Homework 3: Make an MVC Webapp

For this homework, you will expand upon your previously created html/servlets to make your webapp actually do something useful. You will also gain more practice using Subversion (svn).

As before, you will do this homework as a team; however, each member of your team will be responsible for the completion of a particular task.

Step 0. Merge in the provided classes/JSPs

In your team's SVN repository, look in the **branches** subfolder of the **xleague** project folder. You will find a folder named **hw3-skeleton**, which contains a partial webapp project. Copy the following files from this project skeleton into your **trunk**:

- All Java classes (both **controller** and **model** packages)
- All JSPs (find them in the **WEB-INF** folder)
- The new **index.html** file.

It's probably best if one person does the above, commits the changes to the **trunk**, and then has everyone else update their working copies.

Step 1. Create MVC-style servlets and JSPs

For this homework, your main task will be to reverse engineer servlets/JSPs that implement portions of this webapp—try it!:

http://utopia.cs.memphis.edu:8080/xleague/

In particular, you must create two servlets and one JSP that follow the MVC pattern. Your code must behave in the same manner as the above webapp. Use the diagram below to help guide what each servlet/JSP should do and how it fits in with the code you copied into your **trunk**. Here are the specific tasks:

- Task 1
  - JSP: /WEB-INF/players/updateForm.jsp
  - Servlet: PlayersUpdateFormServlet
  - Servlet: PlayersUpdateServlet
- Task 2
  - JSP: /WEB-INF/players/deleteForm.jsp
  - Servlet: PlayersDeleteFormServlet
  - Servlet: PlayersDeleteServlet
- Task 3
  - JSP: /WEB-INF/teams/updateForm.jsp
  - Servlet: TeamsUpdateFormServlet
  - Servlet: TeamsUpdateServlet
- Task 4
  - Form: /WEB-INF/teams/deleteForm.jsp
  - Servlet: TeamsDeleteFormServlet

    - Servlet: TeamsDeleteServlet
- Task 5
    - JSP: /WEB-INF/teams/addPlayerForm.jsp
    - Servlet: TeamsAddPlayerFormServlet
    - Servlet: TeamsAddPlayerServlet

Each team member must do one of the above tasks, and all team members must do different tasks. If your team has only four members, then ignore Task 5.

Here are some constraints you must follow as well as some tips for success:

- Do not modify the code you copied into your project in Step 0. Your new code should mesh perfectly with this existing code.
- Use the DAO classes to perform CRUD operations on domain objects, such as player and team.
- Only servlets may interact with DAO objects. The JSP code should include no mention of the DAOs.
- Look at the provided servlets/JSPs for examples.
- In the diagram below, pay close attention to the paths/URLs and how control/data moves between the components (e.g., HTTP GET/POST, forwarding, redirecting).
- Note that some of the GET requests require an player or team ID to be included as a parameter in the URL.
- Use the exact names given in the diagram – names are case sensitive!

Step 2. Submit (by tagging) your team's submission

Attention! Before performing this step, you must make sure that all team members have committed their edits to the **trunk** in the repository.

Only one team member (the leader) performs the following.

First, you must fill out the **README.txt** file in your project's **trunk**. The file should list which team member performed each task (one team member per task; use their login IDs).

