

Testing
SOS MarineVP
12/02/19

System and Unit tests. Include a document under GIT called Testing. Include in this document a list of the tests you have performed for each unit and for the entire system. For acceptance test, you will choose a number of tests from the system test to demonstrate the functionality of the system. Automated build and unit tests. FOR EXTRA CREDIT, your project can be automatically built (using Jenkins/Hudson, and unit tests automatically run. It is possible to run the product executable that is created by this automated build process). EXTRA CREDIT for automatic building and testing.

- Unit tests (25 points): Include a file/directory named 'Testing' in your Git Repository. There should be details (can be in a separate file in the directory) provided by each team member about the module and the functional testing they have done. Each team member picks a module or module and lists the equivalence classes and the test cases selected to cover all equivalence classes.

We performed manual testing throughout the duration of our project to verify the various inputs that our program had to take in (text description, photo, and long/lat coordinates) were being sent to FireBase and retrieved from FireBase correctly.

For the acceptance test, we manually filled in the various fields listed above by selecting a location on a map, verifying the coordinates we selected on the map were being sent to the database on FireStore and accurate. We did the same process for the text description and photo upload.

We also tested by logging into a Google account on the app itself and viewing previous stranding reports. All reports that were submitted to FireBase were then retrieved and displayed on the app. We confirmed that the reports contained accurate information and displayed in the order that they were submitted.

We did not write any official dynamic unit tests and instead relied on static methods of reviewing with our team members.