

## Mobile Platform Development

Lewis Duncan

S1630772

BSc (Hons) Computing

## Table of Contents

[Mobile Platform Development 1](#_Toc509942066)

[Table of Contents 2](#_Toc509942067)

[Project Design Introduction 4](#_Toc509942068)

[Design Report Overview 4](#_Toc509942069)

[Custom App Icon 5](#_Toc509942070)

[Splash Screen Activity 6](#_Toc509942071)

[Home Screen Activity 7](#_Toc509942072)

[Asynchronous Pre-Execute Dialog 8](#_Toc509942073)

[Item List Activity 9](#_Toc509942074)

[Item Activity 10](#_Toc509942075)

[Testing Report Introduction 11](#_Toc509942076)

[Testing Strategy 11](#_Toc509942077)

[Test Cases 12](#_Toc509942078)

[Conclusion 14](#_Toc509942079)

## Project Assignment Information

You can find the whole project folder, recording folder of the video file for screencast-o-matic and the word document at the following link: <https://github.com/lewisduncan93/Mobile-Platform-Development>

## Project Design Introduction

Within the development of this project, one of the requirements is for the app to be compatible with Android Marshmallow (6.0) and to use the Traffic Scotland XML RSS feed API for Current Incidents and Planned Roadworks.

Within this document details many different elements of the project such as:

* **Design Report Overview** – Details the summary of what is included within this project design and implementation and the reasons behind them.
* **App Screens in the Emulator** – Images of each screen within the emulator.
* **Activity Mock Designs** – Mock designs of each activity.
* **Testing Report Introduction** – Introduction on the testing report and what it covers.
* **Testing Strategies** – Details of different testing strategies used within the project.
* **Conclusion** – Overall conclusion of the whole project and any details of changes or deviations from plan originally intended.

## Design Report Overview

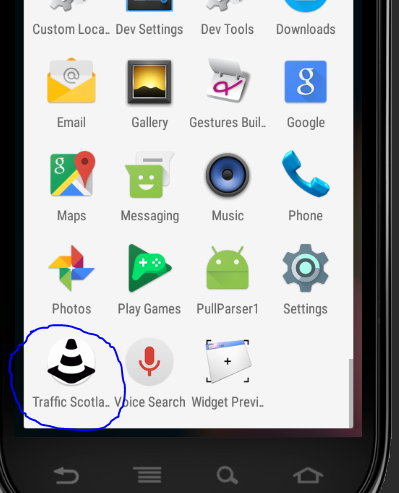
The main objective of this project was to develop an android app which allows users to browse through and search for any planned roadworks and current incidents from the Traffic Scotland RSS feeds. Other features were integrated such as allowing the user to click on a specific search result item that was displayed to see further information regarding that result such as a link to the Traffic Scotland website, a geo point which is shown in Google maps once clicked and other relevant information. A lot of thought for the design was taken into consideration due to the scale and size of the smart phone screens which are very limited and restrictive. As a result of this, each of the the components had to be placed in an efficient but effective manner.

Human Computer Interactive design principles such as Shneiderman’s Golden Rules influenced a lot of the decision making in this project, as the application contains shortcuts such as a back button on the toolbar for easy transitions to the previous page. The components within the app were designed primarily to be consistent and align with each other so that the margins from the sides were equally the same width. The app is responsive to portrait and horizontal screen views so that the user can tilt the screen in any direction. Asynchronous method tasks were implemented to allow for tasks to run in the background smoothly without overflowing the processing capacity and to ultimately reduce short term memory load. Top down interaction was the preliminary concept for this app which enables users to take control more easily. The font family used for the text throughout the app was monospace which provided a very unique style to stand out aesthetically.

The following pages outline each screen or activity within the app and the reasons for them.

