# Homework 4: JSPs and MVC

For this homework, you will practice writing Java EE web apps that have a model-view-controller (MVC) architecture.

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 1. Make GameDen pages do something

You will continue working on the **GameDen** pages that you began in Homework 2. In particular, each team must complete one of the tasks below. Each task describes a dynamic web page based on the forms you created in Homeworks 2.

You must use an MVC architecture in implementing your task (i.e., using a combination of JSPs, servlets, and POJOs).

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

*The following instructions are essentially the same as last time; only the tag name has changed.*

**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).

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 **hw4** (case sensitive, no spaces).

To tag the current revision of your trunk as **hw4**, 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 **hw4** 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 the following URL in a web browser (replacing *YOUR_TEAM* with the appropriate name):

https://utopia.cs.memphis.edu/course/comp7012-2013spring/teams/*YOUR_TEAM*/GameDen/tags/

You should see an **hw4** folder, and within that folder should be **src** and **WebContent** folders along with the **README.txt** file. Everyone's HTML files should be in the **WebContent** folder.

## The Tasks

### Task 1: User Info Form

- The system must store the information submitted in the form (in memory).
- Submitting the form must bring up a user-info page that displays the information entered.
- The user-info page must have an edit button that brings up the user-info form.
- The user-info form must be populated initially with any data that the user previously submitted.

### Task 2: Blackjack Game

- The dynamic web page must enable the user to play a hand of blackjack against the computer. See http://en.wikipedia.org/wiki/Blackjack for rules.
- Initially, only one of the computer's cards is revealed. Display "???" where the other card will eventually be displayed.
- Initially, the player has two cards.
- When the user submits a "Hit me!" move, a card is added to his hand.
- If the total of the user's cards exceeds 21, the game displays a "You Lose" message and reveals the computer's hidden card.
- When the user submits a "Hold" move, the computer draws cards until its hand exceeds that of the player.
- If the computer's hand exceeds 21, then the player wins, and a "You Win!" message is displayed (along with all the computer's cards).
- If the computer's hand is less than 21, then the player loses, and a "You Lose" message is displayed.
- When a game has ended, there should be a "Try Again" button that starts a new hand.

### Task 3: Minefield Game

- The dynamic web page must enable the user to play a game of minesweeper. See http://en.wikipedia.org/wiki/Minesweeper_%28video_game%29 for more on the game.
- Each cell on the board may or may not contain one of 10 randomly placed mines. The location of the mines is hidden from the user.
- For each move the user applies, he/she may place a flag on a cell (which indicates that he/she thinks there is a mine there), or he/she may explore a cell.
- Flagging a cell places the word "Flagged" in the cell along with an "Unflag" option that the user can apply to remove the flag.
- Exploring a cell has one of three effects:
  - If the cell contains a mine, then the player loses the game, a "You Lose" message is displayed and the contents of all cells on the board is revealed.
  - If the cell does not contain a mine, then a number is placed in the cell that is the sum of cells adjacent (left, right, up, down, and diagonals) to the explored cell that contain mines.

    - If the cell does not contain a mine, and the all unexplored cells remaining contain mines, then the player wins the game, and a "You Win!" message is displayed.
- The Restart button enables the user to start a new game.

### Task 4: Tic-Tac-Toe Game

- The dynamic web page must enable the user to play a game of tic-tac-toe against the computer. See http://en.wikipedia.org/wiki/Tic-tac-toe for more on the game.
- The computer places moves randomly.
- Each time the user submits a move the game places an "X" in the cell the user chose and also places an "O" marking the computer's move.
- If the user's move wins the game, then a "You Win!" message is displayed (and the computer does not make a move).
- Clicking the "Quit" button restarts the game.

### Task 5: Yahtzee Game

- The dynamic web page must enable the user to play a round of Yahtzee. See http://en.wikipedia.org/wiki/Yahtzee for more on the game.
- Initially a roll of 5 dice is displayed, with the results for each dice being displayed as a row of bullets.
- The user may select dice from the roll to hold and then reroll the remaining dice.
- Including the initial roll, the user gets a total of 3 rolls (i.e., two rerolls).
- After the 3rd roll, the page must display the users score (see the above web page for scoring rules).
- The Restart button restarts the game.

### Task 6: Battleship Game

- The dynamic web page must enable to play a modified game of Battleship. See http://en.wikipedia.org/wiki/Battleship_%28game%29 for more on the game. The modification will be that the user is only attacking the computer's ships and has no ships of his/her own.
- Initially, the computer places the following ships on the board (non-overlapping), and because the ships are hidden, an empty board is displayed to the user.
    - Aircraft Carrier – Size 5
    - Battleship – Size 4
    - Submarine – Size 3
    - Destroyer – Size 3
    - Patrol Boat – Size 2
- Each turn, the user fires at coordinates on the board. If part of a ship is hidden in the cell, then the user scores a hit, and an X appears in the cell. If no ship is in the cell, then a bullet appears in the cell.
- Hitting all cells associated with a particular ship causes a message "You sank my *X*" to appear such that *X* is the name of the ship.
- Once the last ship is sunk, the page displays the message "Congratulations! You sank all ships in *Y* turns" such that *Y* is the number of times the user fired.
- The Start Over button causes a new game of Battleship to start.