

# Loppuraportti - Tietokantasovellus

Marko Haanranta

1. syyskuuta 2013

# 1 Johdanto

Toteutin "aineopintojen harjoitustyö: Tietokantasovellus"kurssin harjoitustyönä projektin työaikaseuranta järjestelmän. Työ toteutettiin valmiista aiheesta johon löytyy linkki tämän dokumentin osiosta liitteet.

## 1.1 Järjestelmän tarkoitus

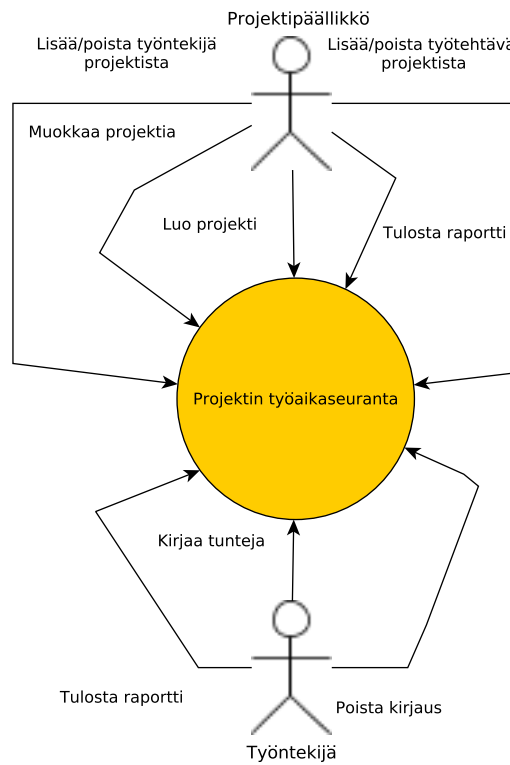
Projektin työaikaseuranta järjestelmää käytetään projekteissa työaikakirjanpitoon. Projektia suunniteltaessa pyritään määrittelemään millaisia erilaisia työtehtäviä projekti sisältää ja miten paljon työaikaa näihin tulisi varata. Projektin aikana työntekijät sitten kirjaavat tekemänsä työtunnit työaikaseuranta järjestelmään sopivien työtehtävien alle.

Projektin työaikaseuranta järjestelmällä kerätään yksityiskohtaista tietoa siitä miten hyvin ennen projektin alkua tehdyt arviot toteutuvat. Tavoitteena on paitsi valvoa työntekijöiden ajankäyttöä myös helpottaa tulevien projektien suunnittelua, kun voidaan käyttää hyväksi tätä kerättyä tietoa toteutusta projektista.

## 1.2 Toteutusympäristö

Työ on toteutettu Netbeans kehitysympäristössä. Ohjelmointi kielenä käytin Javaa ja sen tarjoamaa servlet tekniikkaa. Tietokantana käytin postgresql tietokantaa.

## 2 Yleiskuva järjestelmästä



Kuva 1: Järjestelmän sidosryhmäkaavio

## 3 Käyttötapaukset

- 1) Kirjautuminen järjestelmään
- 2) Projektin perustaminen
- 3) Henkilön liittäminen projektiin
- 4) Työaikakirjauksen teko
- 5) Raportti

### 3.1 Kirjautuminen järjestelmään

Käyttäjä: projektipäällikkö, työntekijä

Tavoite: Kirjautua järjestelmään

Alkuehdot:

Järjestelmän kirjautumissivu on toteutettu.

Käyttötapauksen kulku:

Käyttäjä syöttää kirjautumissivulle käyttäjätunnuksen ja salasanan. Käyttäjätunnuksen ja salasanan lähettämisen jälkeen käyttäjä näkee olevansa kirjautunut järjestelmään.

Poikkeustilanteet:

Salasana tai käyttäjätunnus on väärä. Tällöin käyttäjää ei pidä päästää sisään järjestelmään vaan tulee tarjota kirjautumissivu uudestaan ja kertoa että käyttäjätunnus tai salasana oli väärin.

### **3.2 Projektin perustaminen**

Käyttäjä: projektipäällikkö

Tavoite: Lisätä projekti järjestelmään ja määritellä sille työtehtäviä.

Alkuehdot:

Käyttäjätunnus ja salasana ovat olemassa. Käyttäjä on kirjautunut järjestelmään.

