN	tyyppi	kuvaus
id	X		X	luku	Tietueen tunniste
tunnus				luku	Projektin numero (näkyvät esim. valittaessa kirjaukselle projektia)
projektipäällikkö		X	X	luku	Viittaus työntekijään joka johtaa projektia ja mm. hyväksyy tuntikirjaukset ja laskut
nimi			X	teksti	Projektin nimi (näkyvät esim. valittaessa kirjaukselle projektia)
kuvaus			X	teksti	Projektin tarkempi kuvaus (näkyvät esim. kirjausten ohjeena tai kuvauksena)
tuntimäärä				luku	Projektin arvioitu työmäärä
alkaa				pvm	Projektin alkamispäivämäärä, kirjauksia ei sallita tätä edeltäville päiville
loppuu				pvm	Projektin loppumispäivämäärä, kirjauksia ei sallita tämän jälkeisille päiville

## TEHTÄVÄ

Tehtävä on projektin sisäinen osakokonaisuus.

ominaisuus	PK	FK	NN	tyyppi	kuvaus
id	X		X	luku	Tietueen tunniste
tunnus				luku	Tehtävännumero (näkyvä esim. valittaessa kirjaukselle tehtävää)
nimi		X	X	teksti	Tehtävän nimi (näkyvä esim. valittaessa kirjaukselle tehtävää)
kuvaus		X	X	teksti	Tehtävän tarkempi kuvaus (näkyvä esim. kirjauksen ohjeena tai kuvauksena)
tuntimäärä		X	X	luku	Tehtävän arvioitu työmäärä

## VAIHE

Vaihe tarkoittaa tehtävää, jotta tuntikirjaukset voidaan paremmin kohdentaa tehtyyn työhön.

ominaisuus	PK	FK	NN	tyyppi	kuvaus
id	X		X	luku	Tietueen tunniste
tunnus				luku	Vaiheen numero (näkyvä esim. valittaessa kirjaukselle vaihetta)
nimi			X	teksti	Vaiheen nimi (näkyvä esim. valittaessa kirjaukselle vaihetta)
kuvaus			X	teksti	Vaiheen tarkempi kuvaus (näkyvä esim. kirjauksen ohjeena tai kuvauksena)

## TUNTIKIRJAUS

Tuntikirjaus on tietokannan faktataulu, johon jatkuvasti lisätään uusia kirjauksia. Myös vanhojen kirjausten muuttaminen tietyn ajan sisällä on mahdollista.

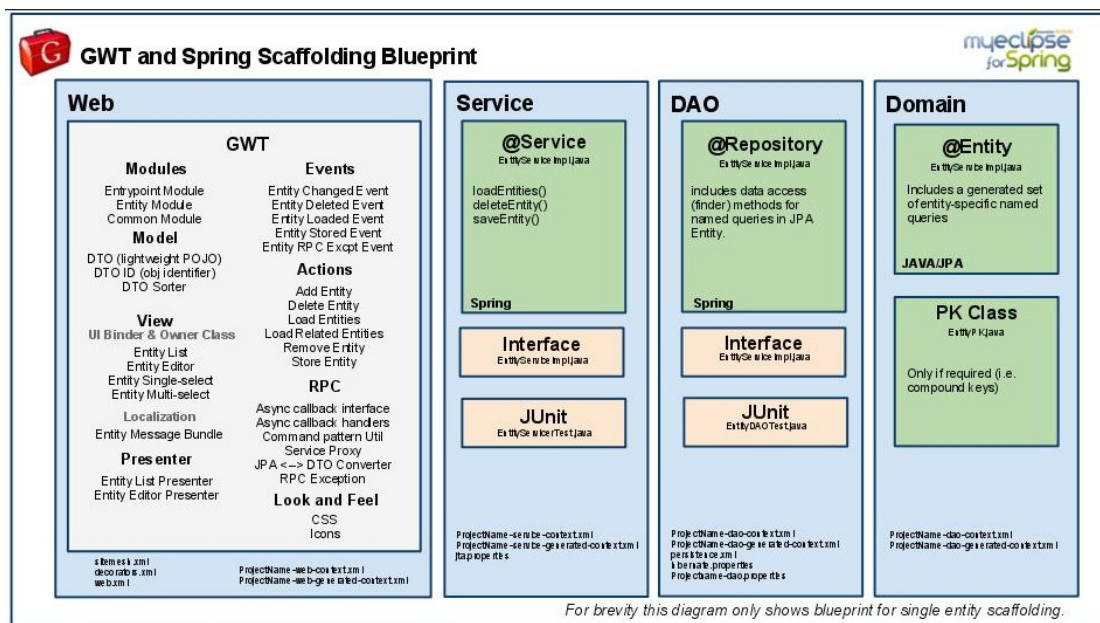
<b>ominaisuus</b>	<b>PK</b>	<b>FK</b>	<b>NN</b>	<b>tyyppi</b>	<b>kuvaus</b>
id	X		X	luku	Tietueen tunniste
työntekijä		X	X	luku	Viittaus työntekijään jota kirjaus koskee
projekti		X	X	luku	Viittaus projektiin johon kirjaus kohdistuu
tehtävä		X	X	luku	Viittaus tehtävään johon kirjaus kohdistuu
vaihe		X		luku	Viittaus vaiheeseen johon kirjaus kohdistuu
alkuaika			X	pvm / aika	Tuntikirjauksen alkuajankohta tai päivämäärä
loppuaika				pvm / aika	Tuntikirjauksen loppuajankohta
kesto			X	aika / luku	Tuntikirjauksen kesto, syötetään loppuajan asemasta tai lasketaan loppuajan ja alkuajan erotuksena
ylityö				totuusarvo	Sisältääkö kirjaus ylityötä
kuvaus			X	teksti	Vapaa kuvaus kirjauksen työstä
kuittaus työntekijä	/	X		luku	Viittaus projektipäällikköön tai pääkäyttäjään joka on tarkistanut ja hyväksynyt kirjauksen
kuittaus aika	/			pvm / aika	Päivämäärä ja aika jolloin kirjaus on tarkistettu





avulla alkuun pääsee nopeasti, mutta sen luomat näytöt ovat käyttöliittymän suunnittelun kannalta melko rajoittuneita eikä se tarjoa CRUD (Create, Update, Delete) -näytymien lisäksi muunlaisia mahdollisuuksia räätälöidä sovellusta kun tyylien ja kuvien muuttaminen.

Yrityksessä oli tuntikirjausprojektin kanssa samaan aikaan alkamassa toinen projekti, jossa tarvittiin tietokantakytkentää ja graafisia ominaisuuksia. Koska GWT oli jo ennestään vähän tuttu ja sen voisi integroida yhteen Spring Roon kanssa, päätettiin tässä työssä tutkia, kuinka tuntikirjaussovelluksen toteutus onnistuisi näillä työkaluilla. Samalla tietotaitoa voisi siirtää myös yrityksen toiseen projektiin. GWT tarjoaa mahdollisuuden graafisten ominaisuuksien toteuttamiseen, kun taas Spring Roo rakentaa tietokantaliitännät näppärästi. GWT on Googlen kehittämä avoimen lähdekoodin työkalu, joka antaa mahdollisuuden luoda ja ylläpitää monimutkaisia web-sovelluksia Java-ohjelmointikielellä. GWT SDK sisältää Java API:n ja widgetit. Nämä mahdollistavat AJAX (Asynchronous JavaScript And XML) -sovelluskehityksen Java-ohjelmointikielellä. Kuviossa 15 on kuvattuna Spring + GWT- sovelluksen arkkitehtuuri.



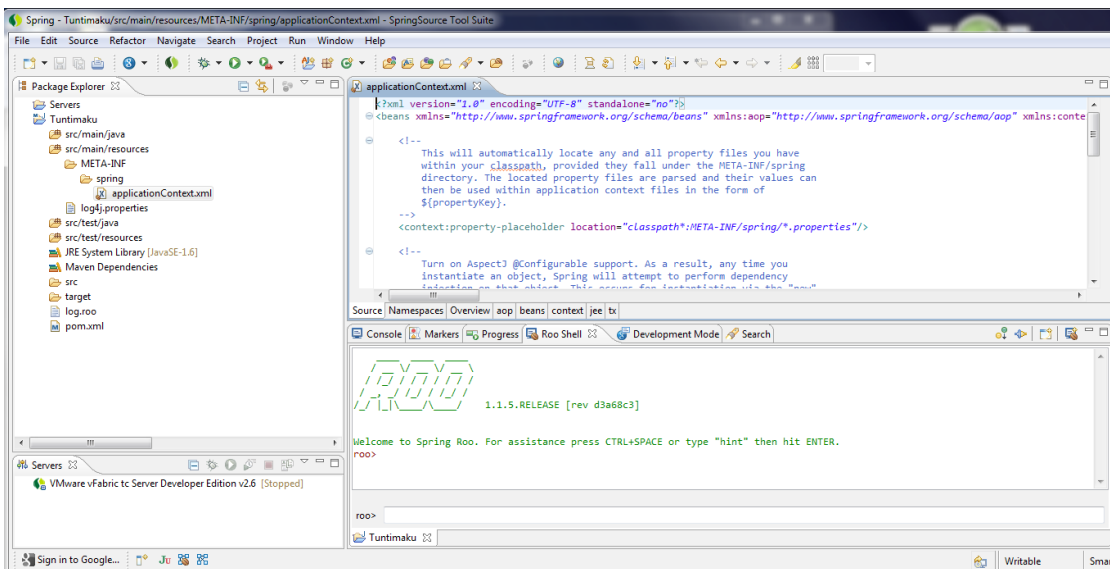
KUVIO 15. Spring + GWT-sovelluksen arkkitehtuuri. (Genuitec 2010)

### 3.2 Ympäristön pystytys

Jotta projektia päästään tekemään, täytyy olla asennettuna seuraavat ohjelmistot:

- SpringSourceTool 2.9.0 Release
- Google Plugin for Eclipse 3.7
- Spring Roo 1.1.5
- GWT SDK 2.3.0
- GWT Designer
- Apache Tomcat 6.

