

Aineopintojen harjoitustyö: Tietokantasovellus,

Toteutusdokumentti

Heikki Haapala  
Opiskelijanumero 014090190

7.3.2013

# Sisällysluettelo

1Johdanto.....	2
1.1Järjestelmän tarkoitus.....	2
1.2Toimintaympäristö.....	2
1.3Rajaukset.....	2
1.4Toteutusympäristö.....	2
2Ohjelmiston yleisrakenne.....	3
2.1Selainpuoli.....	3
2.2Palvelinpuoli.....	3
2.3Suojaus.....	4
3Järjestelmän komponentit.....	5
3.1HTML.....	5
3.2JavaScript.....	5
Backbone-mallit (models).....	5
Backbone-näkymät (views).....	6
3.3Java.....	7
Tarkistuksista.....	7
Data-lisäluokista.....	7
4Asennustiedot.....	8
5Käynnistys- ja käyttöohje.....	8

# 1 Johdanto

## 1.1 Järjestelmän tarkoitus

Työvuorosuunnittelu-sovellus on tarkoitettu työvuorojen perussuunnitteluun. Työvuoroja ja työntekijöitä voi hallita ja työvuoroihin voi asettaa työntekijöitä päivä-/työvuorotasolla. Suunnittelun tekee työnantaja ja työntekijät voivat tarkastella työvuorolistaa.

## 1.2 Toimintaympäristö

Järjestelmä vaatii Java Servlet -suoritusympäristön (esim. Tomcat). Järjestelmä testattiin lähes yksinomaan Tomcatilla, Tomcatin lisäksi sovelluksen pitäisi toimia muillakin alustoilla. Sovellus vaatii myös MySQL-tietokannan toimiakseen.

## 1.3 Rajaukset

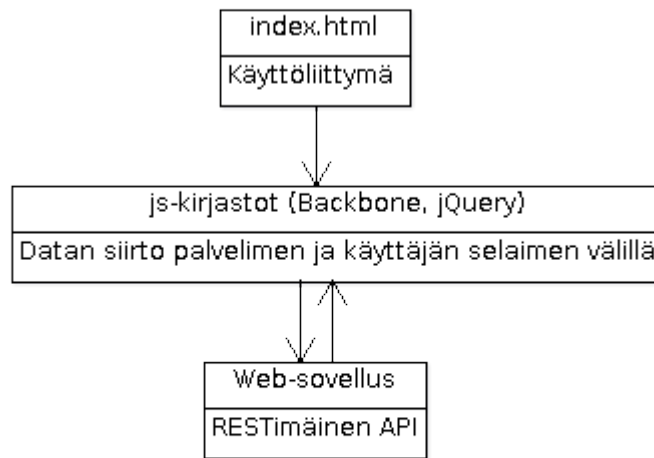
Työssä ei toteutettu mahdollisuutta työntekijöiden lisäämiin ehdotuksiin. Työvuorojen ja käyttäjien muokkaus ei onnistu, ainoastaan lisäys ja poistotoiminnallisuudet ovat tällä hetkellä sovelluksessa.

## 1.4 Toteutusympäristö

Sovellus toteutettiin käyttämällä NetBeans-kehitysympäristöä. Muita työkaluja käytössä oli Git versionhallintaa varten, Maven kirjastonhallintaa varten ja JRebel kehityksen nopeuttamista auttamaan. Toteutuskoneella pyöri MySQL-testipalvelin ja Tomcat-testiympäristö.

## 2 Ohjelmiston yleisrakenne

Sovellusta tehdessä noudatettiin selainpuolella MVC-mallia. Palvelinpuoli tarjoaa RESTimäisen rajapinnan, jota selainpuolen toiminnallisuus käyttää hyväkseen.



Kuva 1: Yleinen rakenne

### 2.1 Selainpuoli

HTML-sivu löytyy sijainnista “webapp/index.html”. Koko käyttöliittymä on tällä yhdellä sivulla. Sivun yleinen rakenne on: headin sisällä css-viittaukset, sitten bodyn alussa div-johon sivumallit rakennetaan ja tämän jälkeen sivupohjat piiloitettuna script-elementteihin. Dokumentin lopussa on viittaukset JavaScript-tiedostoihin, eli ne ladataan vasta viimeisenä.

Handlebars.js-kirjasto muokkaa sivumalleista halutun näkymän. Backbone.js:n näkymät syöttävät Backboneen mallien datan Handlebarsin käsiteltäväksi, joka datan ja pohjien perusteella generoi halutun näkymän.