Käyttötapauksen kulku:

Käyttäjä luo projektin. Käyttäjä lisää työtehtäviä projektiin.

Jälkiehto:

Luotu projekti näkyy järjestelmässä.

### **3.3 Henkilön liittäminen projektiin**

Käyttäjä: projektipäällikkö

Tavoite: Liittää työntekijä projektiin.

Alkuehdot:

Työntekijä roolin omaava käyttäjä on luotu järjestelmään.

Käyttötapauksen kulku:

Projektin hallintaoikeuksin varustettu käyttäjä valitsee listalta projektin johon haluaa lisätä työntekijän ja lisää sen jälkeen työntekijän projektiin. Poikkeustilanteet:

Projektipäällikkö roolin omaavan käyttäjän liittäminen projektiin.

### **3.4 Työaikakirjausten teko**

Käyttäjä: työntekijä

Tavoite: Työaikakirjauksen suorittaminen.

Alkuehdot:

Käyttäjätunnus ja salasana ovat olemassa. Käyttäjä on kirjautunut järjestelmään. Järjestelmässä on projekteja ja niille on määritelty useita eri työtehtäviä.

Käyttötapauksen kulku:

Käyttäjä valitsee projektin. Käyttäjä valitsee projekti kohtaisen työtehtävän sekä kirjaa tunnit, päivämäärän ja mahdollisen selityksen. Käyttäjä näkee kirjaamansa tiedot järjestelmässä.

### 3.5 Raportti

Käyttäjä: projektipäällikkö, työntekijä

Tavoite: Tulostaa tietyllä aikavälillä tehdyistä kirjauksista raportti.

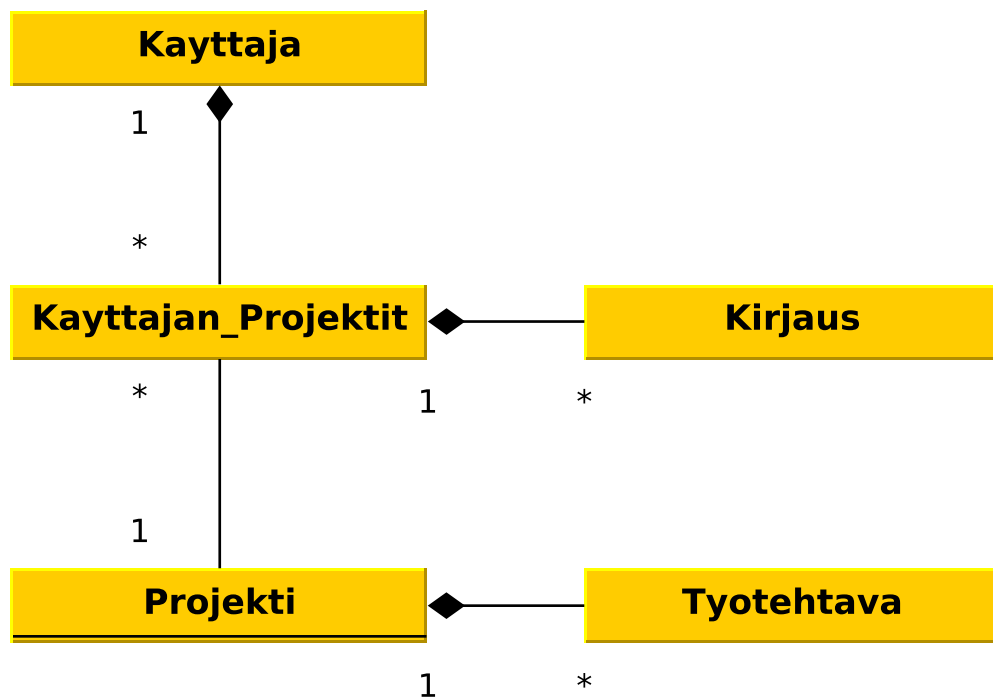
Alkuehdot:

Järjestelmässä on projekteja joihin on lisätty työtehtäviä. Käyttäjä on kirjautunut järjestelmään.

Käyttötapauksen kulku:

Käyttäjä valitsee aikavälin jolta raportin haluaa ja painaa tulosta raportti painiketta. Raportti ilmestyy verkko-sivulle. Raportista näkee kirjaukset, jotka kukin työntekijä on aikavälillä tehnyt. Jos käyttäjä on työntekijä niin hän voi nähdä vain omat kirjauksensa.

