Home
Announcements
Piazza
Calendar
Textbooks
Syllabus
Assignments
Gradebook
Mailtool
Roster
Files & Resources
Site Settings
Evaluation
Help

Assignments

Viewing assignment...

Settings for "Project Design (2015 Team Assignment 4)"

Created by	Gail E. Kaiser
Date created	Oct 14, 2015 2:45 pm
Open	Oct 15, 2015 12:00 am
Due	Oct 27, 2015 11:55 pm
Accept Until	Oct 28, 2015 11:55 pm
Modified by instructor	Oct 15, 2015 10:32 am
Student Submissions	Single Uploaded File only
Number of resubmissions allowed	Unlimited
Accept Resubmission Until	Nov 5, 2015 11:55 pm
Grade	Points (max 20.0)
Alert:	Yes
Honor pledge:	No

Assignment Instructions

Get together with the other members of your entire team to design Version 0.1 of your application.

Use the noun/verb technique together with CRC cards to find potential classes, responsibilities and collaborators. Merge and split as needed. It is best to use real index cards, or pieces of paper approximately the size of index cards, to make sure you aren't including too much for a single class.

Then draw class diagrams describing the structural aspects of your design. Include visibility and multiplicity (note this was not required for the practice assignment, but is required for your project design). It is best to show the details of individual classes (attributes and operations) in one diagram with just that class, and use a separate diagram (here with just the class names for each class) to show the associations among classes. If the latter gets messy, then replicate the same class in multiple different diagrams where you show associations with different other classes. Just make sure all classes are associated with at least one other class, otherwise why would you need them? In general, the only attributes you need to show are those needed to "remember" state across multiple operations, e.g., to reference specific other associated objects.

Now draw sequence diagrams showing the behavioral aspects of your design, i.e., the run-time interactions among objects (instances of classes). There should be at least one sequence diagram for each use case. You only need to show the major interactions among and within classes. Make sure all the operations defined in the corresponding class diagrams are used, otherwise why would you need them?

Submit on courseworks (per team) a document with the following contents:

page 1: Team name, and full names and unis of all members of the team. This should be on a separate page, do not start the material for page 2 on the same page.

page 2: Write a header "Synopsis" at the top of page 2. Then briefly describe your idea for the application, including who are the target users and what value your application will provide to those users. You can copy/paste this from a previous team assignment if nothing has changed, but it might be wise to reduce the scope of what you expect to be able to demonstrate for Version 0.1.

pages 3 to N: Write a header "CRC Cards" at the top of page 3. Then transcribe your set of CRC cards.

pages N+1 to M: Make sure to start on a new page. Write a header "Class Diagrams" at the top of page N+1. Then draw your Class Diagrams.

pages M+1 to Q: Start on a new page. Write a header "Sequence Diagrams" at the top of page M+1. Then

draw your Sequence Diagrams.

Your document can be either MS Word, or equivalent (.doc), or Adobe pdf.

In addition, for each CRC card, class diagram and sequence diagram in your document, make sure there is a corresponding issue in JIRA. JIRA supports attaching images (for the diagrams) to issues, as well as cross-linking among issues (e.g., the CRC card should be cross-linked to any class and sequence diagrams involving that class). Revisit your user stories, use cases and technical tasks issues from the previous team assignments; have your user stories and use cases been downsized to match your design? Do the technical tasks cover the design work you've actually done? If not, update them. Ask Georgios for help if you have trouble working with JIRA; if you're doing something we didn't envision, he may need to modify the relevant issue definitions.

How do we know when we're done? When you have developed a design sufficiently complete, and understood by all team members, to enable any member of your team to implement Version 0.1 of your application by themselves (although in reality you will split the coding and testing assignments among pairs, rather than everyone implementing their own copy of the system).

After initial submission, your team should meet with your team mentor (your assigned IA) for informal feedback. Due to the university holiday (election day), the revision deadline will be pushed off for two days beyond the usual week. As usual, we will use the revision for grading; you are *not* required to submit a revision, in which case the original submission will be graded.