### 2.2 Palvelinpuoli

Palvelinpuoli käyttää Spring-sovelluskehystä. Springin XML-konfiguraatio löytyy sijainneista “webapp/WEB-INF” ja “resources/META-INF”.

MySQL-konfiguraatio (polku/portti, käyttäjätunnus, salasana) löytyy sijainnista “resources/jdbc.properties”. “resources/sql” alta löytyy määrittelylause ensimmäiselle käyttäjälle “admin”.

Sijainnista “webapp/resources” löytyy palvelimen tarjoamat staattiset resurssit, kuten css-, JavaScript- ja kuvatiedostot. Tiedostot on jaettu alikansioihin (css/img/js) tyyppinsä mukaan.

Palvelinpuolen luokat on jaettu neljään pääosaan: data-, repository-, service- ja controller-luokkiin:

### **Pakkaus “ht.haapala.wss.data”**

Data-pakkaus sisältää luokat, jotka kuvaavat tietokannan sisällön (ORM), sekä lisäluokkia lähinnä auttamaan JSON-datan käsittelyssä.

### **Pakkaus “ht.haapala.wss.repository”**

Repository-pakkauksen luokat tarjoavat DAO-rajapinnan tietokantaan.

### **Pakkaus “ht.haapala.wss.service”**

Service-pakkauksen luokat käyttävät repository-luokkia ja tarjoavat “hiotut” palvelut controller-luokille.

### **Pakkaus “ht.haapala.wss.controller”**

Controller-pakkauksen luokat tarjoavat sovelluksen RESTimäisen rajapinnan ja käyttävät service-luokkia.

## **2.3 Suojaus**

Sovelluksessa käytetään autentikointi- ja autorisointitoimintoihin Springin Security-pakettia. Security-konfiguraatio löytyy sijainnista “webapp/WEB-INF/applicationContext-security.xml”. Sisään- ja uloskirjautuminen on muokattu perustoiminnallisuudesta toimimaan Ajax-pyyntöjen kanssa. Omat toteutukset näihin löytyvät paketista “ht.haapala.wss.authentication”.

Tässä esittely eri käyttäjien oikeuksista sovelluksessa:

(\*\* = polku ja kaikki sen alla)

#### **Yleinen**

/index.html

/resources/\*\*

/login

/logout

#### **Työnantaja**

/wss/users/\*\*

Metoditasolla (kontrollerit) on suojattu /wss/shifts/\*\* ja /wss/plannedshifts/\*\*, koska näihin sijainteihin on päästävä sekä kaikkien käyttäjien, että molempien oikeusluokkien.

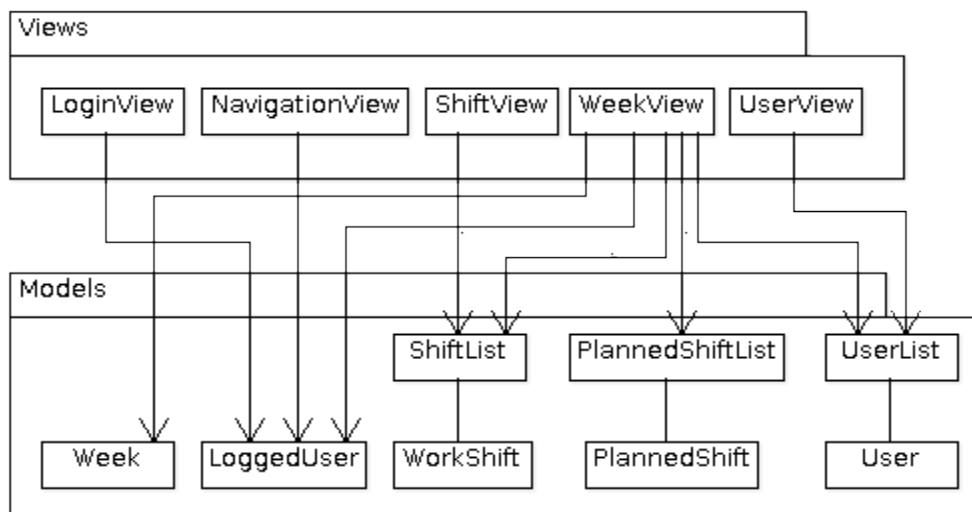
## 3 Järjestelmän komponentit

### 3.1 HTML

Sovelluksen selainpuolen toiminnallisuus tiivistyy sivulle “webapp/index.html”. Kaikki Handlebars.js-kirjaston käyttämät sivupohjat löytyvät tästä dokumentista. Käytetyt css- ja JavaScript-tiedostot ovat linkitettynä tiedostoon.

