

## Tehtävienhallinta

Tuomo Pohjola

Opinnäytetyö

Tietotekniikan koulutusohjelma

14.3.2013



<b>Tekijä tai tekijät</b> Tuomo Pohjola	<b>Ryhmätunnus tai aloitusvuosi</b> HETI09
<b>Raportin nimi</b> Tehtävienhallinta	<b>Sivu- ja liitesivumäärä</b> 19 + 6
<b>Opettajat tai ohjaajat</b> Jukka Juslin	
<p>Tehtävienhallinta projektin toimeksiantajana toimii IMS Business Solutions Oy.</p> <p>IMS Business Solutions Oy kehittää ja myy omaa selainpohjaista IMS toimintajärjestelmää. Toimintajärjestelmän avulla asiakkaat voivat hallita kaikkea olemassa olevaa materiaaliaan tai luoda uutta IMS ohjelman avulla.</p> <p>Opinnäytetyön aikana toteutettiin tehtävienhallinta prototyyppi, määrittäydokumentti, testausdokumentti ja jatkokehityslistaus. Tekniikoina käytetään Javaa, Spring Frameworkkia ja JPA:ta.</p> <p>Tehtävienhallinnalla voidaan hallita yritysten henkilöstön päivittäisiä tehtäviä.</p> <p>Opinnäytetyö sisältää liitteinä salaisiksi luokiteltuja dokumentteja, joita ei voi tämän vuoksi näyttää opinnäytetyössä.</p>	
<b>Asiasanat</b> Spring, JPA, jQuery, Toimintajärjestelmä	

Degree programme

<p><b>Authors</b></p> <p>Tuomo Pohjola</p>	<p><b>Group or year of entry</b></p> <p>HETI09</p>
<p><b>The title of thesis</b></p> <p>Task management</p>	<p><b>Number of pages and appendices</b></p> <p>19 + 6</p>
<p><b>Supervisor(s)</b></p> <p>Jukka Juslin</p>	
<p>Task management project is done for IMS Business Solutions Oy.</p> <p>Thesis contains confidential company documents that can't be shown in thesis.</p> <p>IMS Business Solutions Oy develops and sells browser based IMS quality management system. With the system, customers can manage all their existing documents or create new using IMS.</p> <p>Task management prototype, configuration document, test document and list for future features were done during thesis.</p> <p>Technologies that is used in prototype is Java, Spring, JPA.</p>	
<p><b>Key words</b></p> <p>Spring, JPA, jQuery, Quality management system</p>	

# Sisällys

1. Johdanto .....	1
1.1. Tehtävienhallinnan jaottelu .....	2
1.2. Tehtävienhallinta osio .....	2
1.3. Tehtävienhallinta moduuli.....	3
1.4. Automaatio.....	3
1.5. Kommentointi .....	3
2. Teoriatausta.....	4
Java.....	4
Maven .....	4
Spring Framework.....	5
Hibernate.....	5
JPA.....	5
JavaScript.....	6
jQuery .....	6
Json .....	6
Ajax .....	7
Mustache .....	7
Handlebarsjs.....	7
Css .....	8
Git .....	8
Cacoo.....	8
YouTrack.....	9
3. Projekti.....	10
3.1. Tavoite .....	10
3.2. Laadullinen tavoite .....	10
3.3. Projektin kustannukset .....	11
3.4. Resurssit.....	11
3.5. Riskit.....	11
3.5.1. Rajaus .....	11
3.5.2. Tekniikka .....	11

3.5.3.	Työaika.....	12
3.6.	Ajoitus.....	12
3.7.	Projektin kulku.....	12
3.7.1.	Aloitus.....	12
3.7.2.	Välivaihe .....	13
3.7.3.	Päätösvaihe.....	13
4.	Lopputulokset.....	14
4.1.	Liiketoiminta saavutukset.....	14
4.2.	Omat saavutukset ja oppiminen .....	15
5.	Jatkokehitys .....	16
	Lähteet.....	17
	Learning JavaScript Design Patterns, Addy Osmani.....	17
	Liitteet.....	19
	Salaiset liitteet .....	19

# 1. Johdanto

Tehtävienhallinta prototyyppi on tehty omana kokonaisuutena IMS-toimintajärjestelmään

HAAGA-HELIA Ammattikorkeakoulun tietojenkäsittelyn opinnäytetyönä 1.11.2012 – 14.3.2013. Projektin toimeksiantajana toimii IMS Business Solutions Oy.