## Custom App Icon

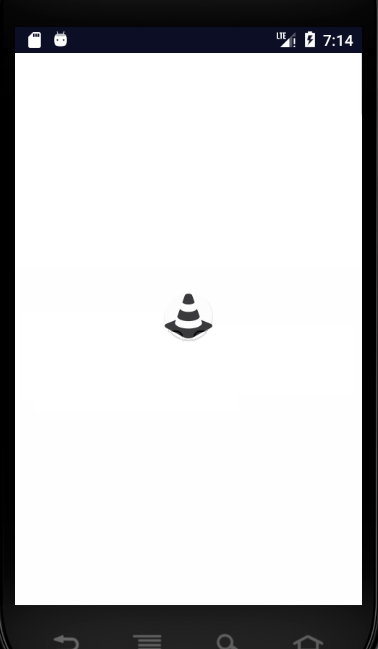
A custom icon was made for the app which is shown below in figure 1. This was produced to achieve an aesthetic effect which displays a traffic cone as the logo or icon to identify it is an app primarily associated with traffic. The default icon images were replaced in the ic\_launcher folder within the mipmap folder.



*(Figure 1: Custom app icon circled in blue)*

## Splash Screen Activity

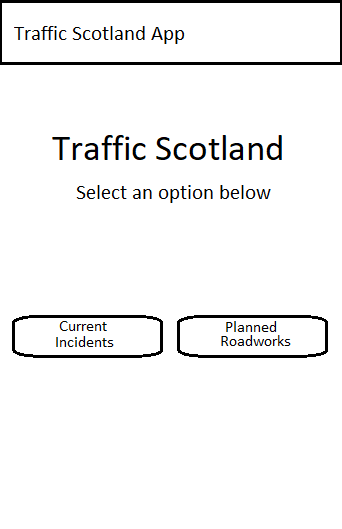
Once the user clicks on the icon a splash screen is displayed before loading the home screen as shown below in Figure 2. The splash screen uses the image as the custom icon mainly for consistency purposes. The gravity of the image was centred within the splash screen XML file.



*(Figure 2: Custom splash screen)*

## Home Screen Activity

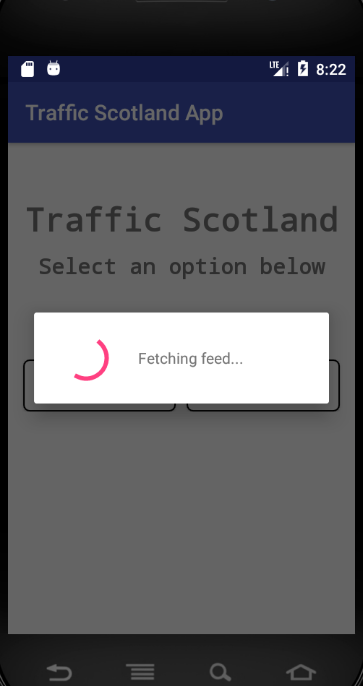
Within the home screen there is two buttons to select from which show you either the current incidents or planned roadworks feed. The theme was designed in a way that it’s simple contrasting shades or colours which can be easily identifiable for anyone who might be colour blind or have problems identifying different colours or shades that are too close to tell from. The buttons are both equal in size and both leave the same space between them for consistency. Each button is aligned by a linear layout and has a border defined within the XML file. The two textviews above the buttons are large enough which acknowledges that this is the home screen. These textviews are both encapsulated within separate linear layouts. The home screen mock design is shown below in figure 3 and what it looks like in the emulator in figure 4.