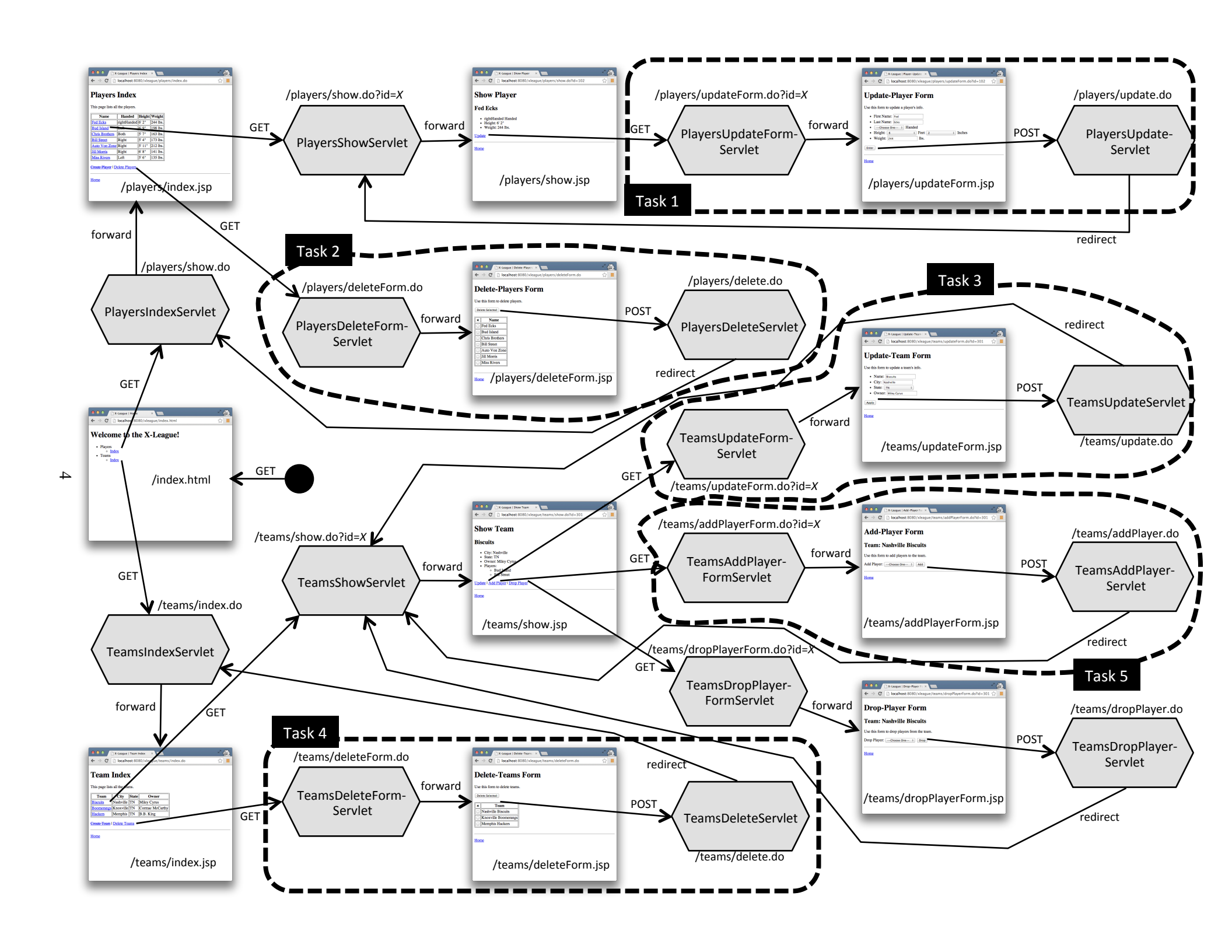
To submit work in this course, you must tag it. Then, I will checkout the revision that you tagged and grade it. By tagging, you tell me that you are done, and this is the version you want me to grade.

The tag you must use for this homework is **hw3** (case sensitive, no spaces).

To tag the current revision of your trunk as **hw3**, do as follows:

1. Go to the **SVN Repository Exploring** perspective in Eclipse.
2. In the **SVN Repositories** view, find the **trunk** folder that you want to tag.
3. Right-click on the **trunk** folder, and click **Show History**. This should open the **History** view with a table listing the past commits to the **trunk**.
4. In the History table, right-click the newest revision (i.e., the one with the greatest revision number), and click **Tag from…** This should open a **Create Tag** dialog.
5. Enter **hw3** into the **Tag** field and optionally enter a log comment, then click **OK**. This should create the tag!

To verify that tagging was successful, open your team's repository URL in a web browser, and inspect what's in the repository.


Players Index
/players/index.jsp
GET
/players/show.do?id=X
PlayersShowServlet
forward
Show Player
Fed Ecks
/players/show.jsp
Task 1
/players/updateForm.do?id=X
PlayersUpdateForm-Servlet
Update-Player Form
/players/updateForm.jsp
POST
/players/update.do
PlayersUpdate-Servlet
redirect
forward
/players/show.do
PlayersIndexServlet
Task 2
/players/deleteForm.do
PlayersDeleteForm-Servlet
Delete-Players Form
/players/deleteForm.jsp
/players/delete.do
PlayersDeleteServlet
Task 3
Update-Team Form
/teams/updateForm.jsp
/teams/update.do
TeamsUpdateServlet
TeamsUpdateForm-Servlet
/teams/updateForm.do?id=X
Welcome to the X-League!
/index.html
Show Team
Biscuits
/teams/show.jsp
/teams/show.do?id=X
TeamsShowServlet
/teams/addPlayerForm.do?id=X
TeamsAddPlayer-FormServlet
Add-Player Form
Team: Nashville Biscuits
/teams/addPlayerForm.jsp
/teams/addPlayer.do
TeamsAddPlayer-Servlet
/teams/index.do
TeamsIndexServlet
/teams/dropPlayerForm.do?id=X
TeamsDropPlayer-FormServlet
Drop-Player Form
/teams/dropPlayerForm.jsp
/teams/dropPlayer.do
TeamsDropPlayer-Servlet
Task 5
Task 4
Team Index
/teams/index.jsp
/teams/deleteForm.do
TeamsDeleteForm-Servlet
Delete-Teams Form
/teams/deleteForm.jsp
TeamsDeleteServlet
/teams/delete.do