IMS Business Solutions Oy on laadunhallintaan erikoistunut yritys, jolla on useiden vuosien laadunhallinta kokemus toimintajärjestelmäkoulutusten ja kehitysprojektien kautta.

IMS Business Solutions Oy kehittää ja myy omaa selainpohjaista IMS toimintajärjestelmää.

Toimintajärjestelmän avulla yritykset voivat hallita kaikkea olemassa olevaa materiaaliaan

tai luoda uutta IMS ohjelman avulla.

Tehtävienhallinta on työkalu jolla voidaan hallita erilaisia työtehtäviä, joita henkilöiden tulee tehdä päivän, viikon, kuukauden tai jopa vuoden aikana riippuen kuinka isoa kokonaisuutta halutaan hallita.

Tehtävienhallinnalla voidaan luoda tehtäviä liittyen IMS-toimintajärjestelmään tai tehtävät voivat olla täysin vapaamuotoisia. Tehtävienhallinnan on tarkoituksena helpottaa

ja nopeuttaa asiakkaiden päivittäisten työtehtävien hallinnoimista.

Projektin tuloksena syntyy prototyyppi, sisältäen tehtävien lisäämisen, poistamisen, muokkauksen ja listaamisen erillisessä tehtävienhallinta osiossa. Osion lisäksi toteutetaan

erillinen tehtävienhallinta moduuli, jonka avulla tehtävienhallintaa voidaan nopeuttaa

päivittäisessä käytössä, sekä automaatio toiminto. Moduuli tulee tehdä niin, että sen pystyy liittämään IMS toimintajärjestelmän jokaiseen osa-alueeseen.

Opinnäytetyön painoarvo on asetettu prototyypin toteutukseen ja tämän määrittäisiin, testaukseen, sekä kehitysehdotuksiin.

### **1.1. Tehtävienhallinnan jaottelu**

Opinnäytetyössä tehty tehtävienhallinta on jaettu neljään eri kokonaisuuteen. Tehtävienhallinta osio, jossa hallinnoidaan, muokataan ja listataan tehtäviä.

Tehtävienhallinta moduuli, joka sisältää nopeaan käyttöä tarvittavat toiminnot ja listauksen, kaikkialla johon se on liitetty.

Automaatio, jonka avulla tehtäviä voidaan hallita automaattisesti ja käyttäjät saavat näin tehokkaamman kokonaisuuden.

Sekä kommentointi, jolla käyttäjät voivat antaa palautetta tehtäville. Kommentointi on oma kokonaisuus jota ei prototyypin aikana toteuteta loppuun asti.

### **1.2. Tehtävienhallinta osio**

Tehtävienhallinta osio on kokonaisuus, jossa käyttäjät voivat listata, luoda, muokata ja kommentoida tehtäviä.

### **1.3. Tehtävienhallinta moduuli**

Tehtävienhallinta moduulilla tarkoitetaan ominaisuutta, jonka on tarkoituksena näkyä kaikissa IMS- toimintajärjestelmän osioissa.

Toiminto pystytään tarvittaessa liittämään myös IMS toimintajärjestelmän ulkopuolelle.

Moduuli sisältää listauksen lisäksi pikatoimintoja tehtävien hallinnoimiseen, kuten tehtävän varaamisen ja tehtävien merkkäminen tehdyksi.

### **1.4. Automaatio**

Automaatio ominaisuuden ansiosta, tehtäviä voidaan halutessa luoda tai poistaa eri tapahtumien yhteydessä. Tämän avulla käyttäjille saadaan viimeistelty kokonaisuus, jossa tehtävät hoituvat ilman erillistä työtä.

Ylläpitotoimintojen avulla voidaan automaattisia tehtäviä asettaa toiminta kohtaisesti.

Opinnäytetyötä varten on toteutettu IMS dokumentin hyväksyntäpyynnöstä luotava automaattinen tehtävä. Kun dokumentista lähetetään hyväksyntäpyyntö luo järjestelmä uuden tehtävän valituille käyttäjille. Dokumentin hyväksynnän jälkeen tehtävä poistetaan automaattisesti henkilöiltä, joille tehtävä luotiin.

### **1.5. Kommentointi**

Kommentointi ominaisuus on yleisesti käytettävä toiminto, joka on tehty omaksi kokonaisuudeksi tehtävienhallinnan ohelle.