*(Figure 3: Home screen mock design) (Figure 4: Home screen in emulator)*

## Asynchronous Pre-Execute Dialog

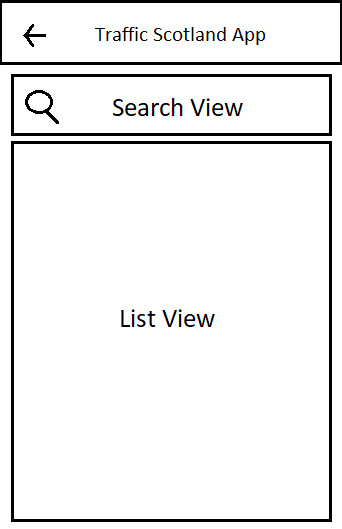
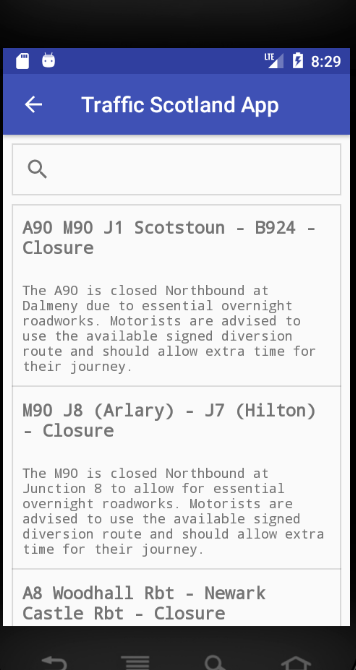
Once the user clicks on either button from the home screen, the task will then perform depending on which button clicked. Asynchronous tasks were implemented as mentioned earlier. A pre-execute dialog was designed to display a message that the data is being fetched from the feed, shown below in figure 5. This informs the user of the current situation.



*(Figure 5: Async Pre-Execute Loading Dialog in emulator)*

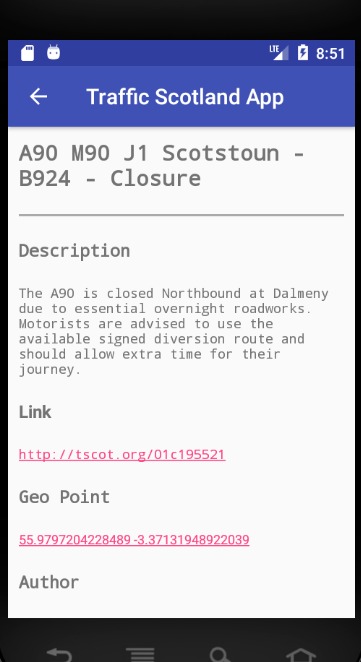
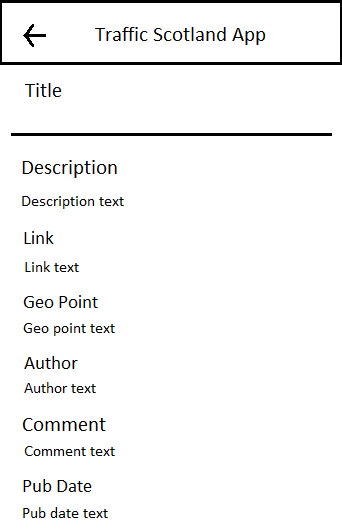
## Item List Activity

The results from the RSS feed –depending on the button the user has selected- will display in a well presentable form, each item is separated and contained within a listview which can be scrolled. A searchview is above the listview which can be used to filter through and search for any specific results by clicking on or near the magnifying glass icon which will activate a keyboard on the phone to allow the user to enter text. The searchview and listview components are aligned by a constraint layout which provides spaces between them instead of them being crammed together which may result in any confusion. For each of the components a border was defined within the XML file of the item list activity. As mentioned before, a back button was implemented on the toolbar to allow the user to go to the previous page. The back button is primarily a shortcut which is ideal ease of access for helping out users get back to the last page. Each item within the listview can be clicked on which displays further details regarding that item. All of this discussed is shown below in figures 6 and 7.



*(Figure 6: Item List Activity mock design) (Figure 7: Item List Activity in emulator)*

## Item Activity

The item activity, as shown in figures 8 and 9, displays further detailed information from the item such as description, link, geo point, author, comments and published date. A separator line is used below the title of the item to differentiate between the title and the detailed information. The text is all encapsulated within a scrollview which can be used to scroll through the item. The item activity allows for the user to click on the link which will take them to that item result within the Traffic Scotland website, this was done within the XML file by assigning the autolink as a ‘web’. The geo point is also clickable which shows the specific location on Google maps. The red text was set within the Java code of the activity and the link was set within the XML file by assigning the autolink as a ‘map’. Like the item list activity, a back button was also added on the toolbar within the item activity to allow the user to go back to the item list activity.