## 4 Järjestelmän tietosisältö



Kuva 2: Järjestelmän käsitekaavio

#### 4.1 Tietokohde: Käyttäjä

Attribuutti	Arvojoukko	Kuvailu
kayttajatunnus	merkkijono(255)	primary key
salasana	merkkijono(255)	
nimi	merkkijono(255)	
rooli	int	Nolla tarkoittaa työntekijää ja kaikki sitä arvot suuremmat projektipäällikköä.

#### 4.2 Tietokohde: Projekti

Attribuutti	Arvojoukko	Kuvailu
projektin_nimi	merkkijono(255)	primary key
tyotuntibudjetti	float(4)	
alkamispäivä	date	Projektin alkamispäivämäärä.
loppumispäivä	date	Projektin loppumispäivämäärä.

#### 4.3 Tietokohde: Työtehtävä

Attribuutti	Arvojoukko	Kuvailu
tyotehtavan_nimi	merkkijono(255)	primary key
budjetoidut tyotunnit	float(4)	
projektin_nimi	merkkijono(255)	foreign key
Työtehtävä on olemassa olo riippuvainen projektista.		

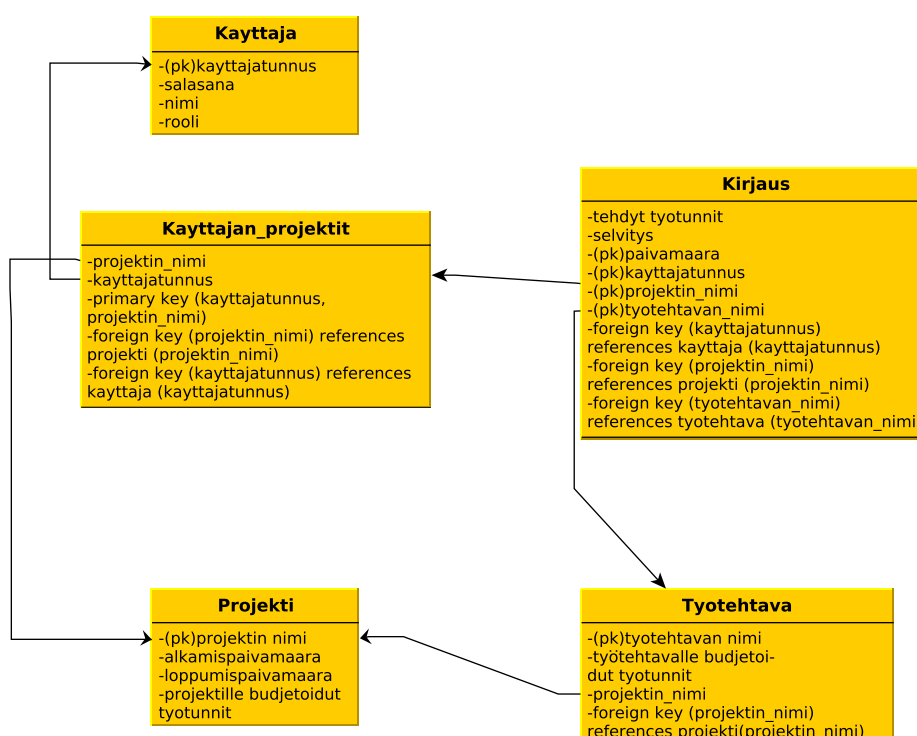
#### 4.4 Tietokohde: Kirjaus

Attribuutti	Arvojoukko	Kuvailu
paivamaara	date	
tehdyt tyotunnit	float(4)	
selvitys	merkkijono(1000)	
kayttajatunnus	merkkijono(255)	foreign key
projektin nimi	merkkijono(255)	foreign key
tyotehtavan nimi	merkkijono(255)	foreign key
Kayttajatunnus, projektin_nimi, tyotehtavan_nimi ja paivamaara muodostavat yhdessä primary keyn. Kirjaus on olemassa olo riippuvainen Kayttajan_Projektit		

## 4.5 Tietokohde: Kayttajan Projektit

Attribuutti	Arvojoukko	Kuvailu
kayttajatunnus	merkkijono(255)	foreign key
projektin nimi	merkkijono(255)	foreign key
Kayttajatunnus ja projektin_nimi muodostavat yhdessä primary keyn. Kayttajan_Projektit on olemassa olo riippuvainen Kayttajasta.		

