CSE 403, Winter 2007
Homework #3: Software Design Specification ("SDS"); 50 points

Due: Friday, February 9, 2007
(at start of lecture, by hand and/or electronically)

Assignment Description:

The second milestone you will submit for this project is a set of documents about your project's design. Your design should specify information about how to implement an object-oriented software product that meets the requirements laid out in your SRS. Your SDS should answer the following three questions: what are your classes, what are the responsibilities of each class, and how do these classes collaborate?

This project description can be considered a partial design request specification, but some information is left out. This is to encourage your group to come up with questions to ask "the customer." You should come up with some specific questions to ask, either in lecture, by email, or on the course message board. A submission that does not reflect questions and answers with the "customer" will not receive full credit.

Documents Required:

Your SDS must contain the following three (3) types of documents:

A. UML class diagram

Submit a UML class diagram for your system in the proper format as explained in Fowler, Chapter 3. Your class diagram should display all major classes in the system, each class's attributes (fields), methods (do not list get / set / is methods), inheritance and interface relationships between classes, and named directed associational relationships with multiplicity adornments between classes that collaborate.

Your design will be evaluated on completeness as well as level of thought, and attention to principles discussed in class. You should follow Riel's OO design heuristics from the reading, such as:

- use encapsulation (Heuristic 2.1)
- keep related data and behavior in the same place (Heuristic 2.9)
- minimize each class's public interface (Heuristic 2.3, 2.6)
- emphasize cohesion and limit coupling (Heuristic 2.7, 2.8)
- avoid "god classes" (Heuristic 3.2)
- avoid insignificant or irrelevant classes (Heuristic 2.11, 3.7, 3.8)
- model-view separation and model independence from view (Heuristic 3.5)
- avoid irrelevant "agent" or "controller" classes (Heuristic 3.10)

Your design should also provide good horizontal distribution of your project's functionality and allow for features to be developed in parallel, as much as possible.

If your project is web-based or not object-oriented, your "classes" may actually represent pages, files or modules. You can still diagram their names, relevant attributes (state), and behavior and/or connect the modules that associate with each other. Please speak to the "customer" to resolve any issues regarding how to fit your project to this sort of diagram.

B. Two (2) UML sequence diagrams

Submit two UML sequence diagram that depict your product executing two of its important use cases. These can be the same use cases you wrote about in your SRS. The sequence diagrams should follow the format of the examples from Fowler, Chapter 4. Your diagram should show all participants (objects) in the sequence, all important directed messages between them and their return values (if any), as well as interaction frames with proper operator adornments as appropriate. You should demonstrate good design by showing decentralized control in your sequence diagrams; no one class or object should do the bulk of the work.

If your project is web-based, the sequence diagrams are especially important because they show the "life" of a user's web request. Show the request's path through your UI, server, and/or data layers as it interacts with each to accomplish the overall task.

Accompanying one or both of the sequence diagrams should be a pseudo-code description of the same algorithm, similar to Fowler's Figure 4.4.

C. Feature-specific schedule outline

Your next two submissions after your SDS will be a prototype version of your code, and a more quality-tested, feature-complete version of your code. Each release will occupy roughly half of the remaining time of the rest of the quarter. Because of time and other constraints, it is important to plan out what work will be done in each release and at what level of quality.

Please submit a brief list (either as a table, bulleted list, in paragraph form, etc.) of what features you expect to have finished in your prototype vs. in your final version, and at what level of quality. (e.g. Will it be bug-free? Will it be robust against errors? Will the feature be complete, or will parts be missing? etc.) For each feature, list its approximate state in the prototype as well as in the final version. If necessary, note some features that will be early to be dropped or scaled back as a form of risk mitigation if schedules start to slip.

D. Presentation

Submit a set of 5-10 slides that represent a brief presentation of the above material to the rest of the class. Each group will be given approximately **9-10** minutes to present. At least two group members must participate in some way in the presentation, but not every group member needs to be involved in it.

Your presentation slides should summarize important parts of the preceding elements for your product. Specifically, talk about your design and briefly discuss at least one sequence diagram or use case. Include a title slide that states your project's name and authors. You should have at least two figures or diagrams in your slides to receive full credit; these can be taken directly from your other documents described previously.

Submission and Grading:

You may create your diagrams on paper or using software tools. We will accept `.violet` files, Office documents, PDF, or image files. Please ask us if you would prefer to use a different file format.

If you choose to turn in your SDS documents A-C electronically, please submit them in Word (.doc) or PDF format; otherwise turn them in as printed pages in lecture. Submit your presentation slides online in PowerPoint (.ppt) or PDF format. Each document's file name should begin with your project's name and reflect what it contains. For example, if your project is called "SuperAwesome", acceptable file names might be `SuperAwesome_class_diagram.pdf` or `SuperAwesome_presentation.ppt`. You may receive a deduction if you turn in clumsily named or organized files.

Make sure that your project's name and all group members' names appear clearly atop each document. Only one copy of the documents should be submitted for each group.