Ohjelmistoversioiden kanssa täytyy olla tarkkana, sillä uusimmat Roon ja GWT:n versiot (vuoden 2012 alkupuolella julkaistut) toimivat hyvin huonosti yhteen. Sopiva ohjelmistokokonaisuus löytyi lopulta kokeilun ja erehdyksen kautta ja sovelluskehitys päästiin aloittamaan. Koska ohjelmistoversiot (GWT, Spring Roo) kehittyvät koko ajan, tullaan uusien versioiden julkaisemisen myötä selvittämään, onko toiminnallisuutta saatu parannettua. SpringSourceTool -työkalulla luodaan SpringRoo-projekti, johon ominaisuuksia ryhdytään rakentamaan, (kuva 1). Projektiin syntyy hakemistorakenne ja SpringRoo-komentotulkki avautuu (kuvassa 1 oikealla alhaalla).



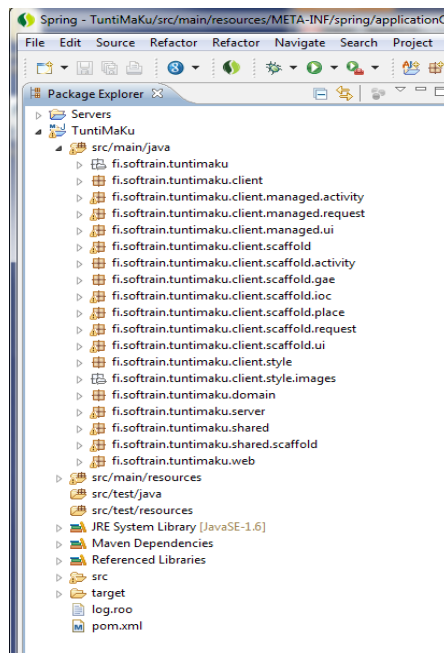
KUVA 1. SpringRoo-projekti SpringSourceTool -työkalulla luotuna

### 3.3 GWT web -sovelluksen rakenne ja toiminnallisuus

Tässä sovelluksessa näkymät luodaan Spring Rooilla käyttäen GWT (Google Web Toolkit) -lisäosaa. GWT-lisäosa asennetaan Spring Roo -komennolla `web gwt setup`. Web gwt setup luo kuvion 16 mukaisen hakemistorakenteen. Samalla olemassa olevasta Spring Roo -projektista tulee GWT web -sovellus. GWT web -sovellus rakentuu MVP-mallin mukaan (Model, View, Presenter). Nämä kuvataan luvuissa 3.3.1 - 3.3.3. (Ramsdale 2010.) Kaksi tärkeintä pakettia, jotka `web gwt setup` -komennolla luodaan, ovat `managed` ja `scaffold`.

`Managed`-paketti sisältää kaikki Spring Rooon hallinnoimat tiedostot. Näitä tiedostoja luodaan ja päivitetään silloin, kun muutoksia tehdään `domain`-pakettiin.

`Scaffold`-paketti sisältää staattiset tiedostot, joiden perusteella sovellus rakentuu. Näitä tiedostoja ei sovelluksen kehitysvaiheessa tarvitse päivittää tai muuttaa. Ne kopioidaan Spring Roo -projektista `web gwt setup` -komennon suorituksen yhteydessä.



KUVIO 16. Spring Roo + GWT -sovelluksen hakemistorakenne

#### 3.3.1 Model

Model koostuu JavaBean luokista, jotka luodaan Spring Rooon komennolla `entity`. Nämä luokat sijaitsevat `fi.softtrain.tuntimaku.domain`-paketissa. Tietokantayhteyksiä

varten asennetaan JPA (Java Persistence API), joka mahdollistaa kytkennän MYSQL-kantaan. Tämä tapahtuu syöttämällä Spring Roo -komentotulkille komento `persistence setup` (esim. `persistence setup –provider HIBERNATE –database MySQL –databaseName tuntimaku –username 'mysql kannan käyttäjän nimi' –password 'mysql kannan salasana'`). Käytettävä rajapinta tässä projektissa on hibernate. Hibernate on Java-pohjainen ORM (Object-relational mapping) -kirjasto, joka tarjoaa rajapinnan tietokannan ja Java-luokkien välille.

Spring Roon komennolla `entity –class`, muodostuu projektiin luokat tietokannassa olevista tauluista. Komennossa `class` määrittää luokan sijainnin ja nimen (esim. `entity –class ~.domain.worker –testAutomatically`). Sijainnin yhteydessä `~` vastaa sitä pakettia, johon projekti on luotu ja lisäämääre `–testAutomatically` lisää luokan automaattitesteihin. Luokkiin ja samalla myös luotuu tietokantatauluun lisätään kenttiä Spring Roo -komennolla `field` tyyppi `–kenttänimi` (esim. `field string –fieldName username –sizeMin 4 –sizeMax 40 –notNull –unique`). Viitattaessa toisiin luokkiin tyyppi on `reference` ja sen lisäksi annetaan viittauksen tyyppi luokkana (esim. `field reference –fieldName worker –type fi.softtrain.tuntimaku.domain.worker –cardinality MANY_TO_ONE –notNull`). Viittauksen yhteydessä yhteen- tai moneen riippuvuus annetaan määreellä `cardinality`. Esimerkissä edellä on käytetty `MANY_TO_ONE` riippuvuutta, mikä tarkoittaa, että useampi tämäntyyppinen olio voi viitata samaan viitattavaan olioon.

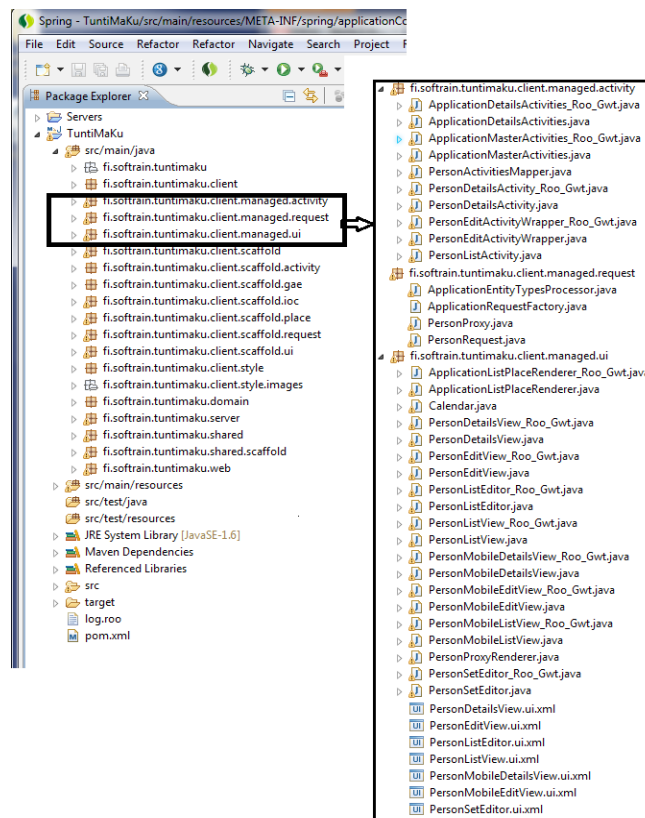
Spring Roo luo komentojen perusteella Java-luokat sekä muutamia AspectJ-tiedostoja. AspectJ on tehokas ja kehittynyt AOP (aspect oriented programming) -rakenne, joka tukee monia suuria järjestelmiä. (Alex, Schmidt, Stewart, Tyrrell, Swan 2011.) Spring Roo -projektit käyttävät AspectJ:n tehokkaita IDT (inter-type declaration) -ominaisuuksia. Se tarkoittaa, että nämä tiedostot kytkevät sovelluksen monta toiminnallisuutta yhteen yhdessä paikassa. Näihin tiedostoihin on luotu `get-` ja `set-` funktiot, `toString()`-toteutus sekä funktiot olioiden hakemiseen ja tallentamiseen tietokannasta. AspectJ -tiedostoiden nimessä on merkintä `*_Roo_*.aj`, esimerkiksi `Task_Roo_Entity.aj`. Nämä AspectJ -tiedostot ovat Spring Roon hallinnoimia ja niihin ei kannata muutoksia tehdä, koska Roo ylikirjoittaa ne aina, kun komentotulkki avataan.

