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

Assignments

Viewing assignment...

Settings for "Project Testing (2015 Team Assignment 6)"

Created by	Gail E. Kaiser
Date created	Aug 8, 2015 11:36 pm
Open	Nov 10, 2015 12:00 am
Due	Dec 3, 2015 11:55 pm
Accept Until	Dec 4, 2015 11:55 pm
Modified by instructor	Dec 16, 2015 5:40 pm
Student Submissions	Single Uploaded File only
Number of resubmissions allowed	Unlimited
Accept Resubmission Until	Dec 10, 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 implement and test Version 0.2 of your application.

Reconsider your collection of user stories and/or use cases, to see if there are any features that you put aside earlier that you would like to add, or if there is any existing functionality you would like to improve. Enhancing your project is optional, but might impress your ASE, LLC angel investor (i.e., your IA); in other words, you will receive extra credit for doing so, scaled according to the quality and utility of the improvements.

If you choose to add functionality, remember you need to update any effected user stories and/or use cases, and add corresponding technical tasks to design and implement. You will need to create/update your CRC cards, class diagrams and sequence diagrams accordingly. Make sure to include these in both for your document and in JIRA.

If you do not want to make any functional improvements or add any features, then your Version 0.2 can be the same as Version 0.1 - except it should be rigorously tested.

For each major class in your application (including any newly added classes), you will need a testing class with a set of test methods. There should be at least one test method for every business logic method in your class, typically more than one (it is very unlikely that your method has exactly one equivalence class and no boundaries, unless it takes no input and always does exactly the same thing). You do not need to test pure getters/setters, small helpers or UI-specific functions (e.g., Javascript). You do not need to test any framework or third-party code, but you do need to test your own code that uses it.

Create a test "issue" in JIRA corresponding to every major class, and describe there your strategy for testing the methods in that class. In particular, for each business logic method, define what are the e*quivalence classes* and *boundary cases* that your tests need to check and how they will do so.

Then write the corresponding testing code and run it using JUnit. Fix bugs and retest, rinse repeat, until all the tests pass or you run out of time (or find some insurmountable problem, in which case create an "issue" to explain what the problem is).

ASE, LLC strongly recommends that you use pair programming for developing and executing your test

cases. Note you can start on testing before completing your demo for Team Assignment 5.

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 change, but if your new/improved functionality was not previously presented, make sure to include it now.

pages 3 to N: Write a header "New/Improved Functionality" at the top of page 3. Then transcribe any modified or new CRC cards, Class diagrams and Sequence diagrams from JIRA. If none, then state "No changes" and nothing else. Do not omit this section.

pages N+1 to M: Make sure to start on a new page. Write a header "Test Classes" at the top of page N+1. Then on the remaining pages, transcribe all your test case issues from JIRA. Make sure to include your explanations of any failing tests (if you just ran out of time, say so).

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

In addition, deposit your entire code base, including any test cases, configuration files, scripts, etc. in STASH. You should do this before submitting to courseworks.

How do we know when we're done? When you run out of time. You will almost certainly run out of time before you have fixed all the failing test cases - assuming you have a reasonable set of tests. If all your tests pass, there is probably something you have forgotten to test.