### 3.2 JavaScript

JavaScript-toteutus painottuu vahvasti Backbone.js-kirjaston käyttöön. Backbone on työkalu, joka auttaa tekemään sovelluksia, jotka noudattavat MVC-mallia. Kuvassa 2 näkyy, mitkä näkymät käyttävät tai muuten tarkkailevat mitäkkin malleja.



Kuva 2: Näkymät ja mallit

#### **Backbone-mallit (models)**

Backbone-mallit kuvaavat lähes samoja olioita, mitä on määritelty palvelinpuolella. Nämä mallit löytyvät tiedostosta “webapp/resources/js/wss-model.js”.

#### **Viikko (Week)**

Mallina sama kuin luokka “ht.haapala.wss.data.Week”. Sisältää näkymässä olevan viikon tiedot: vuosi, viikon numero ja jokaisen viikonpäivän päivämäärä.

#### **Kirjautunut käyttäjä (LoggedUser)**

Mallina yhden käyttäjän tiedot (“ht.haapala.wss.data.WSSUser”). Sisältää kirjautuneen käyttäjän tiedot.

### **Työvuoro (WorkShift)**

Mallina sama kuin luokka "ht.haapala.wss.data.WorkShift". Sisältää työvuoron tiedot.

### **Työvuorolista (ShiftList)**

Kokoelma malleja WorkShift.

### **Suunniteltu työvuoro (PlannedShift)**

Mallina sama kuin luokka "ht.haapala.wss.data.PlannedShift". Sisältää suunnitellun työvuoron tiedot.

### **Lista suunnitelluista työvuoroista (PlannedShiftList)**

Kokoelma malleja PlannedShift.

### **Käyttäjä (User)**

Mallina sama kuin luokka "ht.haapala.wss.data.WSSUser". Sisältää käyttäjän tiedot.

### **Käyttäjälista (UserList)**

Kokoelma malleja User.

## ***Backbone-näkymät (views)***

Backbone-näkymät esittävät pääasiassa mallien tarjoamia tietoja käyttäjälle (MVC-malli). Nämä mallit löytyvät tiedostosta "webapp/resources/js/wss-view.js".

### **Kirjautumisnäky (LoginView)**

Kirjautumisnäky hallitsee oikeassa yläkulmassa sijaitsevaa kirjautumislomaketta. Näky käyttää vain mallia LoggedUser. Jos kirjautunutta käyttäjää ei ole näytetään kirjautumiskentät. Jos kirjautunut käyttäjä löytyy, näytetään teksti "Logged in as username" ja uloskirjautumispainike.

### **Navigointinäky (NavigationView)**

Navigointinäky hallitsee päävalikkoa. Kirjautuneen käyttäjän roolin mukaisesti näytetään valikossa toimintoja: työnantajalle viikkonäkymän lisäksi työvuorojen ja käyttäjien hallinta.

### **Vuoronäky (ShiftView)**

Vuoronäky hallitsee vuorojen hallintasivua. Käyttää mallinaan listaa työvuoroista, jotka näky näyttää sivulla. Hallitsee myös työvuorojen lisäyksen ja poiston näkymässä.

### **Viikkonäky (WeekView)**

Viikkonäky hallitsee pääviikkonäkymää. Käyttää perustoiminnallisuuteen kirjautuneen käyttäjän mallia näyttämään joko perusnäky tai suunnittelunäkymän roolin perusteella. Suunnittelunäky näytetään vain työnantajille. Tiedot näkymään haetaan malleista työvuorolista ja lista suunnitelluista työvuoroista. Mallia käyttäjälista käytetään vain jos suunnittelunäky on käytössä. Tällöin haetaan lisäksi tiedot käyttäjistä, jotka näytetään jos vuoroon ollaan lisäämässä työntekijää.

### **Käyttäjänäky (UserView)**

Käyttäjänäky hallitsee käyttäjien hallintasivua. Käyttää mallinaan listaa käyttäjistä, jotka näky näyttää sivulla. Hallitsee myös käyttäjien lisäyksen ja poiston näkymässä. Käyttäjät erotettu työnantajiin ja työntekijöihin.

