Testing Report

Mobile Platform Development

Word Count: 885

Daryl McAllister

S1222204

## Testing Intro

This report outlines the decisions undertaken on how to test the Earthquake application as well as the reasons for the decisions. The testing is made up of formal white box testing as well as equally important informal testing that occurred throughout implementation. These two forms of tested are outlined as they both occurred to ensure that application has minimal bugs.

## Formal Testing Methods

### White Box Testing

The main method of the formal testing for the application comes in the form of white box testing which is where the tester has knowledge of the implementation of the program. This method was done due to the limited resources during development and testing that made it not possible to have independent users test the application. This does not limit or bias the test however as the program was tested throughout development in white box method so that correct bugs could be ascertained and the formal white box testing at the end was primarily as a find all to find bugs that where missed in implementation so they could be fixed immediately. The white box testing was done by primarily testing features and requirements.

### System and Feature Testing

The formal testing that took place focused on both system testing where the project is tested as a whole in different environments ( in this case mobile phones ) as well as feature testing where individual features where tested to ensure they met the requirements as laid out in the requirements specification. Both the system and feature testing was undertaken to ensure full robustness of the system and to be exhaustive in the testing process as security is a driving development trend in both android applications but also all software.

### Testing Areas

The initial tests done where basic usability testing was done where the navigation and online/offline handling as these where essential to the applications use. The main requirements where then tested such as the searching features (largest magnitude, smallest magnitude etc.), date refinement and the ability to switch map and list views. The last portion of tests where to check that correct earthquake specifics where shown when chosen by the users.

These areas where tested to ensure the correct refinements in the data that they user specified where both functional and correct. In terms of the largest magnitude, it was tested to ensure that the earthquake shown was the earthquake with the largest magnitude. The date refinement was tested to ensure that non-relevant dates did not return a blank view and made users aware that no earthquakes existed on this day. Aspects that where more difficult to test such as the map where just as important as bugs may have gotten through that could not be spotted and the easiest way to test was to use multiple devices to ensure the output was consistent across each device.

### Testing Method

The testing for the specific areas was done by determining what the expected output for the test and comparing it against the actual output which would then show whether the test passed or failed. This was done for each of the test cases outlined before the main testing begain.

## Informal Testing

Testing occurred throughout development and out with formal testing in many forms such as unit tests or integration tests. These where important as they ensured that the program worked before continuing development and are considered an important testing phase.

### Unit Test

Unit tests where used in the application in order to test the model class “Earthquake” that is used to store a sterilized earthquake. This was utilized for the Earthquake class as it is an important class for the manipulation of earthquakes so the use of unit testing to test methods as they are added was beneficial to quickly integrating them into the project without having to worry about extensive testing on their addition. The methods that make up the Earthquake class are used to parse data from the large data strings and so they have to do so accurately which was an important part being tested. The unit test class for Earthquake did so by having a dummy earthquake set up that would be tested with the known outputs already known and tested against.

This proved effective and allowed for controlled unit testing to be accomplished and could be constantly re-ran if any changes were made. The largest downside to the unit testing is that the project was severely limited with what unit tests could be called and the Earthquake class being the only class that could be independently tested as other aspects such as the activities could not be effectively done so.

### Integration testing

The application was developed in segments with different features being developed one by one before then being pieced together. An important aspect for the application was to ensure these features worked together when integrated before moving on so that bugs did not pile up and this is why informal integration testing occurred throughout development. This testing was done by completing a development area and then running the application while testing that it was easily transitional to the features and they the program still worked overall. Often at times this helped find bugs during implementation that where then rectified before the project’s completion.