# 1. Introduction

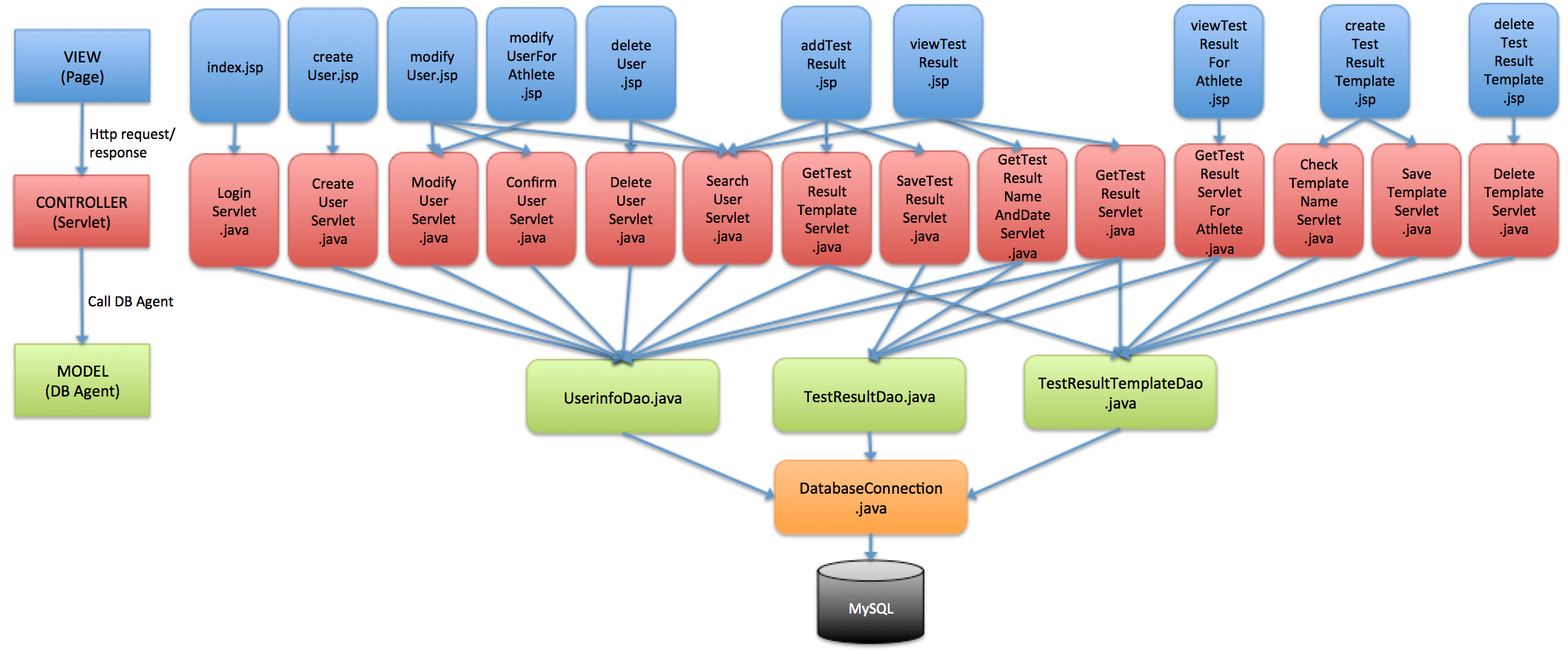
Peak centre project is a web-based system used for administrators and athletes to deal with the training and test data. The main functions of the web application are:

* Add, Modify and View Athletes’ test results.
* Add, Modify and View Athletes’ training plans.
* Add and View Athletes’ workout summaries.
* Create, Modify and Delete user accounts.
* Create and Delete test result templates.

The web application was designed to match Model-Controller-View architectural pattern and it was written in jsp, javascript, css for view module and java for the model and controller module.

# 2. Application Architecture

The following figure shows the main architecture (MCV model) of the Peak centre project:



# 3. Application Component

## 3.1 Development Environment

* Java version 1.8.0\_05
* Eclipse Java EE Kepler Service Release 2
* Tomcat 7.0.61
* MySQL 5.6.20

**3.2 User Types**

* Administrators: They have the highest authority of the system.
  + Add, modify, view anyone’s test result.
  + Add, modify, view anyone’s training plan.
  + View anyone’s workout summary.
  + Create, modify, delete anyone’s user account.
  + Create, delete test result template.
* Coaches: They have less authority than administrators (Then cannot delete anything).
  + Add, modify, view test result of those who come from the same city.
  + Add, modify, view training plan of those who come from the same city.
  + View workout summary of those who come from the same city.
  + Create, modify user account of those who come from the same city.
  + Create test result template.
* Athletes: They can only access to their own data.
  + View their own test result.
  + View their own training plan.
  + Add, view their own workout summary.
  + Modify their own user account information.

