Mobile Platform Development Coursework

James Craig – S1428641

# Introduction

Throughout the course of this project I have created an interactive mobile application, running on the Android operating system, which displays information from Traffic Scotland about current road incidents as well as planned roadworks. This document will discuss the design choices for this application as well as the testing methodology used during the development process. It also contains any relevant links to resources found online pertaining to the project.

## Relevant Links

Main github project: <https://github.com/jcraig203/courseworkmpd>

Video: <https://github.com/jcraig203/courseworkmpd/blob/master/James%20Craig%20S1428641%20Screen%20Recording.mp4>

<http://i.pendual.eu/bsSFBAQC.mp4> (backup link)

Project APK:

https://github.com/jcraig203/courseworkmpd/blob/master/james%20coursework.apk

# Design

The goal of this project was to create a functional, user friendly application which displays information from Traffic Scotland’s RSS feed in an appropriate manner. To make sure the application meets this goal several design requirements must be taken into consideration.

## Overall Design Requirements

The application makes use of the default android UI design guidelines, ensuring that colours and styles are consistent across every view in the application. This is also known as the material design, and makes use of grid based layouts for menus and displaying information.

This includes a top toolbar which displays the page that the user is currently on to ensure the user knows where they are in the application. It also includes any navigation features, which in this instance is represented in the form of a back button, which uses the default android back button. It persists across every single view, always remaining at the top of the screen to allow the user to easily access any navigation features if they so desire. The bottom on screen/hardware buttons on the android device can also be used to perform navigation within the application.

All application backgrounds are white in line with material design guidelines, and all text is formatted appropriately using relevant styling i.e. Titles are larger than plain text and dates are displayed in an unobtrusive lighter colour.

The author’s matriculation number is also present on all views (as per design requirements) and can be seen in the bottom left corner of the screen.

## Home Screen

The home screen only contains two major elements in the form of buttons, which allow the user to either show current incidents or planned incidents. The buttons are placed appropriately in both horizontal and landscape views, allowing the user to easily press them to navigate them to the next page.

## Current Incidents

The current incidents page utilises a RecyclerView to display each of the parsed incidents in a logical, scrollable list.

Whilst loading, a popup message with a spinning circle is shown to indicate that the page is still processing and is dismissed upon load.

This list shows the user the title of the incident as well as the time, and can be pressed to navigate to a page showing more information about the incident.

The list renders correctly in both vertical and horizontal layouts, allowing for users to easily continue using the application in any orientation.

Each item in the list is divided so that the user can differentiate between incidents. Titles are enlarged (16px) text and dates have been coloured lightly in accordance with material design principles.

## Planned Roadworks

The planned roadworks page also makes use of a RecyclerView to display each of the parsed roadworks in an easy to navigate list. As with the current incidents page it shows the title and time, renders correctly in any orientation and divides items on the list, showing titles and dates with appropriate sizes and colours.

Whilst loading, a popup message with a spinning circle is shown to indicate that the page is still processing and is dismissed upon load.

Unique to the planned roadworks page, there is a floating button which appears in the bottom right hand corner. It maintains it position no matter where the user scrolls in the page, allowing for it to be pressed at any time. It is styled with the default android coloured button, which is a vibrant pink colour, which makes sure that it’s easily identifiable.

Another unique design feature on this page involves the date of each item being surrounded by a coloured box, representing the length of each planned roadwork. If the length is less than two days it renders in a green colour, if it is more than two days but less than five it will render in an amber colour and if it is longer than five days it will render in red. This allows a user to see at a glance roughly how long each roadwork will take without having to open up the more information page.

When this button is pressed, it opens a date picker which allows the user to select a date from a calendar. When a date is selected it will refresh the page, showing only results that occur on that specific date.

As with the current incidents page, each item on the list can be clicked to navigate to a page displaying more information about the selected planned roadwork, and can be navigated in either a landscape or horizontal orientation.