### 3.3 Java

Palvelin pohja on tehty käyttämällä Spring-sovelluskehystä. Spring tarjoaa helpot työkalut web-sovellusten luomiseen. Springin ORM-tuen avulla on data-pakkauksessa määritelty luokat (WSSUser, WorkShift, PlannedShift), jotka "vastaavat" tietokantataulujen rivejä. DAO-rajapinnan tarjoavat repository-pakkauksen luokat (alin taso sovelluksessa). Service-pakkauksen luokat "hiovat" repository-tason palvelut haluttuun muotoon. Controller-pakkauksen käyttävät service-luokkia tarjoamaan palvelimen JSON-rajapinnan selaimen käyttöliittymälle.

Tarkemmat kuvaukset luokista ja niiden toiminnasta löytyy JavaDoc-dokumentaatiosta. Tässä yleistä tietoa.

#### ***Tarkistuksista***

Sovelluksessa on tietyissä kentissä tarkistuksia. Nämä tarkistukset ovat sekä JavaScript-, että Java-puolella.

WSSUser: käyttäjätunnus oltava vähintään 3 merkkiä pitkä, sähköpostiosoitteen oltava vähintään muotoa a@a (tämä tarkistus vain selainpuolella).

WorkShift: vuoron nimen oltava vähintään 3 merkkiä pitkä. Alkamis- ja päättymisaikojen oltava muodossa hh:mm.

#### ***Data-lisäluokista***

Luokka BareUser sisältää vain yhden parametrin, käyttäjätunnuksen. BareUseria käytetään piilottamaan liika tieto käyttäjistä suunniteltujen työvuorojen JSON-palautuksissa.

Perustoiminnallisuudella kaikki käyttäjän tieto siirtyisi jokaisen suunnitellun työvuoron mukana roolitietojen jne. kanssa. Nyt käyttäjästä paljastetaan vain käyttäjätunnus.

Luokka Week sisältää viikon tiedot. Selain ei laske päivämääriä tai nykyistä viikkoa, kaikki pyörii palvelimen aikataulussa. Palvelin tarjoaa päivämäärätiedot viikoista ja selainpuolella vain kasataan tieto haluttuun muotoon.



## 4 Asennustiedot

Sovellus vaatii toimiakseen Servlet-alustan (esim. Tomcat) ja MySQL-tietokannan. Oletusasetuksilla sovellus yhdistää MySQL-palvelimen 127.0.0.1 porttiin 3306 tunnuksella heha ja salasanalla heha, näiden vaihtaminen käydään läpi myöhemmin. Sovellus käyttää tietokantaa wss.

### Ohjeet Tomcatille

Pakattu sovellus (wss.war) siirretään Tomcatin webapps-kansioon ja Tomcat käynnistetään, Tomcat purkaa oletuksena war-paketin webapps-kansion alikansioon wss. Jos haluat vaihtaa MySQL-palvelimen asetuksia, sulje Tomcat, muuten sovellus pitäisi olla käytettävissä Tomcat-palvelimesi osoitteessa (alipolku wss).

Vaihtaaksesi MySQL-palvelimen asetuksia, avaa tiedosto "wss/WEB-INF/classes/jdbc.properties". Voit vaihtaa parametreja haluamaksesi: jdbc.url sisältää palvelimen osoitteen, portin ja tietokannan nimen muodossa mysql://host:port/database. jdbc.username on tietokannan käyttäjätunnus ja jdbc.password on käyttäjätunnuksen salasana.

Sovellus luo tietokannan taulut ensimmäisellä käynnistyskerralla jos tauluja ei ole olemassa. Kaikki taulut luodaan tyhjiksi ja niihin ei rivejä lisätä automaattisesti. **Pääkäyttäjä on luotava manuaalisesti: MySQL-tietokantaan on otettava yhteys ja tietokannassa on suoritettava tiedostossa "wss/WEB-INF/classes/sql/data.sql" oleva kysely, joka lisää käyttäjän admin salasanalla admin.** Tämän voi tehdä sovelluksen ollessa päällä.

## 5 Käynnistys- ja käyttöohje

Sovellus on käytettävissä osoitteessa <http://t-heha.users.cs.helsinki.fi/wss>. Pääkäyttäjän tunnus on admin ja salasana admin. Käytön pitäisi olla intuitiivista.