## 5 Relaatiotietokantakaavio



Kuva 3: Järjestelmän relaatiotietokantakaavio

## 6 Järjestelmän yleisrakenne

Järjestelmä koostuu pääosin jsp-sivuista ja servleteistä.

Yleisesti ottaen ohjelman toiminta on seuraava. Kirjautunut käyttäjä syöttää tietoja lomakkeelle ja painaa submit nappia. Servlet saa käsiteltäväkseen määrätyn lomakkeen tiedot jsp sivulta ja sen jälkeen tekee tarkistuksia ja jos kaikki tarvittavat tiedot ovat kasassa tekee tietokanta kyselyn tai vie

tietoa tietokantaan. Tietokannan käsittelyä varten on TietokantaYhteys.java luokka jonka metodeita käyttäen tietokannan kanssa asioidaan.

## 6.1 Jsp-sivut

JSP sivut löytyvät projektin juuresta löytyvästä Web Pages kansioista. Siellä on myös jsp-sivujen muotoiluun käytetty css-tiedosto.

## 6.2 Servletit

Kaikki servletit ovat source packages-kansiossa sijaitsevassa servlet kansiossa.

## 6.3 Tietokanta

Kaikki tietokannan käsittely tapahtuu TietokantaYhteys-luokan kautta. TietokantaYhteys-luokka on source packages-kansiossa sijaitsevassa tietokanta kansiossa. Tässä kansiossa on myös luokat jotka kuvaavat tietokanta tauluja. Näistä luokista luotuja olioita käytetään tiedonsiirtoon servletien ja TietokantaYhteys-luokan metodien välillä.

## 6.4 SQL-luontilauseet

Tietokantataulujen luonti lauseet ovat source packages-kansiossa sijaitsevassa SQL-kansiossa.

## 6.5 Kirjastot

Projekti käyttää kahta kirjastoa. JSTL1.1 on tarpeen jsp sivujen takia. Lisäksi käytetään postgresSQL JDBC driver luokkaa tietokantayhteyden takia. Kirjastot löytyvät libraries-kansiosta projektin juuresta.

# 7 Järjestelmän komponentit

style.css: Web pages kansiossa sijaitseva tyyli tiedosto jsp sivujen muotoiluun.

luontilauseet.sql: SQL kansiossa on projektin tietokanta taulujen luontilauseet.

## 7.1 JSP-sivut

Yleistä: Jokaisella kirjautumisen jälkeen näytettävällä sivulla on "Kirjaudu ulos"painike. Painikkeen käsittelyn hoitaa KirjauduUlosServlet.

index.jsp: Sisältää linkin kirjautumissivuille.



kirjaudu.jsp: Järjestelmään sisään kirjautuminen. Submitin painamiseen jälkeen käsittely ohjautuu Kirjaudu servletille.

kirjautunut.jsp: Sen jälkeen kun käyttäjä on onnistuneesti kirjautunut sisään järjestelmään hän näkee kirjautunut.jsp sivun. Sivun näkyy erilaisena käyttöroolista(projektipäällikkö, työntekijä) riippuen. Projektipäällikkö näkee sivulla formit projektien luomiseen(LisaaProjektiServlet) ja poistamiseen(PoistaProjektiServlet) sekä listan olemassa olevista projekteista. Työntekijä näkee listan projekteista joihin hänet on liitetty työntekijäksi. Klikkaamalla jonkin projektin linkkiä käyttäjä ohjautuu kyseisen projektin sivulle.

projekti.jsp: Projektipäällikkö voi muokata kyseisen projektin päivämääriä ja työtuntibudjettia. Lisäksi hän voi lisätä ja poistaa sekä työtehtäviä että työntekijöitä projektista.

kirjaatunteja.jsp: Työntekijä voi kirjata tekemänsä työtunnit tämän sivun kautta. Ainoastaan työntekijä roolin käyttäjä voi kirjata tunteja.

raportit.jsp: Projektipäällikkö ja työntekijä voivat tulostaa raportin kirjauksista. Projektipäällikön raportissa näkyvät kaikkien työntekijöiden kaikkiin projekteihin tekemät kirjaukset tietyllä aikavälillä. Työntekijän raportissa näkyvät vain hänen itsensä tekemät kirjaukset tietyllä aikavälillä.