Tämän toteutuksen avulla kommentointia pystytään käyttämään myöhemmin myös muissa IMS kokonaisuuksissa.

Tehtävienhallintaa varten toteutettiin kommentoinnin listaus, lisäys ja poisto toiminnot.



## 2. Teoriatausta

Projektissa hyödynnetään useita eri tekniikoita. Pääosin sovellus on toteutettu käyttäen Javaa, Spring Frameworkkia ja jQueryä.

Projektin tietokantataulujen rakentamiseen on hyödynnetty Jpa:n annotaatioita. Kantarakenteessa on pyritty ottamaan huomioon ohjelmistoon tulevat laajat muutokset.

### Java

Opinnäytetyön prototyyppi on pääosin toteutettu Javalla, koska IMS toimintajärjestelmä on pohjautuu Java kieleeseen. Java on olio-ohjelmointiin perustuva ohjelmointikieli.

(Manning The Well Grounded Java Developer foreword & preface)

### Maven

Maven auttaa projektissa tarvittavien kirjastojen hallinnoimisessa. Mavenin automaatio pitää huolen, että projektissa on aina oikeat ja ajan tasalla olevat kirjastot, kun projekti käännetään uudestaan. Maven tukee lähes kaikkia suosittuja kehitysympäristöjä, kuten Eclipse, NetBeans ja IntelliJ IDEA. Riippuvuudet tulee kirjoittaa oikeaan muotoon, jotta Maven osaa löytää tarvittavat kirjastot.

(Manning The Well Grounded Java Developer appendix E)

## **Spring Framework**

Spring Framework on avoimeen lähdekoodiin perustuva kevyt, annotaatioihin suuntautunut Java kehys. Spring Framework suurimpana hyötynä on, että se on suunniteltu helpottamaan ja nopeuttamaan Java EE perustuvien sovelluksien tekoa. Spring käyttää Java luokissaan erilaisia annotaatioita, joilla voidaan toteuttaa esimerkiksi Controller luokka. (Spring in action ja Spring in action 3 painos)

## **Hibernate**

Hibernate on ORM tekniikkaan pohjautuva kokoelma eri kirjastoja, joiden avulla kehittäjät voivat hyödyntää Java luokkia tietokantakyselyissä. Se ei tarjoa ainoastaan objektipohjaista tietokanta arkkitehtuuria, vaan myös erittäin hyödyllisiä tekniikoita kuten lazy loading, caching ja eager fetching. Edellä mainituilla tekniikoilla voidaan hakea tietoa kannasta eri tavoilla. (Java Persistence With Hibernate ja Spring in action 3 painos)

Ohjelmistojen koon kasvaessa, muuttuvat tietokantakyselyt yleensä monimutkaisemmaksi. Hibernaten tarkoitus on auttaa kehittäjää näissä kyselyissä. Edellä mainittu Spring tarjoaa omalta osaltaan tuen monelle tietokanarkkitehtuurille, kuten Hibernate tai seuraavassa kappaleessa esiintyvä JPA. (Spring in action 3 painos)

## **JPA**

Jpa on seuraavan sukupolven Java standardi, joka on kehitetty olio keskittyneistä ja relaatiotietokantamallisia järjestelmiä varten. JPA tuo mukanaan joukon toiminnallisuuksia Hibernaten natiivista kirjastosta, sekä Java Data Objectista ja yhdistää nämä keskenään.

Jpa:ta suositellaan käyttämään Spring pohjaisissa sovelluksissa niin pitkälle kuin mahdollista, jos JPA:n avulla ei pystytä toteuttamaan jotain kokonaisuutta, pystytään tämä yleensä aina toteuttamaan tarvittaessa Hibernaten toiminnallisuuksien avulla.

Jpa käyttää tavallisia Java-luokkia tietokanta taulujen luomiseen, annotaatioiden avulla voidaan kertoa, että luokka on esimerkiksi taulu tai että taulun sarake pitää olla kokonaisluku tyyppiä.

(Java Persistence With Hibernate 2012, Spring in action 3 ja Pro JPA 2: Mastering the Java™ Persistence API Expert's Voice in Java Technology)

## **JavaScript**

JavaScript on kevyt, mutta erittäin tehokas ohjelmointikieli, jonka tarkoituksena on lisätä vuorovaikutteisuutta sivustoilla. JavaScript käännetään käyttäjän koneella.