### 3.3.2 View

View sisältää kaikki UI (User Interface) -komponentit. Kun GWT-projektin alkuasennus on tehty, kaikki muutokset kohdistuvat pääsääntöisesti managed-pakettiin, joka koostuu activity-, request- ja ui -paketeista (kuvio 17). Activity-paketti sisältää kaiken toiminnallisuuden, joka liittyy GWT:n MPV-malliin.

Request-paketti sisältää ne tiedostot, jotka liittyvät tiedonsiirtoon RequestFactory-ra-japinnan kautta. Jokaista entityä kohden luodaan Proxy- ja Request-luokat. Proxy-luokka edustaa palvelinpuolen entityä ja Request-luokka palvelua.

Ui-paketti sisältää näkymät ja kaikki käyttöliittymään liittyvät luokat ja tiedostot. Kun entity luodaan (luku 3.3.1), syntyy kahdeksan tiedostoa ja niitä vastaavat ui.xml-tiedostot jokaista luotua entityä kohti.



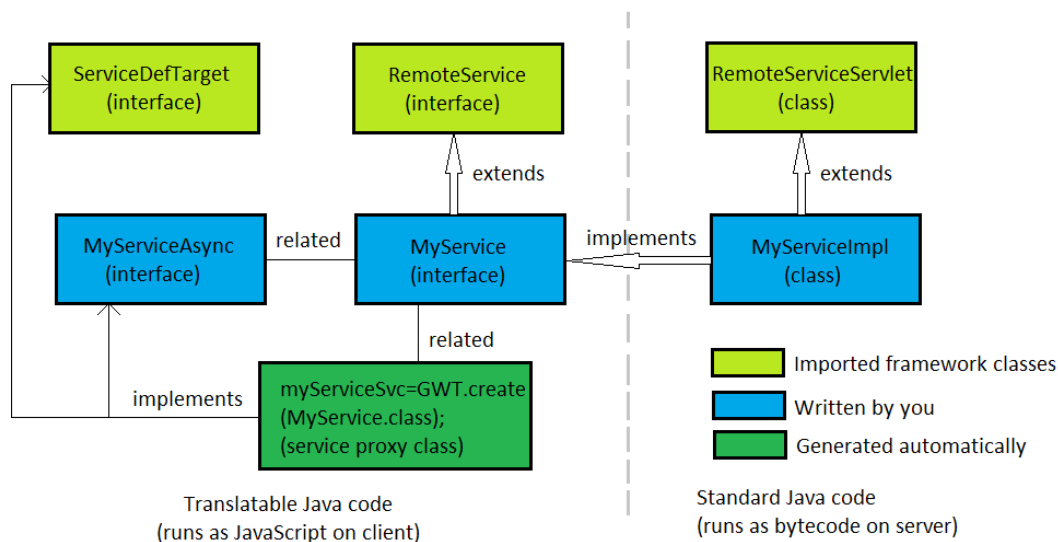
KUVIO 17. Managed-pakkauksen tiedostot

### 3.3.3 Presenter

Presenter käsittää toimintalogiikan sisältäen näyttöjen väliset siirtymät, datan siirron clientin ja serverin välillä (Remote Procedure Call) sekä historiatietojen keräilyyn.

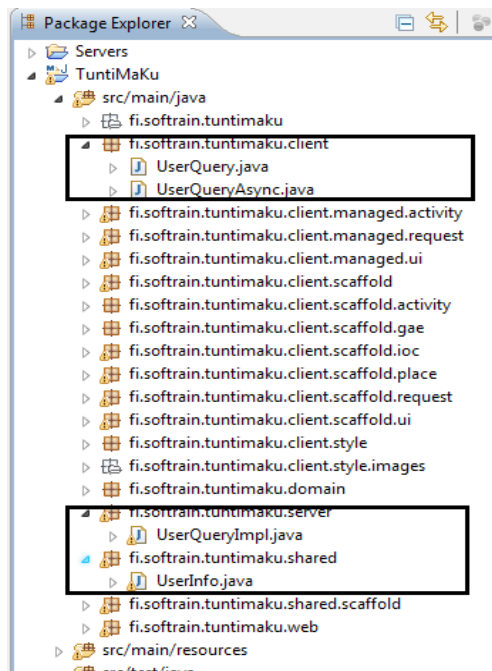
GWT-RPC (Remote Procedure Call) tarvitaan tiedonsiirtoon asiakkaan (client) ja palvelimen (server) välille, silloin kun siirretään tietoa, jota ei ole suoraan Spring Roon avulla luotu tietokannasta näkymäksi. Tiedonhaku tapahtuu palvelimelta asynkronisesti, mikä tarkoittaa, että pyynnön ja vastauksen väli voi olla pitkä ja sitä ei jäädä odottamaan. Palvelin käsittelee pyynnön ja palauttaa vastauksen. RPC-rajapinnan määrittelemiseksi (kuvio 18) täytyy kirjoittaa kolme komponenttia (kuviossa 18 MyService.java, MyServiceAsync.java ja MyServiceImpl.java).

Ensimmäiseksi tarvitaan rajapinta (MyService), joka määrittelee palvelun, jota laajennetaan RemoteService-rajapinnan avulla. Se sisältää listan kaikista RPC-metodeista, joita tarvitaan. Seuraavaksi luodaan luokka (MyServiceImpl), jonka laajenuksena on RemoteServiceServlet. Tämä luokka toteuttaa myös ensimmäiseksi luodun rajapinnan. Kolmanneksi määritellään asynkroninen rajapinta (MyServiceAsync), jota kutsutaan clientin puolelta. (Google Web Toolkit 2012a.)



KUVIO 18. GWT-RPC rakenne

RPC-komponentit sijoitetaan projektissa kolmeen eri pakettiin, jotka näkyvät alla olevassa kuvassa (kuvio 19). Edellä kerrottujen tiedostojen lisäksi tarvitaan luokka, joka toteuttaa Serializable-rajapinnan. Rajapinta tarvitaan Spring Roolla luotujen objektien siirtoon verkon yli GWT-RPC palvelun kautta. Luokassa määritellään set()- ja get()-palvelut kaikille siirrettäville parametreille ja niiden paluuarvoille. Tämä luokka sijoituu `fi.softrain.tuntimaku.shared`-pakettiin.



KUVIO 19. RPC- rajapinnan toteuttavat tiedostot

### 3.5 Spring Security

Käyttäjien hallintaa varten tarvitaan Spring Security-lisäosa, joka asennetaan Spring Roo -komennolla `security setup`. (Mularien 2010.) Komento luo `applicationContext-security.xml` -tiedoston, joka sisältää käyttäjien hallintaan tarvittavat asetukset ja rajoitukset. Käyttäjien hallintaa varten luodaan kaksi tietokantataulua, toinen käyttäjälle (person) ja toinen käyttäjäroolille (userrole). Käyttäjää varten lisätään tietokantaan entity `userrole`, jossa on kentät `name` ja `description`. Käyttäjän tietokantatauluun `person` lisätään kenttä `userrole` (field `set` – fieldName `userrole` –type `fi.softrain.tuntimaku.domain.Userrole` –cardinality `MANY_TO_MANY`), johon voidaan määrittää useita rooleja kullekin käyttäjälle. Käyttäjät tallennetaan rooleinensa tietokantaan, jotta tunnistus olisi mahdollista.

ApplicationContext-security.xml -tiedostossa suojataan kaikki servletit, kuten esimerkiksi gwtRequest ja RPC-palvelut, jotta sovellukseen ei pääse ilman kirjautumista sisään.

```
<intercept-url pattern="/gwtRequest**" access="isAuthenticated()" />
<intercept-url pattern="/applicationScaffold/UserQuery**" access="isAuthenticated()" />
```

Securityn entry-point muutetaan Http403ForbiddenEntryPoint:ksi, jotta kirjautumissivua ei palauteta RPC-palvelun kautta.

```
<http auto-config="false" use-expressions="true" entry-point-ref="http403ForbiddenEntryPoint">
```

Lisäksi ApplicationContext.xml:ään lisätään

```
<bean id="http403ForbiddenEntryPoint"
      class="org.springframework.security.web.authentication.Http403ForbiddenEntryPoint" />
```

Tämä entry-point palauttaa HTTP status koodin 403, jonka perusteella ScaffoldDesktopApp.java tiedostossa kirjautuminen ohjataan uudelle sivulle.

```
private void init() {
    GWT.setUncaughtExceptionHandler(new GWT.UncaughtExceptionHandler() {
        public void onUncaughtException(Throwable e) {
            if (e instanceof UmbrellaException) {
                for (final Throwable e2: ((UmbrellaException) e).getCauses()) {
                    if (e2.getMessage().startsWith("Server Error 403")) {
                        Window.open("login", "_self", null);
                    } else {
                        Window.alert("Error: " + e.getMessage());
                        Log.log(Level.SEVERE, e.getMessage(), e);
                    }
                }
            } else {
                Window.alert("Error: " + e.getMessage());
                Log.log(Level.SEVERE, e.getMessage(), e);
            }
        }
    });
}
```