## 7.2 Servletit

KirjauduServlet: Servlet hoitaa käyttäjätunnuksen ja salasanan tarkistamisen tietokannasta. Jos käyttäjätunnus tai salasana on väärä niin kirjaudu servlet tulostaa virhe ilmoituksen. Jos ne ovat oikein niin siirrytään ProjektitServletin suorittamiseen. Servlet myös luo HttpSession luokan ilmentymän johon tallennetaan ne tiedot sisään kirjautuneesta käyttäjästä joita tarvitaan ohjelman suorituksen aikana.

KirjauduUlosServlet: Jokaisella kirjautumisen jälkeen näytettävällä jsp sivulla on "Kirjaudu ulos"painike. KirjauduUlosServlet hoitaa tämän painikkeen painamisen aiheuttamat jatkotoimenpiteet eli heittää session mäkeen ja ohjaa käyttäjän takaisin sisäänkirjautumis sivulle.

ProjektitServlet: Tämä sivu on kytköksissä kirjautunut.jsp sivuun ja hoitaa projektin lisäämisen aiheuttamat toimet. Ensin tarkistetaan ettei tämän nimistä projektia jo ole olemassa. Jos ei niin lisätään projekti tietokantaan annetuilla parametreilla. Servlet hoitaa myös käyttäjälle näytettävien listojen muodostamisen.

LisaaTyontekijaProjektiinServlet: Kun projekti on luotu niin voidaan liittää siihen työntekijöitä.

LisaaTyotehtavaServlet: Projektipäällikkö voi lisätä työtehtäviä sen jälkeen kun hän on luonut projektin.

MuokkaaProjektiaServlet: Projektipäällikkö voi muokata projektin tietoja valitsemalla muokattavan projektin. Projektin nimeä ei pysty muuttamaan.

PoistaProjektiServlet: Projektipäällikkö voi poistaa projektin. Projektin poisto poistaa myös kaikki siihen lisätyt työtehtävät ja kaikki siihen tehdyt kirjaukset.

PoistaKirjausServlet: Työntekijä voi poistaa tekemänsä kirjauksen.

PoistaTyontekijaServlet: Projektipäällikkö voi poistaa työntekijän projektista. Tällöin myös kaikki tämän työntekijän tekemät kirjaukset poistetaan kannasta.

PoistaTyotehtavaServlet: Projektipäällikkö voi poistaa työtehtävän projektista. Tällöin myös kaikki tähän tietyn projektin työtehtävään liittyvät kirjaukset poistetaan kannasta.

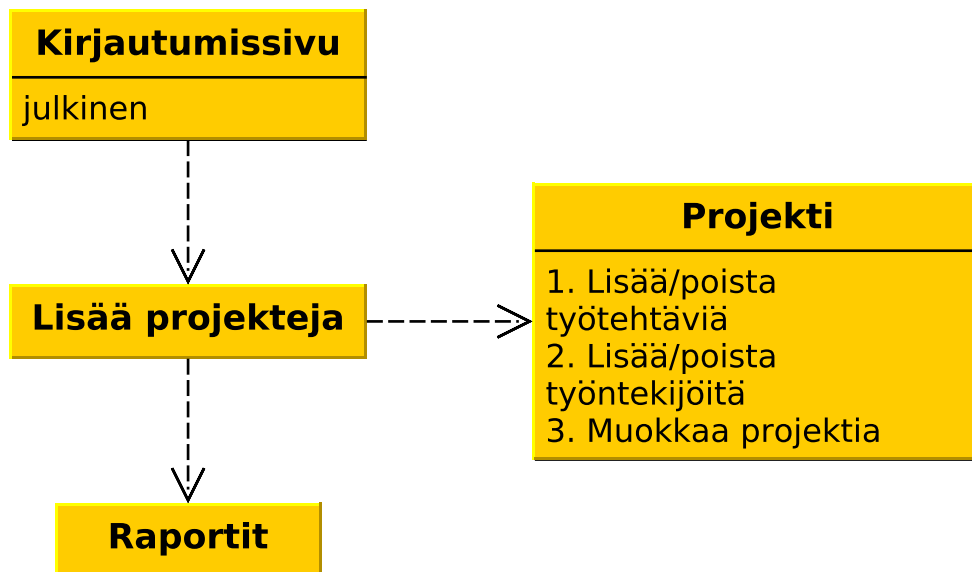
KirjaaTuntejaServlet: Sen jälkeen kun työntekijä on kirjautunut ja valinnut projektin hän voi syöttää työtuntikirjauksia Valitsemalla työtehtävän valikosta ja syöttämällä käytetyn ajan, päivämäärän ja halutessaan selityksen. Servlet hoitaa käsittelyn ja näyttää työntekijän tekemät kirjaukset. Työntekijä voi myös poistaa kirjauksia valitsemalla valikosta työtehtävän nimen ja kirjaus päivämäärän.

