Miki Sugimoto
Donovan Roseau
Mac Knight

Software Engineering Team Term Project:
Deliverable #2

For this week's deliverable, the team dove into the task of testing code written by another developer (Martus). The lack of documentation on the methods was a setback to producing our first 5 test cases, but we managed to find some simple methods that should be the easiest to assess for our first experience with the process.

Requirements

Although the mission of the project is stated on the website, we have been unable to find any concrete requirements or expectations as to how the app should perform, or what features, etc., the developers are hoping to eventually add. This is not critical to the testing we will be performing, but it also leaves us without many clues as to how the different features are supposed to function.

Test Cases

Test Name: testOnOptionsItemsSelected
Method: onOptionsItemsSelected
Input: Menu Item Object
Output: boolean

Test Name: testOnCreateOptionsMenu
Method: onCreateOptionsMenu
Input: Menu Object
Output: boolean

Test Name: testGetSettings
Method: getSettings
Input: None
Output: mProgress Dialog Handler Object

Test Name: testGetBackButton

Method: getBackButton
Input: None
Output: Image backButton

Test Name: testGetApplicationName
Method: getApplicationName
Input: None
Output: String

Testing Schedule

9/27 - Create and specify five of the eventual twenty five test cases that we will eventually test.
10/18 - Rework test plan if necessary and create a framework to implement test plan.
11/10 - Framework should be completed and twenty-five cases should be ready for testing.
11/22 - Use five faults into code that will cause tests to fail.

Constraints

The team has already encountered several restraints and expects several more before the testing is complete. The primary concern is that the source code has little to no documentation as to what the methods are supposed to do, and several take in objects which are equally undocumented. This has presented a challenge in trying to figure out how to create tests for these methods. Another challenge has presented itself in that we are unable to run one test individually and examine the output, therefore adding another layer of confusion in creating our own tests for the methods.

The software also runs on very outdated on versions of all of the build tools and environments, even to the point where it will likely not function correctly on a current Android device. This is creating a very slow and buggy process, especially when it is required that the software interact with any new versions of the software, which means we are unable to test the product on an actual phone, which would be a helpful resource.

System Tests

Full system test are almost impossible at the moment, as the server connection with the emulator is not working. As mentioned previously, the app is also not allowing access to the site on a real phone, so it is unknown to us at this time if it even works in the “real” environment.