Käyttäjän haku tehdään hakulausekkeella

```
users-by-username-query="SELECT person.username AS username, person.password as password,
person.active as enabled FROM person where person.username=?"
```

ja käyttäjän roolien haku lausekkeella

```
authorities-by-username-query="SELECT person.username as username, A.name as authority FROM
person left join person_user_role UA on person.id=UA.person left join user_role A on UA.user_role =
A.id WHERE person.username=?"
```

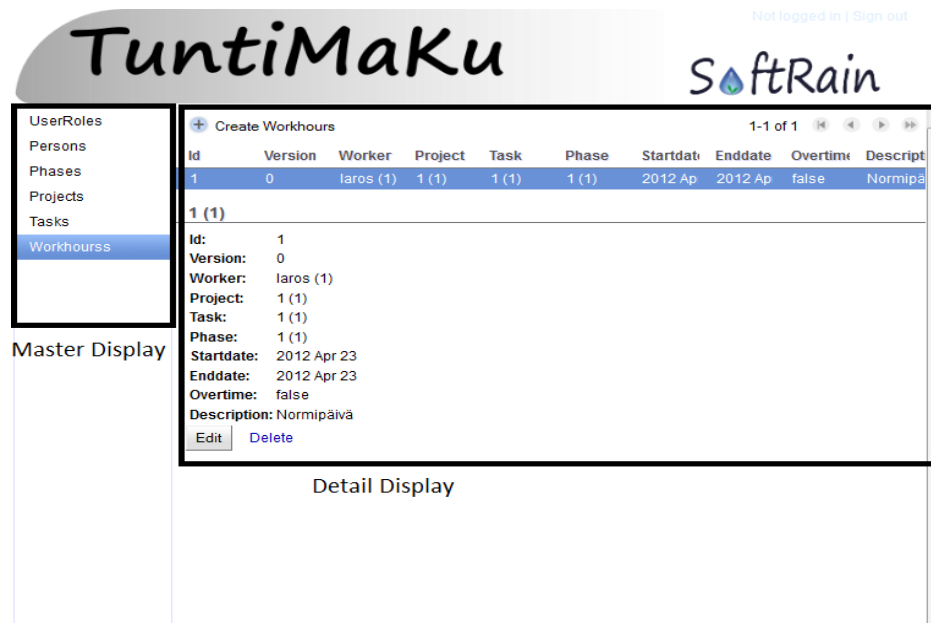


### 3.6 Näyttöjen rakenne ja muodostuminen

Tuntikirjaussovelluksen käyttöliittymään toteutetut näytöt voidaan jakaa kahteen osaan niiden toteutuksen mukaan. Nämä ovat Spring Roo:n muodostama perusnäyttö ja GWT:llä räätälöity näyttö osaksi perusnäyttöä. Käyttöliittymätoteutuksen pohjana on perusnäkö, johon ominaisuuksia lisätään. Sivujen välillä ei ole monimutkaista navigointia, mikä helpottaa käyttäjän liikkumista sivustolla.

#### 3.6.1 SpringRoo:n avulla toteutettu perusnäkö

Spring Roo -komentoilla toteutettu perusnäkö jakautuu kahteen osaan: master display ja detail display (kuviokuva 20). Kuvassa vasemmalla on master display. Se sisältää listan kaikista niistä sivuista, joihin käyttäjällä on mahdollisuus siirtyä. Kuvassa oikealla oleva Detail display sisältää CRUD (Create, update, delete) toiminnallisuuden. Ylimmäiseksi oikealle syntyy taulukko tietokantaan luoduista tiedoista ja sen alapuolelle yksityiskohtaiset tiedot kustakin taulukon rivistä, joita pystyy muuttamaan ja poistamaan. Tämä Spring Roo:lla luotu perusnäkömalli syntyy GWT:lle puolesta tunnissa samalla, kun domain-pakettiin (fi.softrain.tuntimaku.domain) lisätään uusia tiedostoja entity -komennolla. (kuvattu luvussa 3.3.1.) Nämä tiedostot myös päivittyvät automaattisesti, jos domain-paketin sisältämiin tiedostoihin tehdään muutoksia.



KUVIO 20. Päänäyttö, johon master ja detail display rajattuna

\*ListView.java, missä \* kuvaa luotua ominaisuutta (UserRoleListView.java), hallitsee perussivulle luotua taulukkonäkymää. Taulukon sarakkeiksi tallentuu automaattisesti kaikki tietokantaan syötetyt kentät, mutta niitä pystyy koodissa muokkaamaan ja näyttämään vain ne tiedot, joita tarvitaan. Esimerkiksi käyttäjäroolin tiedoista halutaan näyttää taulukossa vain nimi, jolloin se määritellään UserRoleListView.java -tiedoston init() -funktioon seuraavalla tavalla.

```

@Override
public void init() {
    paths.add("name");
    table.addColumn(new TableColumn<UserRoleProxy>() {

        Renderer<java.lang.String> renderer = new AbstractRenderer<java.lang.String>() {

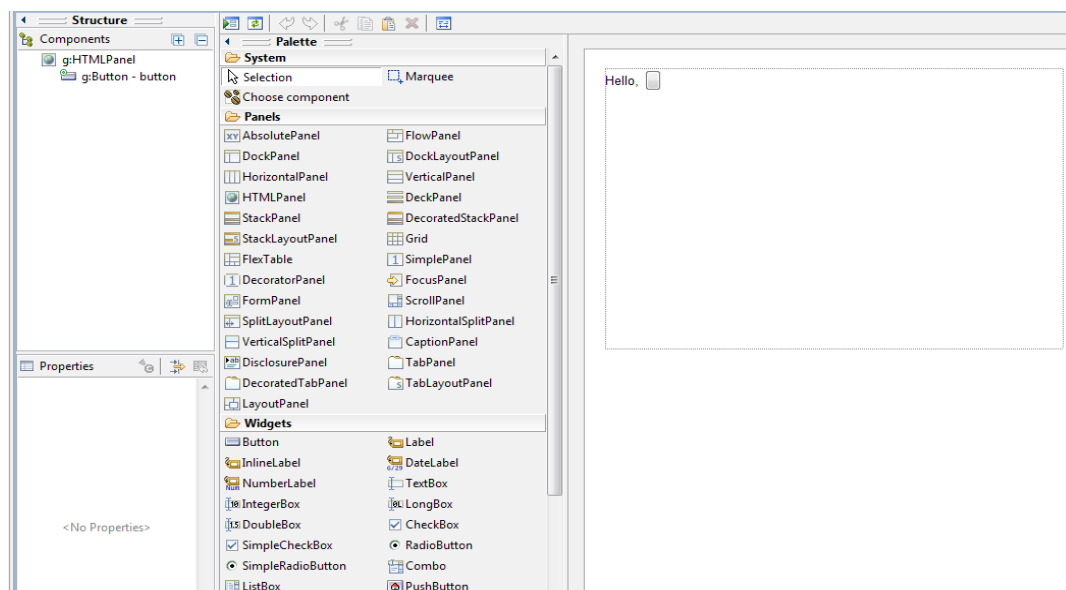
            public String render(java.lang.String obj) {
                return obj == null ? "" : String.valueOf(obj);
            }
        };

        @Override
        public String getValue(UserRoleProxy object) {
            return renderer.render(object.getName());
        }
    }, "Name");
}

```

### 3.6.2 Oman näkymän luonti

Tässä kappaleessa kerrotaan, kuinka Spring Rooilla luotua perusnäköön (kuvio 20) saadaan liitettyä GWT Designer -työkalulla luotu näkymä. Aluksi luodaan Spring Roo -projektiin uusi UIBinder -luokka. Tähän luokkaan voi GWT:n graafisessa suunnittelutilassa (Design) luoda näyttöjä sen tarjoamilla komponenteilla (kuva 2). Graafisten ominaisuuksien kirjasto gwt-graphics.jar lisätään ohjelmointiympäristöön, jolloin voidaan piirtää vapaata viivaa, kuten ympyrä ja suorakaide.



KUVA 2. GWT Designer -näkö

Tietokantahakuja varten tarvitaan RPC-palvelut, jotta tietoa saadaan siirrettyä clientin ja serverin välillä (kuvattu luvussa 3.3.3).

Luotu näkymä saadaan liitettyä Spring Roon peruspohjaan `fi.softrain.tuntimaku.client.managed.activity` -paketissa sijaitsevan ”entity”`DetailsActivity.java` -tiedoston `start()`-metodissa. Esimerkiksi tuntikirjaus sovelluksessa tuntikirjaus näkymään halutaan saada kalenterinäkymä, jolloin muutoksia tehdään `WorkhoursDetailsActivity.java` -tiedostoon.

```
public void start(AcceptsOneWidget displayIn, EventBus eventBus) {
    this.display = displayIn;
    Receiver<EntityProxy> callback = new Receiver<EntityProxy>() {
        public void onSuccess(EntityProxy proxy) {
            if (proxy == null) {
                placeController.goTo(getBackButtonPlace());
                return;
            }
            if (display == null) {
                return;
            }
            view.setValue((WorkhoursProxy) proxy);
            CalendarDetailView calendarDetailsView = CalendarDetailView.instance();
            CalendarDetailsInstance calendarDetailsInstance = calendarDetailsView.setCalendar();
            display.setWidget(view);
        }
    };
    find(callback);
}
```

Tällä tavoin saadaan aikaan uusi näkymä, kuvassa (kuvio 21) Customized Display.