## More Information

The “More Info” page displays the title of the selected incident or roadwork, as well as its date and description. This allows the user to easily read more information about the event, as provided by Traffic Scotland.

This page, as with every other view in the application, has an appropriate vertical and horizontal view allowing for it to be accessed no matter the orientation of the device.

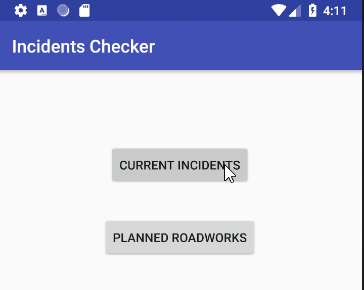
# Testing

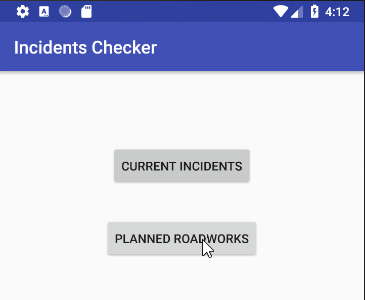
Throughout the development of this application, to ensure that requirements were met and that no errors were present, each iteration and implemented feature had to be tested. This section of the document will describe the testing processes used during each development phase of the project.

## MainActivity

This is the start page for the application developed in the project, and as such was the first thing to be implemented. It contains two buttons which will start the appropriate activity – either the IncidentsActivity or the PlannedActivity.

To check that this works, the app was loaded on to a device and each button was tested, making sure to check the console output to ensure that there are no issues.

C:\Users\James\AppData\Local\Microsoft\Windows\INetCache\Content.Word\studio64_2018-03-28_05-14-29.png



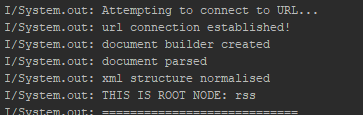
C:\Users\James\AppData\Local\Microsoft\Windows\INetCache\Content.Word\studio64_2018-03-28_05-11-17.png

## Current Incidents

In the current incidents Activity, there are a number of functions that had to be tested to ensure they worked correctly.

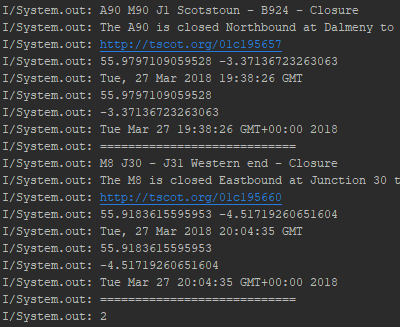
Firstly, when the view opens it must start a new task to connect to the Traffic Scotland feed, after which it will create a new document builder and parse it. The XML structure is normalised and the root node is identified, which allows for the items in the XML file to be identified and stored.

To make sure that each of these steps have been completed correctly, a confirmation message is posted in the console:



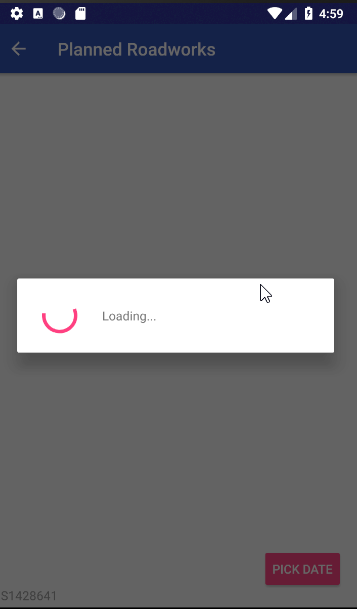
After the document has been loaded and parsed, each element must be stored in an Incidents object. A for loop is created and then each item from the feed is stored in an ArrayList of the Incident object type which will allow it to be displayed.

To ensure that each function correctly exectutes and the items in the feed are read, each item is broken down and output in the console, followed by the size of the ArrayList (to ensure that the correct number of items have been loaded into the list.

