Assignment 5 – Design

Due

24 hours prior to your lab in the week of October 20th, 2014 for everything except the peer evaluation. Late assignments will not be accepted and the whole team will receive 0 for the assignment. No exceptions will be made.

Peer evaluation is due on the day of your lab in the week of October 20th, at midnight (after your lab).

Objectives

The purpose of this assignment is to familiarize you with the architecture you will be working with for Sprints 1 and 2 (by revisiting the StockWatcher system), and also to get you to start thinking about how the StockWatcher architecture relates to the system you plan to build. This will prepare you for more detailed design at the start of Sprints 1 and 2.

Procedure – Design

STEP 1 (checked by your TA in lab) - Reverse-Engineer the StockWatcher System Design:

Take a look at your code for the full version of the StockWatcher system that you completed for Assn 1. Sketch out the components and classes you can identify in the system. Take time to understand how the components and classes communicate with one another. Running the code might assist with this. Changing things and seeing what happens might also help. Ask your TA for advice during the lab. By the end of this process, **you should have a high-level architectural diagram (hand drawn is ok but it should be readable) that shows (your preliminary understanding of)**:

- **Architectural Components and their overall responsibilities**
- **Classes, their responsibilities, and in which architectural component they sit.**
- **Association relationships between classes, and generalisation (subclassing) between classes**
- **Communication relationships between components and classes, with brief descriptions**

There are existing UML diagrams for the StockWatcher system (linked from the milestone page) to which you can refer. Please do so, but note that they do not contain enough explanation. You will need to investigate further to achieve your high-level diagram.

STEP 2

- **FOR GROUPS USING GWT - Change the StockWatcher Design to suit your own application's needs:** look at the diagram you ended up with (tweak it if you want), and consider how your own Sprint 1 application design fits into that architecture. You will need to add and change classes for your own purposes.
- **FOR GROUPS NOT USING GWT - Take inspiration from the StockWatcher Design:** your technological solution might have a completely different architecture than GWT. Come up with a design that is adapted to the framework you chose.

Draw your results the same way you did for Step 1. This diagram isn't really marked - it's just there or not - but it will be a valuable starting point for working together on what to build, so the more investigation you do the richer your preparation will be. In addition come up with a 1-2 paragraph design rationale explaining your design.

Remember, design is an iterative process and getting it right from scratch is very unlikely, so you will probably have several diagrams drawn by the end of this assignment. There should be quite a bit of discussion going on between you and your teammates and there should be refinement steps. Don't be scared to make changes! In later stages of the project you might also adapt your design, which is encouraged.

Deliverables

1. For Step 1: during the lab explain your high-level architecture diagram with your TA.

2. For Step 2: 24 hours prior to your lab in the week of October 20th you must submit via handin :

- your diagrams representing the design of your project (pictures and scans are okay if they are clearly readable)

- the short design rationale explaining your design

3. Peer evaluation (due on the day of your lab in the week of October 20th, at midnight after your lab)

- Each of you must submit the peer evaluation form via handin (don't forget to include your team name in the evaluation). You can find the form at http://www.ugrad.cs.ubc.ca/~cs310/peerEval.pdf.

Handin

How to use handin for your diagrams:

- Put the files in your `~/cs310/a5-X` directory (where you must replace X with your team name according to the list of groups I posted on Piazza)
- Run the command `handin cs310 a5-X` (again, replace X with your team name)

How to use handin for your peer evaluation:

- Put the file in your `~/cs310/a5-X-peereval` directory (where you must replace X with your team name according to the list of groups I posted on Piazza)
- Run the command `handin cs310 a5-X-peereval` (again, replace X with your team name)

Other notes about handin

1. You can confirm that your assignment was handed in correctly by using the `-c` flag (ie, `handin -c cs310 a5-X-peereval`)
2. If you need to overwrite a previous version that you handed in, using the `-o` flag (ie, `handin -o cs310 a5-X-peereval`) Assignments will be accepted by handin up to the due date.
3. If you are using the web handin, you must zip your assignment first because it will only accept zip files.

Grading Scheme:

1. (10%) Peer Evaluation (completing the evaluation and giving helpful comments)

2. (90%)
   a. Step 1 : participation mark
   b. Step 2 : design Diagrams with the traits listed in the Step 1 and 2 directions + Design rationale.

   NOTE: the grade for this part will be adapted according to the peer evaluations.

References

We have created quite a few sample design diagrams for you, based on the Stockwatcher project. Comparing these diagrams with the projects will help you understand what is going on in the Stockwatcher design. These diagrams are posted on the project webpage. In particular, I recommend carefully looking at how the Stockwatcher example diagrams change through the different versions. The Stockwatch from AppEngine tutorial diagram is a good model for your design as your projects will likely be structured in a similar way. Note that this diagram is more complete than the diagram you are required to submit.

http://www.ugrad.cs.ubc.ca/~cs310/project/Stockwatcher%20from%20Initial%20Tutorial.png

http://www.ugrad.cs.ubc.ca/~cs310/project/Stockwatcher%20from%20GWT-RPC%20Tutorial.png

http://www.ugrad.cs.ubc.ca/~cs310/project/Stockwatcher%20from%20AppEngine%20Tutorial.png