The screenshot shows the TuntiMaku application interface. At the top, there is a navigation bar with the 'TuntiMaku' logo and the 'SoftRain' logo. A user status indicator shows 'Not logged in | Sign out'. On the left, there is a 'Master Display' sidebar menu with options: UserRoles, Persons, Phases, Projects, Tasks, and Workhourss (highlighted). The main area is divided into two sections: 'Detail Display' and 'Customized Display'. The 'Detail Display' section shows a table with one row of workhour data and a list of details for that row. The 'Customized Display' section shows a bar chart representing the workhour data over time.

Id	Version	Worker	Project	Task	Phase	Startdate	Enddate	Overtime	Description
1	0	Iaros (1)	1 (1)	1 (1)	1 (1)	2012 Apr 23	2012 Apr 23	false	Normipäivä

1 (1)

Id: 1  
Version: 0  
Worker: Iaros (1)  
Project: 1 (1)  
Task: 1 (1)  
Phase: 1 (1)  
Startdate: 2012 Apr 23  
Enddate: 2012 Apr 23  
Overtime: false  
Description: Normipäivä

Edit Delete

Detail Display

Customized Display

Back

KUVIO 21. Oman näkymän sijoittuminen perusnäyttöön

### 3.7 Toteutetut ominaisuudet

Tässä työssä toteutettiin näymät työntekijälle, esimiehelle tai projektipäällikölle ja ylläpitäjälle. Spring Roo + GWT -ympäristössä on mahdollista sisällyttää monia toimintoja saman perusnäytön ympärille, jolloin jokaiselle ominaisuudelle ei tarvita rakentaa erillistä näyttöä. Kaikki tarvittavat toiminnallisuudet avautuvat samaan näyttöpohjaan, mikä tuo selkeyttä käyttäjän näkökulmasta, sillä monimutkaisia sivunvaihtoja ja navigointia ei tarvita.

#### 3.7.1 Kirjautuminen

Kun sovellus ladataan palvelimelle saadaan kuvan 3 mukainen näkymä.

#### KUVA 3. Palveluun kirjautuminen

Käyttäjä kirjautuu palveluun annetulla käyttäjätunnuksella ja salasanalla. Salasanaa ei lähetetä tekstinä vaan siitä luodaan hajautusfunktiolla tiiviste. Palvelu vertaa käyttäjätunnusta ja salasanaa aktiivisiin käyttäjiin ja jos käyttäjää ei tunnisteta palautetaan kirjautumissivu (kuva 4) ja tieto siitä, että käyttäjää ei tunnistettu. Kun käyttäjä tunnustetaan, avataan selaimeen tuntikirjausohjelmiston päänäyttö. Päänäytön toiminnot riippuvat kirjautuneen käyttäjän roolista. Uloskirjautuminen tapahtuu päänäytöltä, josta palataan takaisin kirjautumisnäyttöön.

The screenshot shows the TuntiMaku login interface. At the top, the logo 'TuntiMaku' is on the left and 'SoftRain' is on the right. Below the logo is a 'Spring Security Login' box. Inside this box, a red error message reads: 'Your login attempt was not successful, try again. Reason: Bad credentials.' Below the message are two input fields: 'Name' and 'Password'. At the bottom of the box are two buttons: 'SUBMIT' and 'RESET'. Below the login box, there are links for 'Home', 'Language', and 'Theme', and a 'Sponsored by SpringSource' logo.

#### KUVA 4. Kirjautuminen epäonnistui

Käyttäjän tunnistus toimii sovelluksessa hyvin ja kirjautuminen onnistuu eri käyttäjä-tunnuksilla ja rooleilla. Näkymät on rakennettu kolmelle käyttäjäroolille, mutta niihin jäi vielä vähän parannettavaa. Näkymiä siistitään ja toiminnallisuutta parannellaan käyttökokemusten mukaan.

#### 3.7.2 Päänäyttö

Seuraavissa kuvissa esitellään päänäytön näkymät, kun kirjautuneena on työntekijä (kuva 5), esimies tai projektipäällikkö (kuva 6) ja ylläpitäjä (kuva 7).


The screenshot shows the TuntiMaku main dashboard for a worker role. At the top, the logo 'TuntiMaku' is on the left and 'SoftRain' is on the right. Below the logo is a 'Not logged in | Sign out' link. The main content area has a 'Workhourss' tab selected, with a '+ Create Workhours' button. Below this is a table with the following data:

Id	Version	Worker	Project	Task	Phase	Startdate	Enddate	Overtime	Descripti
1	0	Iaros (1)	1 (1)	1 (1)	1 (1)	2012 Apr	2012 Apr	false	Normipä

KUVA 5. Päänäyttö työntekijän roolilla kirjautuneelle käyttäjälle

Not logged in | Sign out

# TuntiMaku



Projects


[+ Create Project](#)
1-3 of 3

	Id	Version	Startdate	Enddate	Active
Workhourss	1	0	2012 Apr 24	2012 Jun 15	true
	2	0	2012 May 17	2012 May 31	true
	3	0	2012 May 14	2012 Jun 29	true

KUVA 6. Päänäyttö esimiehen tai projektipäällikön roolilla kirjautuneelle käyttäjälle

Not logged in | Sign out

# TuntiMaku



UserRoles

[+ Create UserRole](#)
1-2 of 2

	Id	Version	Name
Persons	1	0	ROLE_ADMIN
Phases	2	0	ROLE_USER
Projects			
Tasks			
Workhourss			

KUVA 7. Päänäyttö ylläpitäjän roolilla kirjautuneelle käyttäjälle

### 3.7.3 Työntekijän toiminnallisuus

Työntekijä pystyy suorittamaan niiden projektien tuntikirjaukset, jotka hänelle on merkitty. Työntekijä voi luoda uuden kirjauksen (kuva 8), poistaa jo luodun kirjauksen (kuva 9) tai muuttaa kirjauksia (kuva 10).

Not logged in | Sign out

# TuntiMaku



Workhours + Create Workhours 1-1 of 1

Id	Version	Worker	Project	Task	Phase	Startdate	Enddate	Overtime	Description
1	0	laros (1)	1 (1)	1 (1)	1 (1)	2012 Apr	2012 Apr	false	Normipä

**New Workhours**

Worker:

Project:

Task:

Phase:

Startdate:

Enddate:


Overtime:

Description:

KUVA 8. Tuntikirjausnäkyvä, tuntikirjauksen lisääminen

Not logged in | Sign out

# TuntiMaku



Workhours + Create Workhours 1-1 of 1

Id	Version	Worker	Project	Task	Phase	Startdate	Enddate	Overtime	Description
1	0	laros (1)	1 (1)	1 (1)	1 (1)	2012 Apr	2012 Apr	false	Normipä

**1 (1)**

Id: 1

Version: 0

Worker: laros (1)

Project: 1 (1)

Task: 1 (1)

Phase: 1 (1)

Startdate: 2012 Apr 23

Enddate: 2012 Apr 23


Overtime: false

Description: Normipäivä

KUVA 9. Tuntikirjauksen poistaminen

Not logged in | Sign out

# TuntiMaku



Workhours + Create Workhours 1-1 of 1

Id	Version	Worker	Project	Task	Phase	Startdate	Enddate	Overtime	Description
1	0	laros (1)	1 (1)	1 (1)	1 (1)	2012 Apr	2012 Apr	false	Normipä

**Edit Workhours**

Worker:

Project:

Task:

Phase:

Startdate:

Enddate:

Overtime:

Description:

KUVA 10. Tuntikirjauksen muuttaminen

Tuntikirjausnäkyään toteutetaan myöhemmässä vaiheessa kalenterinäkymä, jossa näkee koko viikon kirjaukset kerralla. Toteutus tulee tähän näyttöön tuntikirjaustoimintojen alapuolelle. Toteutus tehdään oman käyttöliittymänäytön toteutuksen tavalla, joka on kuvattu luvussa 3.6.2.


#### 3.7.4 Esimiehen tai projektipäällikön toiminnallisuus

Esimies tai projektipäällikkö voi työntekijän toiminnallisuuden lisäksi lisätä (kuva 11), muuttaa (kuva 12) ja poistaa projekteja (kuva 13).