JavaScriptillä voidaan muokata reaaliaikaisesti sivuston Css tyylejä tai tavallisia Html elementtejä.

(Learning Web Design 4th Edition s. 460 – 462)

## **jQuery**

Avoimeen lähdekoodiin perustuva ilmainen kirjasto, joka perustuu JavaScriptin eri kirjastoihin. JQuery onkin yksi suosituimmista JavaScript kirjastoista. jQuery lisää sivun käytettävyyttä ja näytettävyyttä tuomalla mukanaan hienoja toiminnallisuuksia.

JQuery käyttäminen on JavaScriptiin verrattuna lyhyempää ja yksinkertaisempaa.

(Learning Web Design 4th Edition s. 498 – 499)

## **Json**

Json on erittäin kevyt teksti pohjainen formaatti, jota käytetään useimmiten jQuery Ajax pyyntöjen ohella. Formaatti on suunniteltu niin, että ihmisen ja koneen on helppoa lukea sitä. Json objekti sisältää listan järjestelemättömistä nimi ja arvo pareista. Json on hyvä vaihtoehto Xml kielelle. (Foundations of Ajax, 68 - 69)

## Ajax

Asynkroninen JavaScript ja xml kirjasto. Ajaxin tarkoituksena on hakea tietoa serveriltä niin, että sivun sisältö latautuu jouhevasti ja reaaliajassa. Ajaxilla pyritään luomaan käyttäjälle vaikutelma, jossa koko sivua ei ladata uudelleen request pyyntöjen yhteydessä. (Learning Web Design 4th Edition s. 497)

## Mustache

Mustache on mallipohjainen syntaksikieli, joka sisältää vähemmän logiikkaa, kuin esimerkiksi JavaScript.

Sitä voidaan käyttää esimerkiksi Jsp sivuilla, vähentämään koodin ja logiikan tarvetta.

Mustache hyödyntää templaatteja, eli valmiita pohjia sivun uudelleen luomiseen.

Kehittäjän tulee vain pitää huolta, että pohja ja tieto ovat oikein muodostettua.

Mustache hoitaa muuttujien lisäämisen Jsp sivulle tämän jälkeen.

(Learning JavaScript Design Patterns, s. 114 ja YUI 3 Cookbook)

## Handlebarsjs

Handlebars on koodikirjaston laajennus mustache tekniikkaan.

```
<ul class="treelist_ul">
  {{#recursive children}}
  <li>
    <div class="task">
      <div id="{{id}}" class="task-tag"></div>
      <div class="acceptTask"></div>
      <div class="task-name float_left"...>
      <div class="startTask"></div>
      <div class="float_left">
        <select class="selectPriority"...>
      </div>
      <div style="clear:both;"></div>
    </div>
  </li>
  {{#if children}}
  <ul class="treelist_ul">
    {{recursive children}}
  </ul>
  {{/if}}
  {{/recursive}}
</ul>
```

*Kuva 1 Handlebars rekursio templaatti*

Handlebars käyttää tavallista Html kieltä, mutta syntaksina toimii kaksinkertainen aaltosulje. Kuvassa on esimerkki handlebars templaattista.

(Smashing Mobile Web Development)

## **Css**

Css on erillinen koodikieli omilla syntakseillaan, jota hyödynnetään esimerkiksi Html ja jsp sivuilla. Tarkoituksena vähentää vaivaa, joka aiheutuu sivuston tyylien muokkauksesta.

Vaihtoehtona prototyypissä olisi voinut käyttää sass tai scss tyylejä. Näiden avulla olisi voinut kirjoittaa prototyyppiin hieman lyhyempää ja tehokkaampaa css kieltä.

(Learning Web Design 4th Edition s. 207)

## **Git**

Git on tehokas, erittäin mukautuva versionhallinta. Git eroaa muista versionhallintatyökaluista siten, että tämän avulla pystytään aidosti mukautumaan tuhansien kehittäjien tarpeisiin. Gitissä olevaa tietoa ei pystytä enää jälkikäteen muokkaamaan, jonka ansiosta versioiden vertaaminen nopeutuu huomattavasti.

(Version Control With Git 2nd Edition s. 19)

Omien kokemusten perusteella, Git on huomattavasti tasapainoisempi verrattuna muihin versionhallinta työkaluihin, kuten cvs tai svn.

