System Guideline

# A. Deployment guidelines

## 1. Preparations

* JDK 8 + NetBeans 8.2 – to build the project
* Apache Tomcat 8.5 – to host the web application

**2. Steps**

### 2.1. Configure project

1. Open project with NetBeans.
2. Open context.xml and configure the data source for the application:
   * url: The JDBC database URL
   * username: The username to connect to database
   * password: The password to connect to database
3. Open hibernate.cfg.xml, then set the *hibernate.hbm2ddl.auto* property to *create* or *create-drop*. That will allow the creation of database schema, and should be used when running the project for the first time only. After that, the property should be deleted, or set to *validate*.
4. Run the project to test if deployment was successful and the database was initialized.

### 2.2. Deploy project

1. Build the project with NetBeans to generate a .war file in *dist* folder.
2. Copy the .war file into *webapps* folder of Tomcat.
3. Copy sqljdbc42.jar into *lib* folder of Tomcat (if not exist).
4. Start Tomcat server.

***Note****:*

* If there is any kind of change, remember to delete the old .war file and update the new one into folder *webapps* of Tomcat.
* If there is problem about Hibernate dependency, manually copy Hibernate 4.3.11 (version with JPA support) jar files to lib folder of the project, then add reference to these files to the project.

# B. Project structure

## 1. Web pages

* All jsp files (the views for the controllers) are in *web/WEB-INF/jsp*
* All jspf files (the fragments used in templating) are in *web /WEB-INF/jspf*
* The custom tags used in jsp files are in *web /WEB-INF/tags*
* The template tag library descriptor are defined in *web/WEB-INF/tlds*
* All frontend plug-ins used in OES are in *web/plugins*
* All .css, .js and image files are in *web/resources*

## 2. Source

* The *controller* package contains all the controllers of the project. Each sub-package contains a group of related controllers.
* The *model* package contains the entity models used in this project.
* The *dao* package provides classes to manipulate database objects with Hibernate.
* The *util* package provides utility functions, mainly:
  + Hashing
  + Google OAuth
  + Hibernate initialization and transaction
  + ManagedServlet used by all the controllers
* The *app* package contains the servlet context listener.
* The *filter* in filter package is used to update the Account object of the current user to the session.

## 3. Configuration

* Hibernate configuration: *src/java/hibernate.cfg.xml*
* Web Application Deployment Descriptor: *web/WEB-INF/web.xml*
* Tomcat Deployment Descriptor: *web/META-INF/context.xml*

# C. Other notes:

* The default administrator account is:
  + Username: admin
  + Password: admin