Not logged in | Sign out

# TuntiMaku



Projects

+ Create Project
1-3 of 3

	Id	Version	Startdate	Enddate	Active
Workhours	1	0	2012 Apr 24	2012 Jun 15	true
	2	0	2012 May 17	2012 May 31	true
	3	0	2012 May 14	2012 Jun 29	true

**New Project**

Startdate:


Enddate:

Active:

KUVA 11. Projektin lisääminen

Not logged in | Sign out

# TuntiMaku



Projects

+ Create Project
1-3 of 3

	Id	Version	Startdate	Enddate	Active
Workhours	1	0	2012 Apr 24	2012 Jun 15	true
	2	0	2012 May 17	2012 May 31	true
	3	0	2012 May 14	2012 Jun 29	true

**Edit Project**

Startdate:


Enddate:

Active:

KUVA 12. Projektin muuttaminen

Not logged in | Sign out

# TuntiMaku



**Projects** + Create Project 1-3 of 3

Workhourss	Id	Version	Startdate	Enddate	Active
	1	0	2012 Apr 24	2012 Jun 15	true
	2	0	2012 May 17	2012 May 31	true
	3	0	2012 May 14	2012 Jun 29	true

**2 (2)**

**Id:** 2  
**Version:** 0  
**Startdate:** 2012 May 17  
**Enddate:** 2012 May 31  
**Active:** true


KUVA 13. Projektin poistaminen

### 3.7.5 Ylläpitäjän toiminnallisuus

Työntekijä- ja esimies tai projektipäällikkö-roolien toiminnallisuuksien lisäksi ylläpitäjä voi luoda, poistaa ja muuttaa käyttäjiä, käyttäjärooleja, vaiheita ja tehtäviä. Tässä luvussa esitellään käyttäjän (kuva 14), käyttäjäroolin (kuva 15), tehtävän (kuva 16) ja vaiheen (kuva 17) luonti. Muuttaminen ja poistaminen ovat samantyyppisiä näkymiä kuin esimerkiksi projektin näkymä (kuva 12,13), joten niitä ei tässä yhteydessä erikseen esitellä.

Not logged in | Sign out

# TuntiMaku



**UserRoles** + Create UserRole 1-2 of 2

Persons	Phases	Projects	Tasks	Workhourss	Id	Version	Name
					1	0	ROLE_ADMIN
					2	0	ROLE_USER


**New UserRole**

**Name:**

KUVA 14. Käyttäjäroolin lisääminen

Not logged in | [Sign out](#)

# TuntiMaku



- UserRoles
- Persons
- Phases
- Projects
- Tasks
- Workhours

+ Create Person 1-1 of 1

Id	Version	Username	Firstname	Lastname	Password	Active	User Role
1	1	laros	Laura	Rosenberg	laros123	true	ROLE_ADM

**New Person**

Username:

Firstname:

Lastname:

Password:

Active:


User Role:

ROLE\_ADMIN (1)  
 ROLE\_USER (2)

KUVA 15. Käyttäjän lisääminen

Not logged in | [Sign out](#)

# TuntiMaku



- UserRoles
- Persons
- Phases
- Projects
- Tasks
- Workhours

+ Create Task 1-2 of 2

Id	Version	Code	Name	Description
1	0	222	Suunnittelu	töiden suunnittelu
2	0	223	Toteutus	töiden toteutus

**New Task**

Code:


Name:

Description:

KUVA 16. Tehtävän lisääminen

Not logged in | [Sign out](#)

# TuntiMaku



- UserRoles
- Persons
- Phases
- Projects
- Tasks
- Workhourss

+ Create Phase 1-2 of 2

Id	Version	Code	Name	Description
1	0	444	Aloitus	työn vaihe
2	0	445	Dokumentointi	toinen vaihe

**New Phase**

Code:

Name:

Description:

KUVA 17. Vaiheen lisääminen

### 3.7.6 Lisäominaisuudet

Myöhemmässä vaiheessa tuntikirjausprojektiin toteutetaan mobiilirajapinta ja toiminnallisuus tilitoimiston tarpeisiin. Mobiilirajapinta saatiin toteutettua Spring Roo -projektiin (Toivanen 2012.), jota sitten hyödynnettiin myös Windows Phone -sovelluskehityksessä. Tilitoimistoa varten tehdään oma käyttäjärooli, jonka avulla voi luoda raportit työntekijöiden kirjauksista palkanmaksun suorittamiseksi. Myös kalenterinäkyvät tuntikirjaustoiminnallisuuteen lisätään. Ne eivät ole kirjauksen kannalta välttämättömät, mutta helpottavat käyttäjän tuntien seuraamista ja parantavat käytettävyyttä.

## 4 MOBIILISOVELLUS TUNTIKIRJAUS-OHJELMISTOON

### 4.2 Sovellusalustan valinta

Sovellusalustan valintaan vaikuttavia tekijöitä on useita. Valinnassa keskityttiin vuoden 2012 alussa suosituimpina olleisiin alustoihin, joita olivat Android, iPhone ja tuolloin Windows Phone 7 (WP7). Lisäksi tuli miettiä, mitkä alustat ja niiden tuoma osaaminen olisivat yrityksen kannalta kaikkein hyödyllisimmät. Sovelluslustoja valittiin kaksi, jotka toteutetaan tuntikirjausohjelmistoon päättötyön yhteydessä. Toteutus päätettiin tehdä kahdelle alustalle, Androidille ja WP7:lle. Nämä kaksi alustaa liittyivät myös opintoihin, sillä niitä oli yhdellä ohjelmistoarkkitehtuuri -kurssilla vähän päästy kokeilemaan. Yrityksellä on iPhone-osaamista ennestään, joten katsottiin, että toteutus pystytään tekemään iPhonelle myöhemminkin.

WP7 on Microsoftin kehittämä mobiilikäyttöjärjestelmä matkapuhelimiin. Se tuli markkinoille uusien puhelinmallien myötä vuoden 2010 loppupuolella. Windows Phone 7 -puhelimia valmistavat muun muassa HTC, Samsung ja Nokia. Käytettävä kehitysympäristö on Microsoft Visual Studio Express for Windows Phone. Se on suppeampi versio Visual Studio 2010 -kehitysympäristöstä, mutta sisältää kaikki toiminnallisuudet, joita WP7-sovelluskehityksessä tarvitaan. WP7:lle voi toteuttaa ohjelmia C#-ohjelmointikielen ja .NET-kehityksen avulla.

### 4.1 Suunnittelun vaatimukset ja määrittely

Mobiilirajapinta perustuu JSON (JavaScript object notation) -kirjastoihin. JSON on yksinkertainen ja tehokas tiedonsiirtomuoto palvelimen ja mobiilirajapinnan välillä pienille datamäärille. Mobiilirajapinta autentikoi käyttäjän jokaisen pyynnön yhteydessä eikä siten käytä tavallista Spring Security -mallia. Mobiilirajapintaa on tarkemmin kuvattu Toivasen työssä. (Toivanen 2012.)

Koska mobiililaitteen käyttöliittymän koko on rajallinen, on sovelluksesta tehtävä helpokäyttöinen ja selkeä. Mahdollisimman paljon listoja ja valmiiksi syötettyä tietoa olisi hyvä käyttää. Jos internetyhteyttä ei ole käytettävissä tai sen käyttäminen on suhteel-

lisen kallista esimerkiksi ulkomailla, pitää kirjaukset pystyä tallentamaan puhelimen omaan tietokantaan.

Mobiilisovelluksesta toteutetaan vain tuntikirjaukseen tarvittavat ominaisuudet. Näytöön haetaan käyttäjälle merkityt projektit, vaiheet ja tehtävät, jotta kirjaus voidaan suorittaa. Käyttäjärooleista vain työntekijällä on oikeus kirjata tunteja mobiilisovelluksessa. Korkeamman käyttäjäroolin vaativat toimenpiteet, kuten projektien ja uusien työntekijöiden luonti, suoritetaan web-sovelluksen kautta.

Suunnittelussa huomioitavia asioita ovat mobiililaitteen pieni koko, rajoitettu muistin määrä ja vähäinen energiankulutus. Käyttöliittymässä asiat on pyrittävä esittämään yksinkertaisesti ja selkeästi.

Palvelinsovellus tarjoaa mobiilisovellukselle kolme rajapintaa:

- käyttäjän tunnistus ja yhteyden sulkeminen
- raportointirajapinta, johon liittyvät projektit, tehtävät, vaiheet ja kirjaukset
- tuntikirjaus (uusi tai muutettu).

Käyttäjän tunnistuksessa mobiilisovellus lähettää palvelimelle käyttäjätunnuksen sekä tiivisteiden salasanan. Palvelin vastaa joko virheellä tai käyttäjän nimellä ja yhteyden tunnisteella. Yhteyden sulkemiseen asti tunniste yksilöi käyttäjän ja yhteyden oikeellisuuden.

Raportti-rajapinta sisältää käyttäjälle määräytyvät projektit ja tehtävät sekä vaiheet uutta kirjausta varten tai tuntikirjaukset halutulta ajalta. Palvelin hakee tiedot tietokannasta ja vastaa joko JSON- tai XML-muotoisena palautteena. Uusi tai muutettu tuntikirjaus lähetetään palvelimelle joko parametreina tai JSON- tai XML-muotoisena. Palvelin tarkastaa syötettävien tietojen oikeellisuuden, tallentaa ne tietokantaan ja vastaa virhekoodilla.

Vain työntekijän käyttäjäroolilla kirjautuvat käyttäjät voivat käyttää sovellusta. Projektien ja työntekijöiden lisäykset ja muutokset tapahtuvat palvelimella pyörivän sovelluksen kautta.

#### 4.4 Toteutus

Mobiilisovelluksen suunnittelu saatiin tehtyä tämän projektin aikana. Toteutus päätettiin jättää kevyemmäksi ja siirtää sitä myöhemmäksi, sillä keskityttiin enemmän palvelimella toimivan web-sovelluksen toiminnallisuuteen (luku 3). Tällä hetkellä yrityksellä on käytössä Android-puhelimia, joten tärkeämpää oli saada tuntikirjaussovellus toimimaan ensisijaisesti niillä. Android-toteutuksesta voi lukea Toivasen työstä. (Toivanen 2012.)