## **Cacoo**

Cacoo on kaavioiden tekoon tarkoitettu selainpohjainen editor. Editori sisältää useita eri ikoneita, jolla pystytään tekemään sivustokarttoja tai esimerkiksi pylväsdiagrammeja. Cacoo on erittäin hyvä vaihtoehto Microsoftin Visio tuotteelle.

(Mobile Web Design For Dummies, s 131)

## **YouTrack**

Käytetään yleisesti työtehtävien tai projektien hallinnoimiseen. Projektin aikana ohjelmaan on kirjattu kaikki jatkokehitykseen tulevat ehdotukset ja ominaisuudet.

### **3. Projekti**

Tarkemmin projektin menetelmistä, tavoista ja riskeistä on liitteessä projektisuunnitelma.

#### **3.1. Tavoite**

Projektin tavoitteeksi asetettiin mahdollisimman helppokäyttöinen tehtävienhallinta moduuli, jolla pystytään hallitsemaan henkilön työtehtäviä. Hallitseminen sisältää työtehtävien lisäämisen, poistamisen, hakemisen ja listauksen.

Lisätavoitteeksi asetettiin, että moduuli tulee olla irrotettavissa IMS toimintajärjestelmästä tarvittaessa omaksi kokonaisuudeksi.

Tavoite muuttui hieman määrityksien ja ominaisuuksien lisääntyessä ja tavoitteeseen lisättiin automaation, erillisen hallintaosion tekeminen ja hakutoiminnon pois jättäminen.

#### **3.2. Laadullinen tavoite**

Projektin toteutus on kommentoitua, testattu ja toimivaa. Projektista tehdään pienimuotoinen määrittäydokumentti ja jatkokehityslista, jonka pohjalta prototyyppiä voidaan jatkaa myöhemmin.

Prototyypin ominaisuudet on käyty läpi myyntipuolen työntekijän kanssa ja kaikki kehitysehdotukset on kirjattu ylös jatkokehitystä varten. Ominaisuuksien läpi käynti toimi hyvänä käyttäjä testinä ja tämän avulla selvisi asioita jotka eivät toimi vielä täysin halutulla tavalla.

### **3.3. Projektin kustannukset**

Prototyypin teosta ei ole aiheutunut IMS Business Solutions Oy: le erillisiä kustannuksia. Projekti on toteutettu normaalina palkallisena työnä.

### **3.4. Resurssit**

Aikaa projektiin käytettiin alussa hieman vähemmän verrattuna koko projektin resurssien määrään. Tarkempaan projektiin liittyvät aikataulutukset löytyvät liitteestä loppuraportti.

### **3.5. Riskit**

Projektisuunnitelmaa tehtäessä suurimmiksi riskeiksi selvisi projektin myötä tekniikka ja rajaus.

#### **3.5.1. Rajaus**

Heti projektin alussa kartoitettiin riskejä projektisuunnitelmaan. Suurimmaksi riskiksi projektissa ilmeni projektin rajaus, koska projekti tulee olemaan valtavan suuri., jonka johdosta projekti jaettiin useaan eri kokonaisuuteen työmäärän selvittämiseksi.

Ongelma ratkaistiin siten, että toimintojen ja määritysten kartoitusta tehdään koko prosessin ajan.

#### **3.5.2. Tekniikka**

Projektissa on osittain tuntemattomia tekniikoita, joiden opetteleminen hidastaa omalta osaltaan projektin etenemistä hieman. Tuntemattomat tekniikat pyritään opettelemaan mahdollisimman tehokkaasti ja tarvittaessa pyytämään vinkkejä projektin toimeksiantajalta.



### **3.5.3. Työaika**

Suunnitelmaa tehtäessä työaika ilmeni oleelliseksi riskiksi, koska projektin alkuvaiheessa oli myös useita muita projekteja. Ennaltaehkäisevänä toimenpiteenä toimi projektin määrittäminen mahdollisimman tarkasti ja projektin rajaus, joka oli aiemmin myös riskinä.

### **3.6. Ajoitus**

Projektiin valittiin muutaman viikon sprintit vaikka projektissa on vain 1 tuotekehittäjä. Sprinttien vaiheet kirjataan backlogiin, jotta tehtävien kulkua olisi helpompi seurata.

Ajoituksien kartoittaminen vähensi huomattavasti työajan käyttöä projektin alkuvaiheessa, jossa työtä tehtiin muutamana päivinä viikossa, koska projektissa ei tarvinnut selvittää mitä on tehty jo aikaisemmin.

