Mobile Platform Development

Design Report - Martin Kolev (S1435614)

# Resources

GitHub Repository: <https://github.com/Damnitsmarty/TrafficScotland>

Demo Video: <https://github.com/Damnitsmarty/TrafficScotland/blob/master/Recording%20%232.mp4>

# Design

Per the Coursework Specification, the design requirements of the application can be summed up in the following manner:

* The application should deliver to the user a report of the Current Incidents / Planned Roadworks in a way suitable for mobile browsing.
* The items in this report should contain meaningful information, delivered in an intuitive way.
* The application should differentiate between different device orientations and when needed, change its layout to better suit readability
* The application should provide a way to filter through the displayed information by Date or description of items needed (e.g. searching by road indicators – M000, A000 – or directions – Southbound, WestBound, etc.)

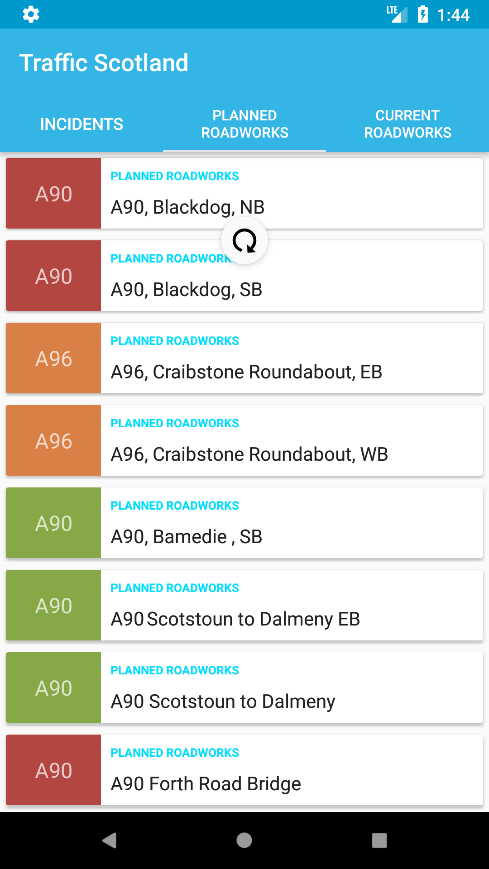
With these objectives in mind, after research conducted on different design paradigms and HCI principles, the conclusion reached was to use Google’s established Material Design patterns and guidelines to produce the application interface.

Figure MainActivity

Material Design is a design language developed by Google for use in mobile and web applications. It offers designers a meaningful way to display content for mobile through the use of surfaces, edges, materials that are “grounded in reality”. Minimal use of different, contrasting colours draw the user’s attention where it is needed. Shadows provide elements with a clear and distinguishable hierarchy. Material Design has been the primary design language used for Android applications, due to its roots in the Android platform. Because of this, the language was determined as the best possible choice and was followed strictly while developing the application.

## Colour Scheme

Colour schemes of applications using Material Design are composed of four colours – a Primary colour, and Accent colour and Light and Dark variations of the Primary colour. Initially, the colour scheme of the application was derived from the TrafficScotland website. However, the colours in this initial scheme were too dark for Material’s preference of “paper-style” pastel colours. Furthermore, no Accent colour could be derived from the website.

The next variation of the colour scheme used material recommendations for colours, derived from the material Colour Tool (available at: <https://material.io/color>). While the preference for blue as the primary colour was carried over from the previous scheme, the second version instead used a lighter, almost cyan hue of the colour as its Primary.