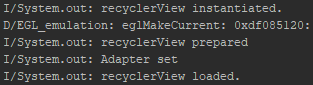


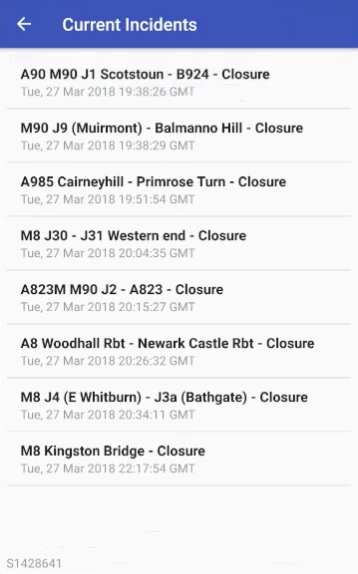
When the document has been parsed, the view is prepared and displayed. This involves instantiating the recyclerView and then making sure that it animates properly. The adapter for each incident in the list is defined (this controls how each item in the list displays).

Whilst the view is being prepared, a loading circle is displayed and is dismissed when the page has fully loaded.



The progress of each item can be seen in the console, which allowed for errors to be tracked if application crashes occurred and to ensure that each stage has worked correctly.



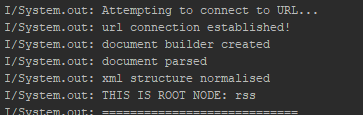


## Planned Roadworks

The planned roadworks activity shares a number of functions and features with the current incidents activity, as such a large amount of the testing performed is practically identical to the

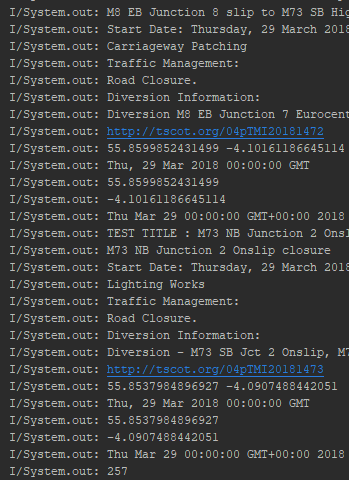
Firstly, when the view opens it must start a new task to connect to the Traffic Scotland feed, after which it will create a new document builder and parse it. The XML structure is normalised and the root node is identified, which allows for the items in the XML file to be identified and stored.

To make sure that each of these steps have been completed correctly, a confirmation message is posted in the console:



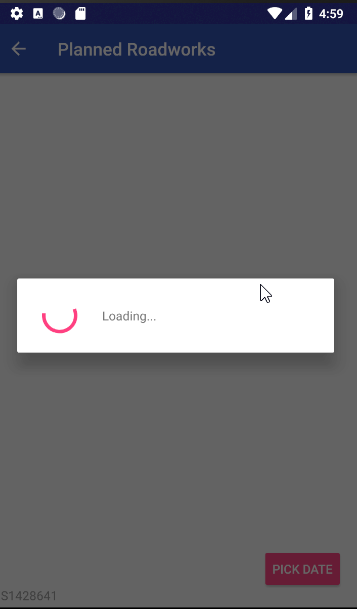
After the document has been loaded and parsed, each element must be stored in an Incidents object. A for loop is created and then each item from the feed is stored in an ArrayList of the Incident object type which will allow it to be displayed.

To ensure that each function correctly exectutes and the items in the feed are read, each item is broken down and output in the console, followed by the size of the ArrayList (to ensure that the correct number of items have been loaded into the list.

