HW5 (team): Implementation Design

In this homework, you will design an implementation of your system. You will start with the architecture that you selected in HW4, plus the paper prototypes and requirements that you created in HW3.5.

Submit a PDF containing the following parts:

1. Draw a UML class diagram showing the key OO entities (and their attributes) in your system (approx 1 page); if the system is too large to fit all the UML classes on 1 page, then just focus on one particularly important part of the system
2. Assess your packaging scheme in terms of coupling and cohesion (approx 1 page)
3. Assess how well your design would support incremental or iterative development, based on the requirements and the potential for code reuse (approx 0.5 page)
4. Explain how one particular design pattern would be useful for implementing a portion of the system, and why (approx 1 page)
5. Select a use case from HW3.5 and draw a sequence diagram showing how that use case would play out in the system; this sequence diagram should differ from the sequence diagrams in HW3.5 in that the sequence for the current assignment should show how specific classes participate in supporting the use case (approx 1 page)
6. Identify one interface that would be needed and specify a contract for it (approx 1 page)
7. Identify one exception that is likely to occur and what the exception handler would do (approx 0.5 page)
8. Briefly summarize the contribution of each of your team members.

Some comments

Your work will be graded based on whether you appear to have produced an implementation design that contains all the parts identified above and that follows the principles described in Section 6.2. The page estimates above total 6 pages, but you may turn in up to 10 pages at your discretion.

You can divide this work however you like among your team, but here is a suggested approach that would complete the assignment very efficiently...

- Day 1, two teammates meet to draw the UML class diagram. They send the results by email to the team.
- Day 2, two teammates meet and figures out how to package the classes and how well the design would support incremental and iterative development. They send the results by email to the team.
- Day 3, two teammates meet and figure out which design pattern would be useful. They send the results by email.
- Day 3 (simultaneously), two teammates meet and draw a sequence diagram for a use case. They send results by email.
- Day 4, two team members meet and revise all of the preceding work so that it is more consistent. Preferably, one of these teammates will have worked on the initial UML class diagram. They send results by email.
- Day 5, two team members tack on the rest of the homework (to discuss an interface and exception), then do some final editing.