## 3.3 Views

Views are jsp files that transfer user requests to the application and send responses back to the user browser. Here are the all jsp files:

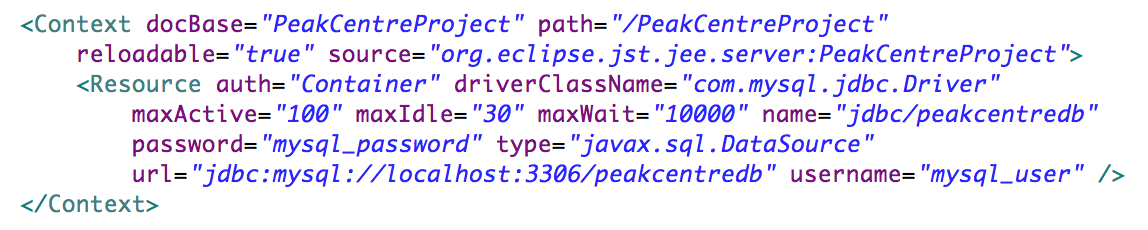
* Login
  + *index.jsp*: Enter username(email), password and choose user type to login.
* Main Page
  + *dashboard.jsp*: The main page of the system.
* Test Result
  + *addTestResult.jsp*: Choose a template and add a test result for a user.
  + *modifyTestResult.jsp*: Modify the data of a test result for a user. (Not finished)
  + *viewTestResult.jsp*: (For administrators and coaches) View a test result of a user.
  + *viewTestResultForAthlete.jsp*: (For athletes) View athlete’s own test result.
* Training Plan (Not finished)
  + *addTrainingPlan.jsp*: Add a training plan for a user.
  + *modifyTrainingPlan.jsp*: Modify the training plan for a user.
  + *viewTrainingPlan.jsp*: View a training plan of a user.
* Workout Summary (Not finished)
  + *addWorkout.jsp*: Add athlete’s own workout summary.
  + *viewWorkout.jsp*: View the workout summary of a user.
* User Account
  + *createUser.jsp*: Create a new user account.
  + *modifyUser.jsp*: (For administrators and coaches) Modify an existing user account.
  + *modifyUserForAthlete.jsp*: (For athletes) Modify athlete’s own account information.
  + *deleteUser.jsp*: Delete a user account.
* Test Result Template
  + *createTestResultTemp*: Create a new test result template.
  + *deleteTestResultTemp*: Delete a test result template.