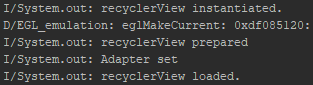


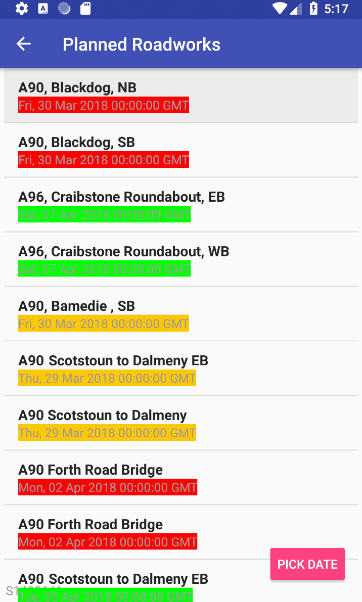
When the document has been parsed, the view is prepared and displayed. This involves instantiating the recyclerView and then making sure that it animates properly. The adapter for each incident in the list is defined (this controls how each item in the list displays).

Whilst the view is being prepared, a loading circle is displayed and is dismissed when the page has fully loaded.

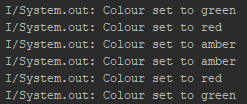


The progress of each item can be seen in the console, which allowed for errors to be tracked if application crashes occurred and to ensure that each stage has worked correctly.

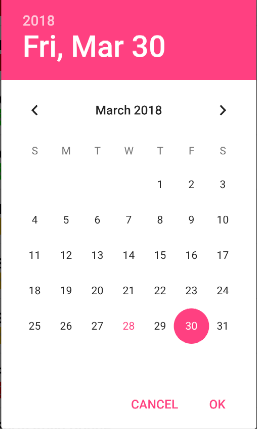
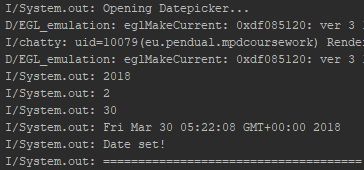




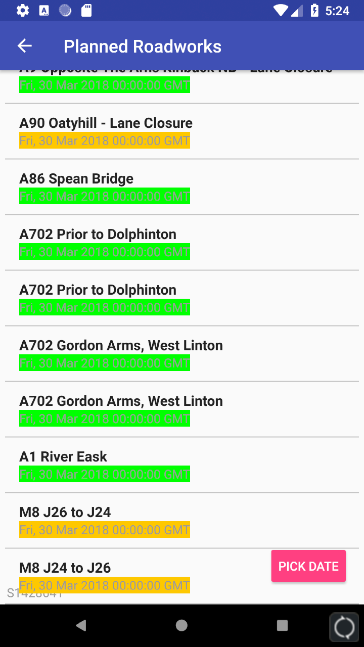
Since the application displays a different colour for each incident length, we must make sure that the colours are being properly set. Each time a colour is applied it is pasted in the console, indicating that it has been set correctly.



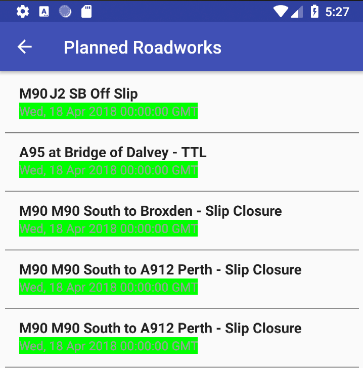
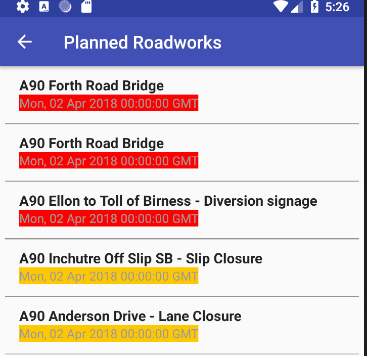
As we can also see, the floating button has also been rendered correctly. When pressed, it indicates as such in console and then opens up a datepicker. When the date has been picked it also indicates as such in the console.



Once a date has been picked, a similar method to the one that displays all of the events in the list is called, which checks if the date is the same as the one selected and then displays it if it is.



This was tested using a number of different dates to ensure that it was indeed functioning correctly.



## More Information

To ensure that information displays correctly for each event when clicked, a number of random events were picked and checked to ensure they displayed unique data.