*(Figure 8: Item Activity mock design) (Figure 9: Item Activity in emulator)*

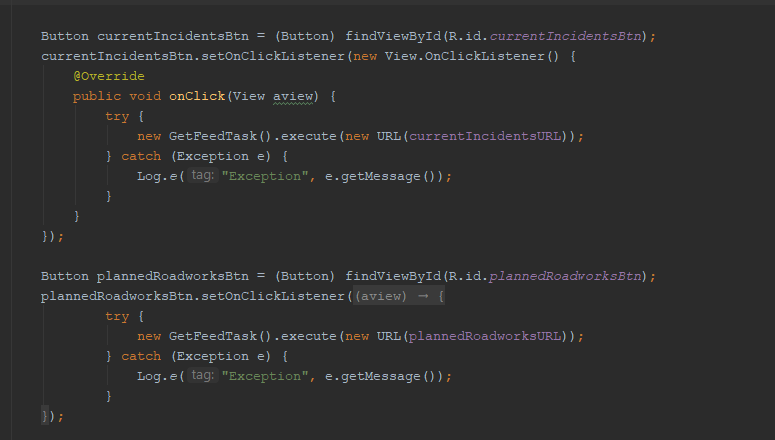
## Testing Report Introduction

Testing is crucial in order to determine if there are any flaws or problems within the program such as programmer mistakes or ineffective programming. Problems can arise within apps and the user may experience unexpected or unwanted results therefore it is essential for error handling to be implemented to determine the root cause. This will be discussed later on in the report.

This testing report will contain two sections. The first section is the testing strategy to develop the project and the second section is the documented testing results.

## Testing Strategy

The testing strategy when developing this project involved black box and white box testing. Any other testing strategies such as stress testing user inputs are not necessary in this project due to the fact that there is only one search bar which is resistant to flaws and errors that are primarily beyond the scope of this project.

Try-Catch statements were implemented to catch any problems as shown below in figure 10. These statements are considered to be the most popular choice for handling errors, as they are very effective against application crashes and can enable unexpected errors to be handled and displayed for programmers to understand the error more clearly.

*(Figure 10: Try-Catch statements implemented into setOnClickListenrer methods)*

## Test Cases

|  |  |  |  |
| --- | --- | --- | --- |
| **Main Activity** | | | |
| **Test Case** | **Expected Result** | **Actual Result** | **Comment** |
| Invoke Current Incidents Button | Open Item List Activity and display Current Incidents results | Opens Item List Activity with Current Incidents results displayed | Operating as expected |
| Invoke Planned Roadworks Button | Open Item List Activity and display Planned Roadworks results | Opens Item List Activity with Planned Roadworks results displayed | Operating as expected |
| Exit the application | Close the app | App closes | Operating as expected |

|  |  |  |  |
| --- | --- | --- | --- |
| **Item List Activity** | | | |
| **Test Case** | **Expected Result** | **Actual Result** | **Comment** |
| Invoke Back Button | Return to the Main Activity | Returns to the Main Activity | Operating as expected |
| Invoke Search Bar | Search Bar should allow user to input data and enter query | Search bar allows user to input data and query | Operating as expected |
| Invoke Item on Listview | Open Item Activity and display correct information | Opens Item Activity and displays correct information | Operating as expected |

|  |  |  |  |
| --- | --- | --- | --- |
| **Item Activity** | | | |
| **Test Case** | **Expected Result** | **Actual Result** | **Comment** |
| Invoke Back Button | Return to the Item List Activity | Returns to the Item List Activity | Operating as expected |
| Invoke Link | Link should direct user to the correct URL as displayed | Link directs user to the correct URL as displayed | Opens up Traffic Scotland with the correct URL externally outside the app |
| Invoke Geo Point | Open Google maps and display the location retrieved from the geo point | Opens Google maps and displays the location retrieved from the geo point | Opens up Google maps externally outside the app |

## Conclusion

In conclusion the development from this project has proven to be a huge challenge and a new learning experience. One of the biggest challenges was working with Android Studio which proved difficult due to the nature of not using it before.

Unfortunately, this project does not meet some of the additional features as intended due to the short time span as well as working alongside other project deadlines. This has made the overall project development very difficult. Initially the Google maps activity was to be implemented, however, due to very limited time this idea was replaced by using Google maps externally outside of the app instead of as an activity.