Mobile Platform Development Testing Report.

Craig McLaren

S1437087

Github Link: <https://github.com/cmclar/MPDAssessment> (video and reports are on the top level)

The application was developed using the agile scrum software development process wherein each aspect of the application was developed and unit tested separately to ensure each aspect worked as accurately and efficiently as possible.

Following on from the initial starter code given the first objective was to test what was given to ensure its functionality and reliability. Firstly the XML DOM was tested on its own to ensure the UI elements were loading properly. Next the onCreate override method was unit tested using the android debugging tool to ensure no missed or null references were found.

As each feature was implemented it was tested for every eventuality ensuring the application was as bug-free as possible while ensuring the required functionality was met. The class task was one of the first things to be developed further and tested. This was unit tested by passing in url strings and ensuring connections could be established. Next once it was verified that connections were established correctly the stream reader functionality was tested using various XML DOMs to ensure reliability with an output string of the class.

With the output of the Task class passing each test the linked list parseData function was created and using a custom incident java class containing four strings with getters and setters.

This function was once again unit tested passing in various XML DOM strings to ensure accurate results. It is through this testing that it was found that a boolean flag needed to be created to ensure only the items were being parsed in the DOM. This issue arose due to there being identical start tags of title and description for the main channel and each item subclass. To this end a boolean operator was created to help the parser distinguish between the channel title and description from each items title and description. It also became apparent through testing that the advice given in the starter code to throw away the first two lines of the parsed XML DOM was wrong and these lines were required to correctly parse the georss points.

Once the functions to fully parse the XML DOM and separate each into separate lists of incidents were created and tested adequately the next aspect of the application was to display this information in a meaningful way. Various approaches were initially taken for this such as displaying each incident in a list scrolling down the page though through user testing this was deemed to clutter the page with unwanted information and so the final iteration was settled upon showing one incident/roadwork alert at a time and cycling through the list using navigational buttons. Once developed this was tested for each possibility of onClick events. Through this testing several null operator checks were put in place to ensure each time a button was pressed there would be no errors received by the user for lack of data available. This testing was done once again using the android debugging tool using error log outputs and breakpoints to ensure correctly matching object references.

The ability to provide more information was moved to a popup toast window instead of having it displayed within the main application window due to user feedback of a cluttered interface. The toast popup was unit tested for reliability passing in various strings to ensure adequate functionality and no errors were present.

Following on from this the specification required a search function to filter a list of incidents or roadworks on a specific road. For this it was decided that the search function would be on another screen to prevent cluttering in the main screen of the application. In order to achieve this a viewflipper element was introduced to the XML DOM to be able to switch seamlessly through the different screens within the application. This function was unit tested by continually cycling through the different screens and choosing screens at random to ensure reliability.

Once the viewflipper element was integrated and found to be working correctly a search function was created to search through each list of incidents looking for a substring in each incidents description string variable. This was unit tested by passing in various substrings with and populating the incident lists with various amounts of incidents including empty lists to ensure correct functionality.

After all these aspects of the application were developed and tested individually the full application was tested from end to end to ensure each aspect worked together in the appropriate way. This was mostly done with user testing though some basic timed automated testing was done also. The user testing was mostly focused around the reliability and performance of the application with feedback given that it was accurate and reliable with no clear errors. From the feedback the loading time at the start of the application was a little slower than what would be desired and there could be scope for further features in the future such as map integration.