RaportitServlet: Työntekijä ja projektipäällikkö voivat tulostaa raportin tiettyjen päivämäärien välillä tehdyistä tuntikirjauksista.

### **7.3 Tietokanta**

TietokantaYhteys.java: Yhteys tietokantaan muodostetaan tämän luokan olioiden kautta. Luokka sisältää kaikki projektin tietokantaoperaatiot. Tietokantakyselyihin käytetään prepaped statementeja, joten sql-injectioiden vaara on pois suljettu.

## **8 Käyttöliittymä**



Kuva 4: Projektipäällikön käyttöliittymä

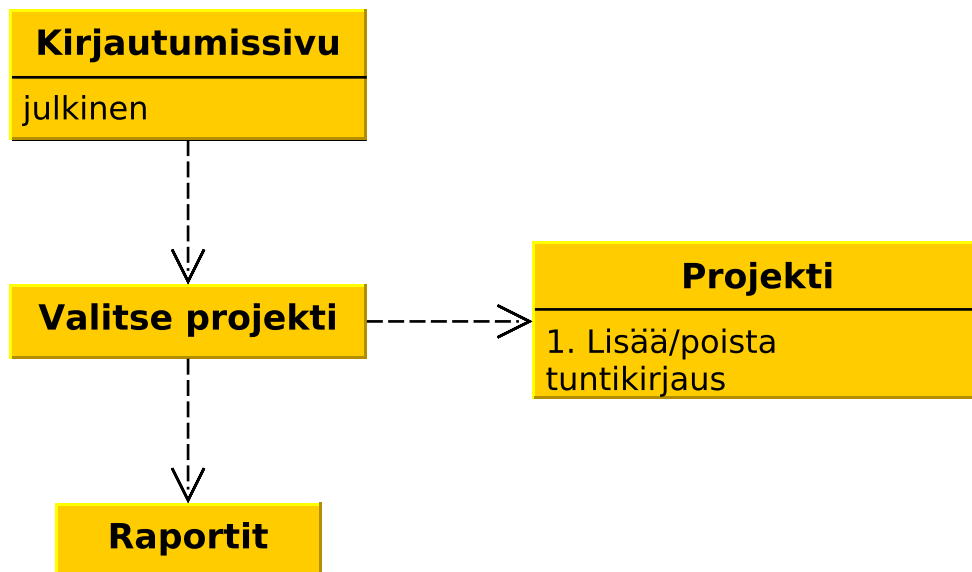
## 9 Asennustiedot

Uudelleen asennettaessa tulee ensin luoda tietokantayhteys postgresql tietokantaan ja luoda sinne sovelluksen käyttämät tietokantataulut. Myös tietokannan käyttäjät on luotava käsin insert lausein.

Esim. insert into kayttaja values ('kayttajatunnus','salasana','nimi','rooli'); Oheisella luontilauseella luodaan käyttäjä joka kirjautuu sisään käyttäjätunnuksella "kayttajatunnus" ja salasanaalla "salasana". Nimeksi on tarkoitus laittaa oma nimi. Rooliksi tulee valita vaikkapa '2' jos haluaa hallinnoida Projektien työaikaseuranta järjestelmää ja '0' jos haluaa tällä käyttäjätunnuksella kirjata tehtyjä työtunteja. Jos haluaa käyttää omaa tietokantaa niin TietoKantaYhteys-luokan DB\_URL, tietokantaKayttaja ja tietokantaSalasana tulee vaihtaa omiin vastaaviin. Sovellus tukee ainostaan postgresql tietokantaa.