Mobiilisovelluksen toteutus tehdään Microsoft Visual Studio 2010 Express for Windows Phone-kehitysympäristöllä, johon luodaan uusi Windows Phone Application.

##### 4.4.1 Kirjautuminen

Kirjautumisnäyttöön (kuva 18) tehdään kentät käyttäjätunnukselle ja salasanalle, sekä kirjautumis-painike. Myöhemmässä vaiheessa tähän näyttöön on suunniteltu toteutettavaksi myös asetus mahdollisuus palvelun url-osoitteelle, koska se on pystyttävä siihen vaihtamaan.



KUVA 18. Kirjautuminen

Kirjautumisessa käytetään samaa käyttäjätunnusta ja salasanaa, jotka on palvelinsovelluksella käyttäjälle annettu. Kirjautumiseen tarvitaan työntekijä tason oikeudet. Salasanan tiivisteet lasketaan käyttämällä SHA256Managed luokkaa (System.security.cryptography). (Microsoft 2012a.) Kirjautumisen yhteydessä haetaan käyttäjän tietojen lisäksi käyttäjän projektit, vaiheet ja tehtävät palvelimella sijaitsevalta tuntikirjaussovellukselta. Tämä toteutetaan HttpRequest- ja HttpResponse-rajapintojen avulla.

```
private void kirjautu_button_Click(object sender, RoutedEventArgs e)
{
    SHA256Managed sha = new SHA256Managed();

    byte[] hash = sha.ComputeHash((System.Text.Encoding.UTF8.GetBytes(salasanatextBox.Password)));
    passwordhash = ByteArrayToString(hash);
    username = tunnustextBox.Text;

    HttpResponseMessage content = doRequest("http://localhost:8080/TuntiMaKu/mobileapi/", "me", "GET");
    content = doRequest("http://localhost:8080/TuntiMaKu/mobileapi/", "projects", "GET");
    content = doRequest("http://localhost:8080/TuntiMaKu/mobileapi/", "tasks", "GET");
    content = doRequest("http://localhost:8080/TuntiMaKu/mobileapi/", "phases", "GET");

}
```

JSON-muunnokseen käytetäänDataContractJsonSerializer-kirjastoa. (Microsoft 2012b.) Jotta JSON-muunnos olisi mahdollista, täytyy muunnettava luokka merkitä DataContract-attribuutilla. Lisäksi jokainen luokan jäsen merkitään DataMember-attribuutilla.

```
namespace MobiiliTuntiMaKu
{
    [DataContract]
    public class Worker
    {
        [DataMember]
        public long id { get; set; }
        [DataMember]
        public int version { get; set; }
        [DataMember]
        public String username { get; set; }
        [DataMember]
        public String firstname { get; set; }
        [DataMember]
        public String lastname { get; set; }
        [DataMember]
        public String password { get; set; }
        [DataMember]
        public String passwordagain { get; set; }
        [DataMember]
        public Boolean active { get; set; }
        [DataMember]
        public List<Workerrole> workerrole { get; set; }
        [DataMember]
        public Boolean overtimeAllowed { get; set; }
        [DataMember]
        public List<Project> projects { get; set; }

        public override String ToString()
        {
            return this.firstname + " " + this.lastname;
        }
    }
}
```

Datan muuntaminen JSON-muotoiseksi tapahtuu DataContractJsonSerializer WriteObject()-metodin avulla.



```

public static string Serialize<T>(T obj)
{
    using (var ms = new MemoryStream())
    {
        var serializer = new DataContractJsonSerializer(typeof(T));
        serializer.WriteObject(ms, obj);
        var json = ms.ToArray();

        String jsonString = Encoding.UTF8.GetString(json, 0, json.Length);
        return jsonString;
    }
}

```

JSON-datan takaisin muuntamiseen käytetään DataContractJsonSerializer ReadObject()-metodia.

```

public static T Deserialize<T>(string json)
{
    String[] reps = { "\"startdate\"", "\"enddate\"", "\"approveddate\"" };
    using (var ms = new MemoryStream(Encoding.UTF8.GetBytes(json)))
    {
        var serializer = new DataContractJsonSerializer(typeof(T));
        return (T)serializer.ReadObject(ms);
    }
}

```

#### 4.4.2 Päänäyttö

Päänäyttö on yksinkertainen näyttö, johon kirjautumissivulta siirrytään. Siirryttäessä näytölle, välitetään tieto kirjautuneen käyttäjän käyttäjätunnuksesta sille varattuun kenttään.

```

NavigationService.Navigate(new Uri("/MainPage.xaml?tunnus=" + tunnustextBox.Text + "&me" + results, UriKind.Relative));

```

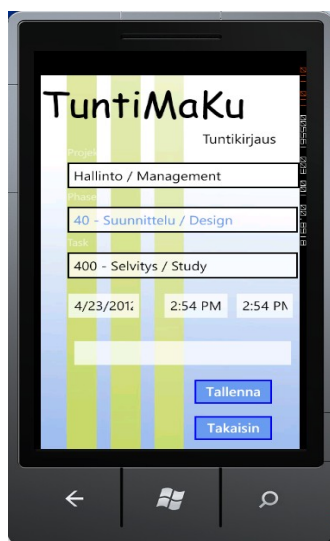
Päänäytöllä (kuva 19) näytetään tuntikirjaussovelluksen toimintoihin liittyvät painikkeet.



KUVA 19. Päänäyttö

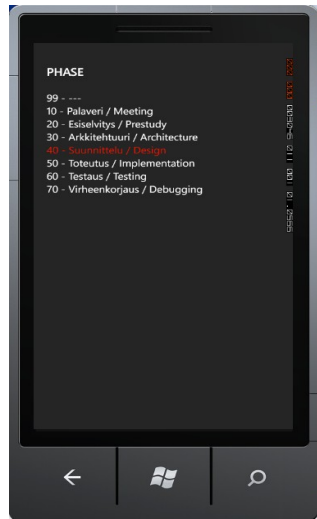
### 4.4.3 Tuntikirjaus

Tuntikirjausnäytössä (kuva 20) tapahtuu varsinaisen tuntien kirjaaminen. Kentissä näytetään palvelimella sijaitsevalta tuntikirjaus sovelluksesta haetut käyttäjän projektit, vaiheet ja tehtävät. Käyttäjä pystyy vaihtamaan niistä kullekin tuntikirjaukselle sopivat tiedot, sekä lisäämään päivämäärän ja kelloajat. Tiedot tallennetaan tallenna-painikkeella.



KUVA 20. Tuntikirjausnäyttö

Kun halutaan valita esimerkiksi vaiheet tuntikirjaukseen, avautuu kuvassa 21 oleva lista valittavista tiedoista. Valitsemalla yhden vaiheen, tieto siirtyy tuntikirjausnäytön laatikkoon vaiheeksi ja tämä lista sulkeutuu. Listojen näyttämiseen käytetty komponentti on ListPicker.



KUVA 21. Vaiheet lista näytöllä

Päivämäärää ja aikaa muutettaessa avautuu seuraavanlaiset näkymät (kuva 22), josta tiedot voi asettaa.



KUVA 22. Päivämäärän ja kellonajan asetusnäytöt

Käytetyt komponentit ovat DatePicker ja TimePicker, jotka tarjoavat tekstikentän, josta käyttäjä pääsee asettamaan tiedot. Molemmat komponentit periytyvät DateTimePickerBase-luokasta ja niillä on sama rajapinta. Erona näillä komponenteilla on se, että DatePicker näyttää päivämäärän ja TimePicker kellonajan. (WindowsPhone Geek 2010.)

Tuntikirjaus tietojen tallennus tapahtuu HttpRequest POST-metodilla, silloin kun internetyhteys on käytettävissä. Lähetys tapahtuu kolmessa osassa. BeginGetRequestStream-metodi aloittaa asynkronisen pyynnön ja lähettää tietoja HttpRequest-metodille. BeginGetResponse-metodi aloittaa asynkronisen kyselyn palvelimelta tulevasta vastauksesta. GetResponseStream-metodi lukee palvelimelta tulevan vastauksen. (Microsoft 2012c.)

Käyttäjän salasanaa ei lähetetä palvelimelle missään muodossa, vaan siitä lasketaan muiden kenttien kanssa tiiviste (hash), jota palvelin vertaa omaan laskemaansa jokaisella yhteydellä.

Uloskirjausta ei ole, mutta headerissa siirtyvistä kentistä timestamp on noin 10 minuuttia, joka määräytyy palvelinsovelluksen asetuksista.

Myöhemmässä vaiheessa käyttäjän kirjausten tallennukseen voidaan käyttää myös puhelimen omaa tietokantaa, silloin kun internetyhteyttä ei ole tai sitä ei haluta käyttää. Isolated Storage on datan tallennukseen tarkoitettu rajapinta WP7-laitteissa. Käyttäjän tietojen tallennukseen voisi käyttää IsolatedStorageSettings-metodeja ja kirjausten tallentamiseen IsolatedStorageFile-metodeja.