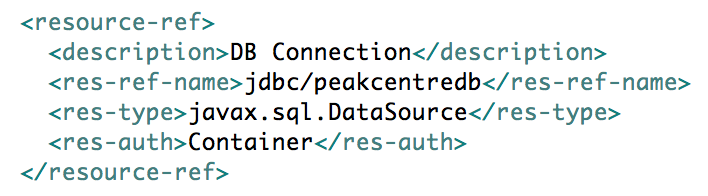
## 3.4 Controllers

* Login
  + *LoginServlet.java*: Get username, password and usertype from index.jsp, check if user account exists and redirect to the main page.
* Logout
  + *LogoutServlet.java*: Clear session and redirect to index.jsp.
* Test Result
  + *GetTestResultNameAndDateServlet.java*: Get username from viewTestResult.jsp and return the user’s test result names and dates.
  + *GetTestResultServlet.java*: Get username, test result name and date from viewTestResult.jsp, read test result xml template, get test result data from database and show result back to viewTestResult.jsp.
  + *GetTestResultServletForAthlete.java*: The same as above but get username from viewTestResultForAthlete.jsp, used only for athlete user.
  + *SaveTestResultServlet.java*: Get data entried from addTestResult.jsp and insert into database.
* Training Plan (Not finished)
* Workout Summary (Not finished)
* User Account
  + *CreateUserServlet.java:* Get new user account information from createUser.jsp and insert into database.
  + *ComfirmUserServlet.java*: Get the chosen user from modifyUser.jsp and return the chosen user’s account information.
  + *ModifyUserServlet.java*: Get updated user information from modifyUser.jsp or modifyUserForAthlete.jsp and update database.
  + *DeleteUserServlet.java:* Get chosen user from deleteUser.jsp and delete the chosen user’s account from table userinfo.
  + *SearchUserServlet.java*: Get user’s first name and last name from modifyUser.jsp, deleteUser.jsp, addTestResult.jsp or viewTestResult.jsp, check if user exists. If there is one or more users exist, return user information list.
* Test Result Template
  + *CheckTemplateNameServlet.java*: Getthe template name from createTestResultTemp.jsp and check if the template name is empty or if the tamplate name already exists.
  + *DeleteTemplateServlet.java*: Get template name from deleteTestResultTemplate.jsp and delete data from table TestResultTemplate and tables of template.
  + *GetTestResultTemplate.java*: Get template name from addTestResult.jsp and, read template xml file and show template back to addTestResult.jsp.
  + *SaveTemplateServlet.java*: Get template elements from createTestResultTemplate.jsp, create xml template file and create template tables in database.

## 3.5 Model

### 3.5.1 Database Connection Pool

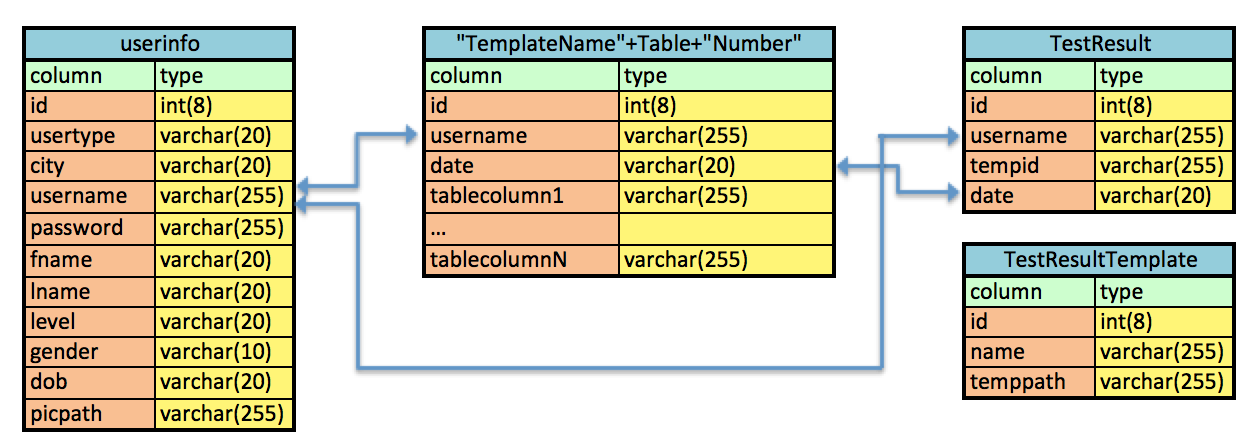
* Tomcat server.xml
* WEB-INF web.xml:



### 3.5.2 Database Agents

* *DatabaseConnection.java*: Get database connection and disconnection.
* *UserinfoDao.java*: Database manipulation of user account information.
* *TestResultDao.java*: Database manipulation of test result information.
* *TestResultTemplateDao.java*: Database manipulation of test result template information.

### 3.5.3 Database Schema



**3.6 Entity**

* *TestResult.java*: Map table TestResult.
* *TestResultTemplate.java*: Map table TestResultTemplate.
* *Userinfo.java*: Map table userinfo.

**4. Dependencies**

In order to operate the project, there are a view libraries should be installed first:

* commons-fileupload-1.3.1.jar
* mysql-connector-java-5.1.32-bin.jar
* servlet-api.jar

**5. Remaining Work**

There are still many work left. Here is a short list need to be finished later.

* The Modify Test Result function in Test Result part.
* The whole training plan part.
* The whole workout summary part.
* Add printer icon on every page to let user print.
* If directly open one page(not login page), need to login first.
* Set session time.
* If user did not upload profile picture, use a default one.
* Validate profile picture before upload (picture must be in jpg format).
* Write all words shown on web page in property file (peakcentre.properties)
* Delete user account must also delete user’s test results, training plan and workout summary.
* In create test result template, column names in a table must not be repetitive.
* In add test result, add option to let user add a row to the table or remove a row from the table.