Mobile Platform Development Testing Report With Test Results

Cameron Crawford

CCRAWF203@caledonian.ac.uk

S1628376

Testing Report

In this report, there will be an insight into how the Traffic Scotland application was tested. Applications must be thoroughly tested, this is done to ensure that all of the functions that are contained within the application works as they should and the behaviours which they are expected to have when executing a specific function. Applications are also tested to ensure that it meets the requirements that it has been set at the start of the project.

# Methods Used

When testing the application, two main testing methods were used, the first was Black Box Testing and the other was Usability Testing.

## Blackbox Testing

The first testing method is black box testing, this type of testing is done on the running of the application, the tester has no knowledge of the code behind or what interactions should happen. The tests are carried out and are compared to the requirements which has been set at the start of the project. In terms of the development of this application, the testing was carried out with the knowledge of the initial document set out at the start of the project, this gave an understanding of what functions are required and if the functions implemented were doing what the program was indented to do.

When the testing began, there was a set of test cases which are predefined, when these are being carried out, documentation is produced, showing the results of each individual test, indicating whether they have passed or failed the test that was carried out. The testing is a thorough run through of the application, the tester tests the application using normal, extreme and exceptional data, to ensure that the application has sufficient try and catches and also has appropriate toast messages to inform the user when an error has been produced, as opposed to the application crashing while doing a process.

## Usability Testing

Usability testing is where the application is presented to potential users. Users have been assigned tasks to complete when they are on the system, they have no knowledge of the system at all and they have to get a hang of the application on their own and it is part of the test to get an insight into how easy the application is to understand and how well different users can interact with the functions of the application.

As strange as it sounds, the aim of usability testing is to highlight as many errors with the application as possible, this will be done prior to launch and if the developers can encounter the problem through users who they trust to test the application then instead of it going public and the application can end up getting hacked or could have an error which crashes the system or returns the wrong outcome from a function. These are all found through usability testing.

In terms of this application, numerous potential users were given the opportunity to look through the application, try and understand how the application worked and if there were any unexpected errors with the application that could be flagged up to be fixed before the submission. The testing returned no errors, the application reacted as it should to all of the tests.

## What was tested?

In order to make sure a successful testing process was carried out on the application, there was a clear process of what needed to be tested but also what each test was looking to do.

On the home page, each button had to be checked to ensure that when they are initiated then they link to the correct page.

On the Planned Roadworks page, a few tests had to be carried out, the first test was to ensure that the planned roadworks actually loaded from the rss feed and had been correctly parsed into application and that the adapter had then passed the information to the user interface to be able to display all of the information that is required on the application’s List View. The next was testing the search function, the user would be able to enter in a search word and once the user pressed search, there had to be a test that the correct query had been made and that the results weren’t the results already present. A test was carried out on the details page of the application, one before it was searched and then one on the results of a search, the user would select a roadwork and then the details of that roadwork was loaded, displaying its details. The back-navigation functionality was also checked to ensure it went back to the page before.

The incidents page was loaded and the test on whether the incidents were displayed was carried out. The incidents were displayed on the page in a list view and they were made selectable, so a test was carried out when the user had selected one of the incidents and then the details were displayed. This was done to ensure that the correct data was being pulled from the feed and displayed on the application.

A test was also done on the orientation management of the system. This was done to ensure that when a user twists their screen then the application will manage the orientation and will adjust the screen to the appropriate layout design for the landscape mode. Most applications, if they cannot cope with this, are marked down on their quality, so it was essential to make sure that the application could cope with the change.

Finally, a test was also carried out on whether the application can display the appropriate message when the device loses its access to the internet. The purpose of this is to ensure that if the device loses connection then there is no access to the rss feed but at the same time the application does not crash. Also, the test was used to ensure that an appropriate error message was displayed when the device did lose connection, informing the user of the problem and that it is not a system error.

Testing

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test Number | Test Case | Input | Expected Output | Actual Output (Y/N) |
| 1 | Test that the application loads, displaying the home page | Load Application | Application Loads | Y |
| 2 | Test that the roadworks button can be selected, and the roadworks page is loaded | Click Roadworks button | Roadworks Page loads displaying the current roadworks | Y |
| 3 | Test that the roadworks has been displayed showing a colour coordinated list | N/A | Roadworks are colour coordinated | Y |
| 4 | Test that a user can select one of the roadworks on the list view, which displays the details page for that roadwork | Select a roadwork | Roadwork details page is displayed | Y |
| 5 | Test that the back button takes you back a page | Click back button | Page goes to the previous page | Y |
| 6 | Test that when a user enters an incorrect value to the search field then an error message is displayed | “#” | Error Toast is displayed | Y |
| 7 | Test that when a user enters a search, the appropriate details are produced | “M74” | All roadworks for M74 is displayed | Y |
| 8 | Test that when the user selects on one of the results from the search, that it loads the appropriate details page | Select Roadwork | Appropriate roadwork details page is displayed | Y |
| 9 | Test that the current incidents page can be displayed | Select current incidents page | Current Incidents page loads, displaying all incidents | Y |
| 10 | Test that when the user selects one of the incidents then the details of that incident is displayed | Select an incident | Appropriate incident details is displayed | Y |
| 11 | Test that the user interface displays the appropriate view when you switch the application | Change Orientation to Landscape | Screen adjusts to the size | Y |
| 12 | Test that when the virtual device has no internet access, the user is informed they cannot connect to the internet | Turn off internet and load roadworks | An Error message is produced informing the user they do not have access to the internet |  |