

Black-box Unit Testing (and Demo): Team Assignment 4

✓ Published

 [Edit](#)

This is a team assignment.

First update your class diagrams to match what you have actually implemented to date; these might be significantly different from the class diagrams you submitted previously. (You do not need to include constructors, finalizers, getters, setters or classes that consist entirely of CRUD methods.)

Second develop a set of black-box unit test cases for your system, covering all the classes shown in these class diagrams. In particular, describe in prose the set of equivalence partitions and boundary conditions that need to be covered for every (non-CRUD) operation in every class, with sample test inputs and expected outputs.

Third implement your equivalence partitions and boundary conditions as executable testing code. Make sure to include appropriate setup/teardown and mocks/stubs as needed, grouping test cases according to dependencies (if any). You do not need to submit any prose for this part. You do not need to include system-level test cases and gray/white-box test cases now, those will be included in an assignment for the second iteration.

Fourth setup your testing code to be automatically invoked using a testing tool suitable for your programming language and development framework, and check that you can indeed run your entire test suite this way (in some cases this might be divided into two test suites using two different testing tools, for frontend vs. backend.) Tell us which tool(s) you used and any challenges that arose.

Fifth run the test suite, fix as many bugs as you can, rinse repeat, before the demo. Tag the repository to clarify which revisions of which files actually contributed to the demo. You do not need to tell us about the bugs you found and fixed, or found and didn't fix ... yet.

Sixth tell us which pair was responsible for what.

Schedule a demo with your team mentor before the deadline for this assignment. Plan for 20-30 minutes. The demo itself only needs to be about 5-10 minutes, but also allow time for your mentor to try to break your system (your mentor gets to drive). Your entire team should attend if possible. (The CVN team should arrange online screen sharing to present their demo.)

If your demo has not yet occurred before the deadline for this assignment, submit the assignment on time anyway and include a note at the top saying you have not yet shown your demo to your team mentor. Then submit again afterwards, removing that note and highlighting any other new revisions.

For the last part of the written assignment, describe what you were able to demo and record any problems that arose, any recommendations from your mentor, etc.

Each team should submit a single file, i.e., one member should submit. The name of your file should include your team name, and the contents of the file should also include your team name. The executable testing code, testing scripts, testing tool reports, sample data sets, etc. should all be stored in your git repository, not in the file submitted on canvas. Include instructions for access as part of the file submitted on canvas.

Points 10

Submitting a file upload

File Types doc, docx, pdf, txt, xls, and xlsx

Due	For	Available from	Until
Nov 9, 2017	Everyone	Aug 15, 2017 at 12am	Jan 31, 2018 at 11:59pm

+ [Rubric](#)