### **3.7. Projektin kulku**

Projekti eteni aikataulun mukaisesti pieniä viivästyksiä mainitsematta.

Alussa käytettiin suunnitelman mukaisesti vähemmän tunteja kuin lopussa.

#### **3.7.1. Aloitus**

Projektin alussa kartoitettiin tietokantaratkaisua ja määriteltiin ominaisuuksia, joita prototyyppi tulisi sisältää. Tietokannan kartoitus todettiin todella tärkeäksi, koska tämän vaikutus on suuri toiminnallisuuksien teon alkaessa.

Muutamia päiviä ennen aloituskokousta projektissa oli valmiina alustavat määritykset, tietokantaratkaisu ja joitain toimivia toiminnallisuuksia. Kokouksessa näytettiin tulokset, sekä käytiin läpi projektisuunnitelmaa.

### **3.7.2. Välivaihe**

Välivaiheessa keskityttiin enemmän toiminnallisuuksien lisäykseen ja toteutettiin tehtävienhallinta moduuli. Ennen ohjauskokousta toteutuksessa ilmeni tarve suurempaan kantamuutokseen, joka vaikuttaa kauttaaltaan prototyypin toiminnallisuuksissa.

Toteutetut toiminnallisuudet ja moduuli esiteltiin koodeineen ohjauskokouksessa. kokouksessa päätettiin, että lisää toiminnallisuuksia ei kannata tehdä, jotta projekti pysyy varmasti aikataulussa.

### **3.7.3. Päätösvaihe**

Ohjauskokouksen jälkeen aloitettiin prototyypin laajemmat testaukset kolmella eri selaimella. Testauksen yhteydessä löytyi muutamia selainkohtaisia ongelmia, joita ei olisi ilman laajempaa testausta saatu selville ja korjattua.

Testaukseen liittyvä dokumentaatio löytyy erillisestä liitteestä.

Päätösvaiheen aikana prototyyppiin lisättiin tarvittavat kommentit ja koodia siivottiin ja kirjoitettiin uusiksi, jotta jatkokehitys olisi mahdollisimman helppoa myöhemmin.

Jatkokehitykseen liittyviä ideat mietittiin ensin omalta osalta, jonka jälkeen myynnin ja toimeksiantajan ehdotukset kartoitettiin ja kirjattiin kehitysehdotusdokumenttiin ja YouTrack ohjelmistoon myöhempää kommentointia ja arviointia varten.

## 4. Lopputulokset

Projektissa on saavutettiin lähes kaikki asetetut tavoitteet. Tärkeimpänä niistä on toimiva prototyyppi, määrittelyt, testausdokumentti ja jatkokehitysehdotukset.

Toiminnallisuuksista vain hakutoiminnallisuus päätettiin jättää tekemättä, koska projektin koko laajenee vielä myöhemmin sen verran paljon, ettei hakuun kannattanut käyttämään yrityksen resursseja vielä tässä vaiheessa.

Prototyyppiä varten on tehty laaja testaus seuraavilla selaimilla: Firefox, Chrome, IE 7, 8, 9 ja 10. Testauksessa todetut selainkohtaiset ongelmat on korjattu kaikilta osin.

Laajempi testausdokumentti löytyy liitteestä testaus.

Koodi on kommentoitu ja siivottu useaan kertaan niin kuin laadullisissa tavoitteissa määriteltiin projektisuunnitelmassa. Prototyyppi on toimiva ja se on lisätty versionhallintaan myöhempää jatkokehitystä varten.

### 4.1. Liiketoiminta saavutukset

Ims voi prototyypin myötä jatkokehittää tehtävienhallinta ominaisuutta prototyypin pohjalta. Jatkokehitys tulee olemaan paljon helpompaa, nopeampaa ja kehitykseen ei tarvitse enää huomattavia määriä resursseja.

Prototyyppi on rakennettu uudistetulle pohjalle, jonka ansiosta projektin sisältöä ei tarvitse täysin uusia lähivuosina.

## 4.2. Omat saavutukset ja oppiminen

Ajoittain projektin aikana on tullut uusien tekniikoiden kanssa haasteita, mutta niistä olen selvinnyt hyvin hankkimalla tietoa itsenäisesti ja viime kädessä kysymällä neuvoa muilta tuotekehittäjiltä.

