HW1: User Stories, Planning, and Setup

In this homework, your team will begin a first iteration on the project. In particular, you will

- Gather some requirements
- Plan work for some of the requirements
- Rough out a web-app code skeleton

Requirements and planning

There will be a requirements workshop in class on Monday 6 February in which you will meet with the client (me) and gather the requirements. Details of the project will be revealed in class.

Based on the workshop, you will create a set of user stories.

For the highest priority user stories (as selected by the client—me!), you will break the user stories into tasks and estimate the work required. Using these estimates, you will plan how to complete the iteration (HW2).

All user stories and task descriptions will be submitted on paper "post-it" notes. The exact requirements will be explained in class.

Coding

To create a web-app skeleton, your team must implement a sort of "hello world" app that follows an MVC design (plain-Java model, JSP view, Java-servlet controller). The model must be a single class. Each teammate will be responsible for parts of each architectural component as well as a JUnit test case (see Productivity details below).

To collaborate in this development task, remove the old code from your project's trunk, and commit a new project. Keep Eclipse config files out of the repository (i.e., .settings/, .project, .classpath). We discussed how to do this in class, but if you have questions, let me know.

To submit, one team member should tag the trunk as HW1 the same way you did for HW0.

Productivity

Regular

To earn the regular productivity points, you (individually) must do the following. Make sure that you label each item with the main author's name (one person only).

- Requirements and planning:
  - Write at least three user stories.
  - Document the set of tasks for completing one user story.

- Coding:
  - Write a model method.
  - Write a servlet that uses your method.
  - Write a JSP to which your servlet dispatches requests.
  - Write a JUnit test for your model method.

Above and beyond

Here are some ideas for above and beyond work:

- [1pt] Write three additional brief USs (one point max per person).
- [1pt] Sketch out a domain model (UML class diagram + glossary) for your system.
- [1pt] Sketch out some substantial design diagrams for your system. (Probably include a glossary.)
- And the tutorial ideas from HW0 are still fair game.

Of course you may always propose work that is not on the list. As always, I must approve all A&B work.