## 5 YHTEENVETO

Projekti on ollut pitkä ja opettavainen, toisaalta tuotekehitysnäkökulmasta katsottuna se on vasta alussa. Työn tavoitteena oli saada suunniteltua ja toteutettua tuntikirjausohjelmisto ja mobiilisovellus yrityksen käyttöön. Suunnittelu saatiin tehtyä tuntikirjaustoiminnallisuuden osalta valmiiksi ja toteutuksesta saatiin aikaan hyvä pohja jatkokehitystä varten.

Kun projekti syksyllä käynnistettiin, oli ajatuksena edetä vaiheittain määrittelystä suunnitteluun, toteutukseen ja testaukseen. Tehtiin projektisuunnitelma ja laadittiin aikataulut ja yritettiin myös pitää niistä kiinni. Syksy oli ajateltu käytettäväksi määrittelyyn ja suunnitteluun sekä uuden työkalun, Spring Roon, opiskeluun. Kokemusta web-sovelluksista, Javasta ja WP7:sta ei juurikaan ollut, joitakin harjoitustöitä oli tullut tehtyä aikaisemmin, joten tiedossa oli runsaasti opiskelua. Projektityöskentelystä on sen sijaan usean vuoden kokemus. Vuodenvaihteen lähestyessä suunnitelmat olivat valmiit ja vaaditut määrittelydokumentit kirjoitettu. Uutta Spring Roo -ohjelmointiympäristöä oli testattu sen verran, että tiedettiin, mitä sillä pystyy tekemään.

Tuntikirjaussovellusta ryhdytään toteuttamaan Spring Roon ja GWT:n yhdistelmällä. Alkuun kaikki näytti lupaavalle, mutta sitä mukaa kun uusia tuntikirjausohjelmiston ominaisuuksia otettiin käyttöön, ilmeni saman verran ongelmia. GWT:n ja Spring Roon ohjelmaversiot eivät olleet kaikilta osin yhteensopivia. Spring Security toi projektiin lisää ongelmia ratkaistavaksi. Internetistä löytyi paljon kirjoituksia siitä, mitä kannattaisi missäkin kohtaa kokeilla, mutta mikään ei tuntunut johtavan haluttuun tulokseen.

Teknisten ongelmien vuoksi menivätkin aikataulut uusiksi. Huhtikuun loppuun mennessä yritettiin tehdä se, mitä suinkin pystyttiin. Olin jo hylkäämässä projektia moneen kertaan, mutta jostain löytyi voimaa, kekseliäisyyttä ja tukea jatkaa eteenpäin. Pala palalta Spring Roon ja GWT:n yhteensovittaminen alkoi onnistua ja ongelmia saatiin ratkaistua. Suurin ongelma oli yrittää saada kaikki Spring Roo -projektissa (Toivanen 2012.) toteutetut toiminnallisuudet toimimaan tässä projektissa ja samalla ymmärtämään samojen asioiden erilainen toiminta näiden kahden ympäristön välillä.

Keväällä todettiin, että ihan kaikkia ongelmia ei saatu ratkaistua päättötyöprojektin aikana. Näistä yksi vielä selvitettäväksi jäävä asia on käyttäjien näkymien hallinta. Järjestelmään pääsee kirjautumaan eri käyttäjärooleilla ja näille on omat näkymät, mutta niitä ei ole saatu toimimaan yhteen halutulla tavalla. Toinen selvitettävä asia on kieli-versioiden i18n (Internationalization) tekeminen. Spring Roo -projektissa nämä molemmat toimivat, mutta se miten GWT:llä saadaan asiat toimimaan, vaatii vielä selvitystyötä. Ratkaisu voi olla yksinkertainen, mutta pahimmillaan viikkojen työ. Kolmas asia, joka jäi hieman kesken, on viikon aikana kirjattujen tuntien kalenterinäkömää. Sen periaate on selvillä, mutta ominaisuuden lisäämistä tuntikirjaussovellukseen päätettiin siirtää myöhäisempään vaiheeseen.

Mobiilisovelluksen suunnittelu saatiin tehtyä projektin aikana. Toteutusta ei ennätetty kovin pitkälle viedä, mutta karkea perustoteutus siitä valmistui.

Eniten vaikeuksia tuotti aikataulu. Vaikka projekti aloitettiin jo hyvissä ajoin, ei aika silti riittänyt aivan niin hyvin, kuin oli suunniteltu. Kaikkiin uuden ympäristön tuomiin ongelmiin ei osattu etukäteen varautua. Opinnäytetyön, työn ja perheen yhteensovittaminen oli haasteellista. Työtunteja kertyi paljon, koska teknisiä ongelmia ympäristön kanssa oli runsaasti. Vaikka rankka vuosi on takanapäin, on opittu paljon ja projekti kokonaisuudessaan meni ihan hyvin. Jos nyt täytyisi valita tuntikirjaussovelluksen alusta uudelleen, ei välttämättä valittaisi toisin.

Vielä ei ole päätetty, kummalla toteutuksella tehty (Spring Roo vai Spring Roo + GWT) tuntikirjausohjelmisto otetaan yrityksessä koekäyttöön kesän 2012 aikana. Lisää ominaisuuksia ja entisten ominaisuuksien parantelua tehdään sitä mukaa, kun aikaa muilta töiltä jää. Aika näyttää sitten, mihin se johtaakaan.

## LÄHTEET

Alex, B. Schmidt, S. Stewart, A. Tyrrell, J. Swan, A. 2011. *Spring Roo- Reference Documentation*. [verkkodokumentti] Spring Framework [viitattu 23.5.2012]. Saatavissa <http://static.springsource.org/spring-roo/reference/html/architecture.html#architecture-critical-technologies-aspectj>

Ebsolut Oy 2012. yrityksen www-sivu [viitattu 18.5.2012]. Saatavissa <http://www.ebsolut.fi/index.asp?pid=123>

Genuitec 2010. *Posts Tagged 'Web Development'* www-sivu [viitattu 21.5.2012] Saatavissa <http://www.genuitec.com/blog/?tag=web-development>

Google Web Toolkit 2012a. *Making Remote Procedure Calls* www-sivu [viitattu 10.5.2012]. Saatavissa: <https://developers.google.com/web-toolkit/doc/latest/tutorial/RPC>

Google Web Toolkit 2012b. *Productivity for developers, performance for users* [viitattu 10.5.2012] Saatavissa <https://developers.google.com/web-toolkit/>

Kujala, T. 2003. *Tietokantasuunnittelusta* [verkkodokumentti] Helsinki [viitattu 6.5.2012] Saatavissa <https://www.cs.helsinki.fi/u/tkujala/tikapeS2003/kalvot/pdf/txsu.PDF>

Logica 2012. yrityksen www-sivu [viitattu 18.5.2012]. Saatavissa <http://www.logica.fi/we-do/mobile-services/reporting/>

Machacek, J., Vukotic, A., Chakraborty, A., Ditt, J. 2008. *Pro Spring 2.5*. Apress.

Microsoft 2012a. MSDN Library SHA256Managed Class. [viitattu 10.5.2012]. Saatavissa <http://msdn.microsoft.com/en-us/library/system.security.cryptography.sha256-managed.aspx>

Microsoft 2012b. *How to:Serialize and Deserialize JSON Data*. [verkkodokumentti] MSDN Library [viitattu 16.5.2012]. Saatavissa <http://msdn.microsoft.com/en-us/library/bb412179.aspx>

Microsoft 2012c. *HttpWebResponse Methods*. [verkkodokumentti] MSDN Library [viitattu 7.5.2012] Saatavissa <http://msdn.microsoft.com/en-us/library/0h6wsc8k>

Mularien, P. 2010. *Spring Security 3*. UK. Packt Publishing Ltd.

Net Weight Oy 2012. yrityksen www-sivu [viitattu 18.5.2012]. Saatavissa <http://www.netweight.fi/tuntikirja/>

Noyra Oy 2012. yrityksen www-sivu [viitattu 18.5.2012] Saatavissa <http://www.simoapp.com/>

Ramsdale, C. 2010. *Large scale application development and MVP* www-sivu [viitattu 16.5.2012]. Saatavissa <https://developers.google.com/web-toolkit/articles/mvp-architecture?hl=fi-FI>

Sarin, A. 2011. *SpringRoo 1.1 Cookbook*. UK. Packt Publishing Ltd.

Toivanen, K. 2012 *Verkkopalvelualustat, Tuntikirjaus verkkopalveluna ja Android-sovelluksena* YAMK opinnäytetyö Savonia ammattikorkeakoulu, Tekniikka. Kuopio

ValueFrame 2012. yrityksen www-sivu [viitattu 18.5.2012]. Saatavissa <http://www.valueframe.com/>

Visma Avendo Oy 2011. yrityksen www-sivu [viitattu 18.5.2012]. Saatavissa <http://www.vismaavendo.fi/helposti-alkuun/tyoajankirjaus/>

Wikipedia 2012. *Toiminnanohjausjärjestelmä* [viitattu 22.5. 2012]. Saatavissa: <http://fi.wikipedia.org/wiki/Toiminnanohjausj%C3%A4rjestelm%C3%A4>

WindowsPhone Geek 2010. *WP7 DatePicker and TimePicker in dept | API and customization*. www-sivu [viitattu 7.5.2012]. Saatavissa <http://windowsphonegeek.com/articles/wp7-datepicker-and-timepicker-in-depth--api-and-customization>