Suurimpana haasteena oli projektissa käytettävä Spring Framework. Kehys ei ollut ennestään täysin tuttu ja opettelin kehystä samaan aikaan koulussa vanhalla tekniikalla ja työpaikalla uudemmalla tekniikalla. Kuitenkin yli vuoden työkokemuksella, kehyksen oppiminen sujui nopeasti ja suuremmilta ongelmilta vältyttiin.

Toiminnallisuuksien toteutuksia on käyty läpi ajoittain yrityksen työntekijöiden kanssa, jotta prototyypin laadullinen puoli säilyy tasokkaana

Projektin koko on suurin tähän asteisista jossa olen ollut mukana. Kuitenkin projektin rajaus on onnistunut niin hyvin, että mitään paniikin omaista kiirettä ei ole tullut, kuten joskus tietotekniikan projekteissa on tapana käydä.

Tekninen osaaminen on kasvanut projektin myötä valtavasti. Jatkossa uusien projektien tekeminen tulee olemaan huomattavasti helpompaa, kun uudet tekniikat ovat jo ennestään tuttuja.

## 5. Jatkokehitys

Kehityksessä on pyritetty ottamaan huomioon tulevaisuudessa tulevat muutokset niin hyvin kuin mahdollista.

Määrittämien ja jatkokehitys dokumentaation myötä kehittäminen on jatkossa jouhevampaa.

Jatkokehitys ideat löytyvät salaisesta liitteestä kehitysehdotukset. Jatkokehitys listaus kattaa kaikki tarpeelliset ominaisuudet joita tehtävienhallinta myöhemmin tarvitsee.

Jatkokehitys listaus kuvauksineen on tehty käyttäen YouTrack ohjelmistoa.

## Lähteet

Spring in Action 3rd Edition, Craig Walls

Spring in Action 2nd Edition, Craig Walls

Pro Spring 2.5 Books for professionals, Anirvan Chakraborty, Jessica Ditt, Aleksa Vukotic and Jan Machacek (Aug 21, 2008)

Java Persistence With Hibernate 2012, Christian Bauer and Gavin King (Nov 24, 2006)

Manning The Well Grounded Java Developer, Benjamin J Evans and Martijn Verburg (Jul 18, 2012)

Learning Web Design: A Beginner's Guide to HTML, CSS, JavaScript, and Web Graphics, Jennifer Niederst Robbins

Foundations of Ajax, Nathaniel T. Schutta, Ryan Asleson

Learning JavaScript Design Patterns, Addy Osmani

YUI 3 Cookbook, Evan Goer

Version Control With Git 2nd Edition, Powerful Tools and Techniques for Collaborative Software Development, Jon Loeliger, Matthew McCullough

Smashing Mobile Web Development, G. Avola, Jon Raasch

Pro JPA 2: Mastering the Java™ Persistence API (Expert's Voice in Java Technology), Mike Keith

Mobile Web Design For Dummies, Janine Warner, David LaFontaine

<http://www.handlebarsjs.com>

<https://github.com/wycats/handlebars.js/>

<https://github.com/janl/mustache.js>

[http://cacoo.com/lang/en/about\\_cacoo](http://cacoo.com/lang/en/about_cacoo)

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

## **Liitteet**

Loppuraportti

## **Salaiset liitteet**

Määrittäydokumentti

Kehitysehdotukset

Testausdokumentti

Prototyypin lähdekoodi



# **Tehtävienhallinta LOPPURAPORTTI**

**Tuomo Pohjola**

**1.3.2013**

## SISÄLTÖ

1. Saavutukset liiketoiminnan kannalta.....	22
2. Kokemukset ja saavutukset.....	22
3. Resurssien opinnäytetyössä.....	23
4. Salaiset liitteet.....	25
Kehitysehdotukset.....	25
Liite 2. Projektin dokumentaatio.....	25
Tietokantaselvitys.....	25
Testaus.....	25
Projektisuunnitelma .....	25

## **1. Tausta**

Tehtävienhallinta projektin toimeksiantajana toimii IMS Business Solutions Oy.

Opinnäytetyö sisältää yrityssalaisuuksiksi luokiteltuja dokumentteja, joita ei voi tämän vuoksi näyttää opinnäytetyössä.

IMS Business Solutions Oy kehittää ja myy omaa selainpohjaista IMS toimintajärjestelmää. Toimintajärjestelmän avulla asiakkaat voivat hallita kaikkea olemassa olevaa materiaaliaan tai luoda uutta IMS ohjelman avulla.