## 10 Käyttöohje

Sovellus on asennettu users.cs.helsinki.fi palvelimelle ja sitä pääsee käyttämään tunnuksin admin, mhaanran ja veli. Salasana on sama kuin kukin käyttäjätunnus. admin käyttäjällä on roolina projektipäällikkö eli hän pääsee lisäämään poistamaan ja muokkamaan. Roolin työntekijä käyttäjät voivat kirjata tunteja sen jälkeen kun heidät on liitetty projekteihin.



Kuva 5: Työntekijän käyttöliittymä

Sovellusta pääsee käyttämään osoitteesta:

<http://t-mhaanran.users.cs.helsinki.fi/ProjTyoAikaSeur/>

Kun lisätään projekteja, tuotetaan raportteja tai kirjataan tunteja niin päivämäärät tulee kirjata muodossa "ddMMyyyy". Ilman välejä ja 8 merkkiä eli esim. kuukaudet "1-9"merkataan "01-09".

## 11 Testaus, tunnetut bugit ja puutteet

### 11.1 Testaus

Olen testannut ohjelmaa manuaalisesti koko sen kehityskaaren ajan.

### 11.2 Tunnetut bugit

1. Ääkköset projektin nimessä eivät toimi oikein. Tietokannassa ääkköset ovat ihan oikein mutta kun siirtyy projekti kohtaiselle sivulle niin siellä ne näyttävät ihan joltain muulta kuin mitä tarkoitin.

### 11.3 Puutteet

Syötteiden tarkistus: En tällä hetkellä tarkista onko syöte "vaarallinen"vai ei.

SQL-injectioiden vaaraa ei kuitenkaan tulisi olla koska käytän prepared statementeja.

## 12 Jatkokehitysideat

1. Tietojen poistaminen tietokannasta ei näytä olevan hyvä idea. Syynä ovat riippuvuudet taulujen välillä. Ongelmana on että jos esimerkiksi sallin käyttäjän poistamisen niin poiston yhteydessä myös kaikki hänen tekemät työtuntikirjaukset poistuvat. Tämä ei ole tarkoituksen mukaista. Ratkaisu voisi olla että lisää kaikkiin tietokantatauluihin "deleted"kentän ja jos käyttäjä haluaa poistaa jotain niin "deleted"kenttään merkataan että tieto on poistettu. Normaalit tietokantahaut tarkistavat normaaliehtojensa lisäksi mikä on rivin "deleted"kentän arvo.

2. Voisi lisätä tarkistuksen että työtuntikirjaus on projektin alkamis ja loppumispäivämäärien sisällä.

3. Voisi myös luoda työtehtäviä joihin saa kirjata tunteja vain jälkikäteen. Esim. loma tai kurssi olisi varmaan ihan ok kirjata järjestelmään ennen kuin se alkaa, mutta normaali työtehtävien kirjaaminen etukäteen voisi olla estetty.

## 13 Omat kokemukset

Tietokantasovelluksen tekeminen oli erittäin opettava kokemus. Javan servlet teknologian tietokannan käyttö tuli opittua tämän kurssin aikana ihan käytännön kautta.

Alkuun minulla oli suuria vaikeuksia päästä tekemisen makuun. Mietin myös alussa vähän liikaa tietokannan rakennetta ja yritin hioa sitä mielessäni. Sitteen kun lähdin koodamaan sovellusta ja tuli tarvetta pistää tietoa kantaan niin tietokannan rakenne jotenkin automaattisesti selkeytyi ja tiesin tarvitsevani Kayttajan\_Projektit ja Kirjaus nimiset liittotaulut.

Osittain alkuvaikeuksien syynä kurssisivujen puutteellisuus. esim. yhteyden muodostamiseen tietokantaan koulun users ympäristössä ei ollut mitään selkeää ohjetta.

## 14 Liitteet

Ohjelmakoodi löytyy githubista osoitteesta:

[https://github.com/madhaanr/tietokantasovellus\\_ke2013](https://github.com/madhaanr/tietokantasovellus_ke2013)

Create table lauseet ovat SQL kansiossa ohjelmakoodin mukana. Aihe: <http://www.cs.helsinki.fi/u/lain>