## **2. Saavutukset liiketoiminnan kannalta**

IMS voi prototyypin myötä jatkokehittää tehtävienhallinta ominaisuutta paljon helpommin ja kehitykseen ei kulu enää niin paljon aikaa.

## **3. Kokemukset ja saavutukset**

Projektin koko on suurin tähän asteisista joihin olen ollut mukana. Kuitenkin projektin rajaus on onnistunut sen verran hyvin, että mitään paniikin omaista kiirettä ei ole tullut.

Ajoittain projektin aikana on tullut uusien tekniikoiden kanssa haasteita, mutta niistä on selvinnyt hyvin hankkimalla tietoa ja viimekädessä kysymällä neuvoa.

Projektissa on saavutettiin lähes kaikki asetetut tavoitteet. Vain hakutoiminnallisuus päätettiin jättää tekemättä, koska projekti laajenee vielä myöhemmin sen verran paljon, ettei hakua kannata alkaa vielä tekemään. Projektissa on tehty testaus ja koodi on kommentoitu ja siivottu useaan kertaan niin kuin laadullisissa tavoitteissa oli määriteltyinä.

Kokonaisuudessaan projektissa olen oppinut paljon useista eri tekniikoista ja projektin hallinnasta.

#### 4. Resurssien opinnäytetyössä

Ajankäyttö on arvioitu lähes tulkoon oikein projektin aikana. Marras- Joulukuu välisenä aikana tunteja kertyi vähän niin kuin oli odotettavissa jo aihe-ehdotuksen lähetyksessä. Tammi- Helmikuu välisenä aikana tunteja on kertynyt muutama tunti ylimääräistä, kun projekti lähestynyt loppuaan.

Noin 125 tuntia Marraskuu – Joulukuu välisenä aikana ja noin 265-280 tuntia

Tammikuu – Helmikuu välisenä aikana.

Alku sprinteillä n.15 tuntia / vk ja vuonna 2013 37.5 / vk.

Backlogissa sprinttien sisällöistä enemmän tietoa.

<b>Sprint</b>	<b>Vko/ vuosi/ tunnit per viikko</b>
Sprint 1	44, 46 / 2012/ 15t
Sprint 2	47, 38 / 2012/ 15t
Sprint 3	49,50 / 2012/ 15t
Sprint 4	51 /2012/ 15t
Sprint 5	1, 2 / 2013/ 37.5t
Sprint 6	3, 4 /2013/ 37.5t
Sprint 7	5, 6 /2013/ 37.5t
Sprint 8	7, 9/2013/ 37.5t
Sprint 9	10/2013/37.5t

## 2. Lopputulokset

Projektissa on saavutettiin lähes kaikki asetetut tavoitteet. Tärkeimpänä niistä on toimiva prototyyppi, määrittelyt, testausdokumentti ja jatkokehitysehdotukset.

Toiminnallisuuksista vain hakutoiminnallisuus päätettiin jättää tekemättä, koska projektin koko laajenee vielä myöhemmin sen verran paljon, ettei hakuun kannatanut käyttämään yrityksen resursseja vielä tässä vaiheessa.

Prototyyppiä varten on tehty laaja testaus seuraavilla selaimilla: Firefox, Chrome, IE 7, 8, 9 ja 10. Testauksessa todetut selainkohtaiset ongelmat on korjattu kaikilta osin. Laajempi testausdokumentti löytyy liitteestä testaus.

Koodi on kommentoitu ja siivottu useaan kertaan niin kuin laadullisissa tavoitteissa määriteltiin projektisuunnitelmassa. Prototyyppi on toimiva ja se on lisätty versionhallintaan myöhempää jatkokehitystä varten.

### **3. Salaiset liitteet**

#### **Kehitysehdotukset**

#### **Projektisuunnitelma**

Suunnitelma mitä projektissa piti tehdä, riskianalyysit ja ajan käyttö.

#### **Projektin dokumentaatio**

Dokumentointi koostuu määrittäydokumentista, jossa on tarkemmin selitettynä kaikki toiminnot jotka toteutettiin. Opinnäytetyössä on avattu mitä varten tehtävienhallinta on tehty ja ratkaisiko tämä ongelman.

#### **Tietokantaselvitys**

Tietokanta löytyy tarkemmin määrittäydokumentista.

#### **Testaus**

Testausta varten on luotu erillinen excel-tilukko, johon on asetettu toiminnallisuuksia joiden tulisi toimia projektin päätyttyä.