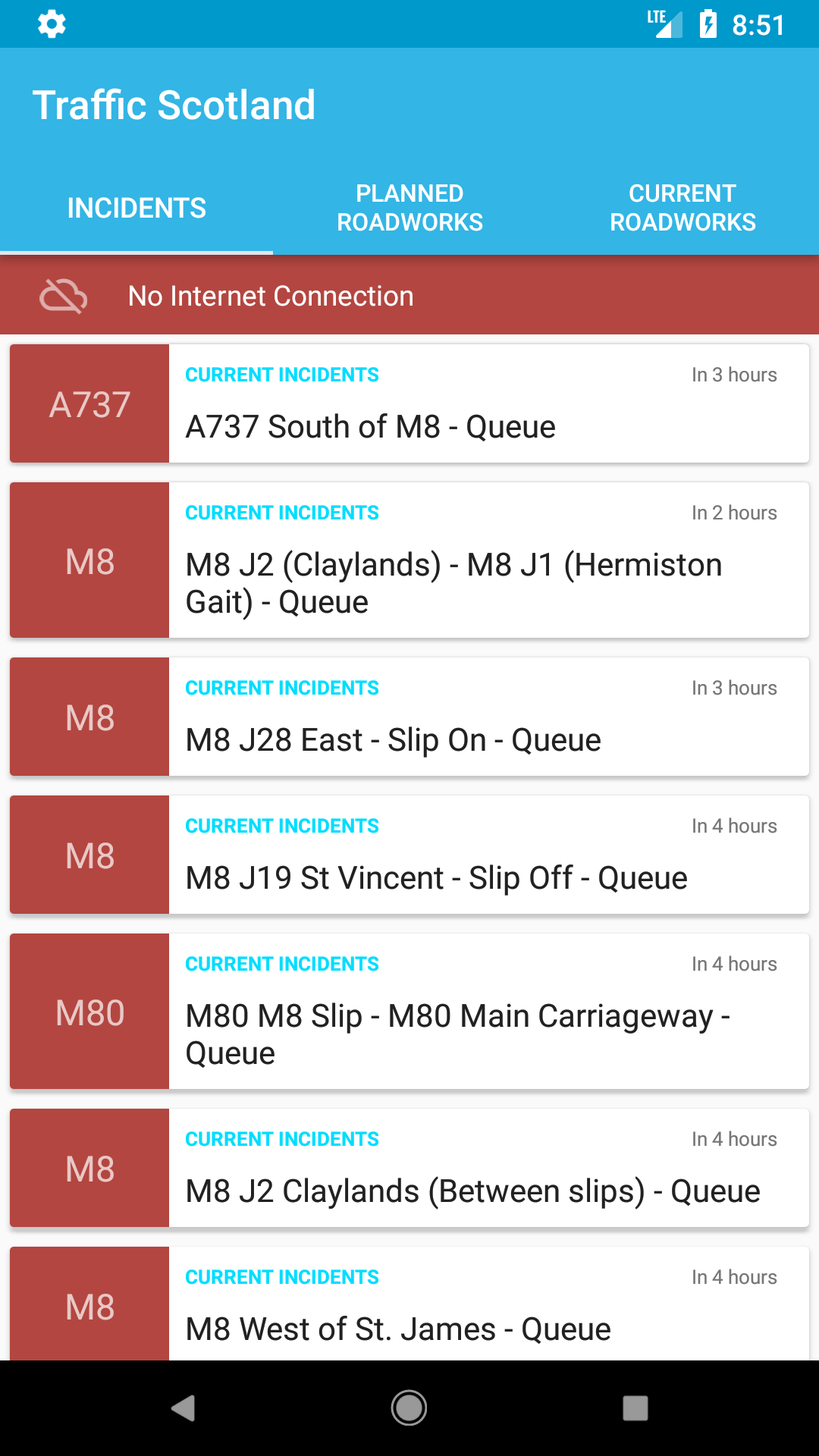
In addition to the four colours of the Material Design specification, three new colours were defined to be used for color-coding the displayed items. These colours were also derived from the Colour Tool.

## Layout

### Main Activity

The Main Activity acts as an information dashboard in the application. The user should be able to see a high-amount of low-intensity information. Naturally, to convey this information a list of data items was used.

Research suggests that the dominant mobile orientation is Portrait mode. Furthermore, because the nature of this application is displaying a list of information, which is more naturally conveyed in this mode, the application design and testing was focused primarily on Portrait, while offering support for Landscape.

When designing the layout of the main activity, the most important View – the List of data items – was given the most amount of space. Because of this, inclusion of Views other than the List view was kept to a minimum. The final version of the design consisted of a Toolbar displaying the name of the application, a TabLayout containing tabs for each available RSS feed and the List itself. Device memory constraints were also taken into account, by replacing the normal ListView with a RecyclerView instead. RecyclerViews operate on the same principle as ListViews with one key difference: items that are not displayed on the screen are not kept in the device memory and are reconstructed when entering the screen.

Error handling is also present in this activity in the form of a layout containing an optional error graphic and error message. This element is positioned below the TabLayout and displays an appropriate description and icon when encountering an error.

Figure Error message being shown

Search functionality was also implemented to allow the user to filter the Incidents / Roadworks through a specific query. The search option was designed as part of the Main Activity’s toolbar, as described in this article: <https://developer.android.com/training/search/setup.html>.

Figure 4 Search functionality in the app toolbar

### Main Activity – List Items

The RecyclerView object occupies about 85% of available screen space on average and displays 6-10 items depending on the device resolution. This abundance of data displayed on screen means that data items need to be easily scannable. Including every piece of information available will increase the short-term memory load of the user, which goes against one of the golden HCI rules defined by Ben Shneiderman. The solution to this problem is to establish a hierarchy and only display the most significant pieces of information on the Main Activity, while allowing the user to ‘expand’ any item and see the data piece in its entirety.

In order to help with establish the hierarchy, a study was performed in a small group of people. This study included showing all pieces of information offered by the RSS Feeds in a non-hierarchical fashion and asking each participant to rank them in order of importance. The study showed that the most valuable piece of information was the road number, followed by the Incident/Roadwork title (which offers further clarification on the location).

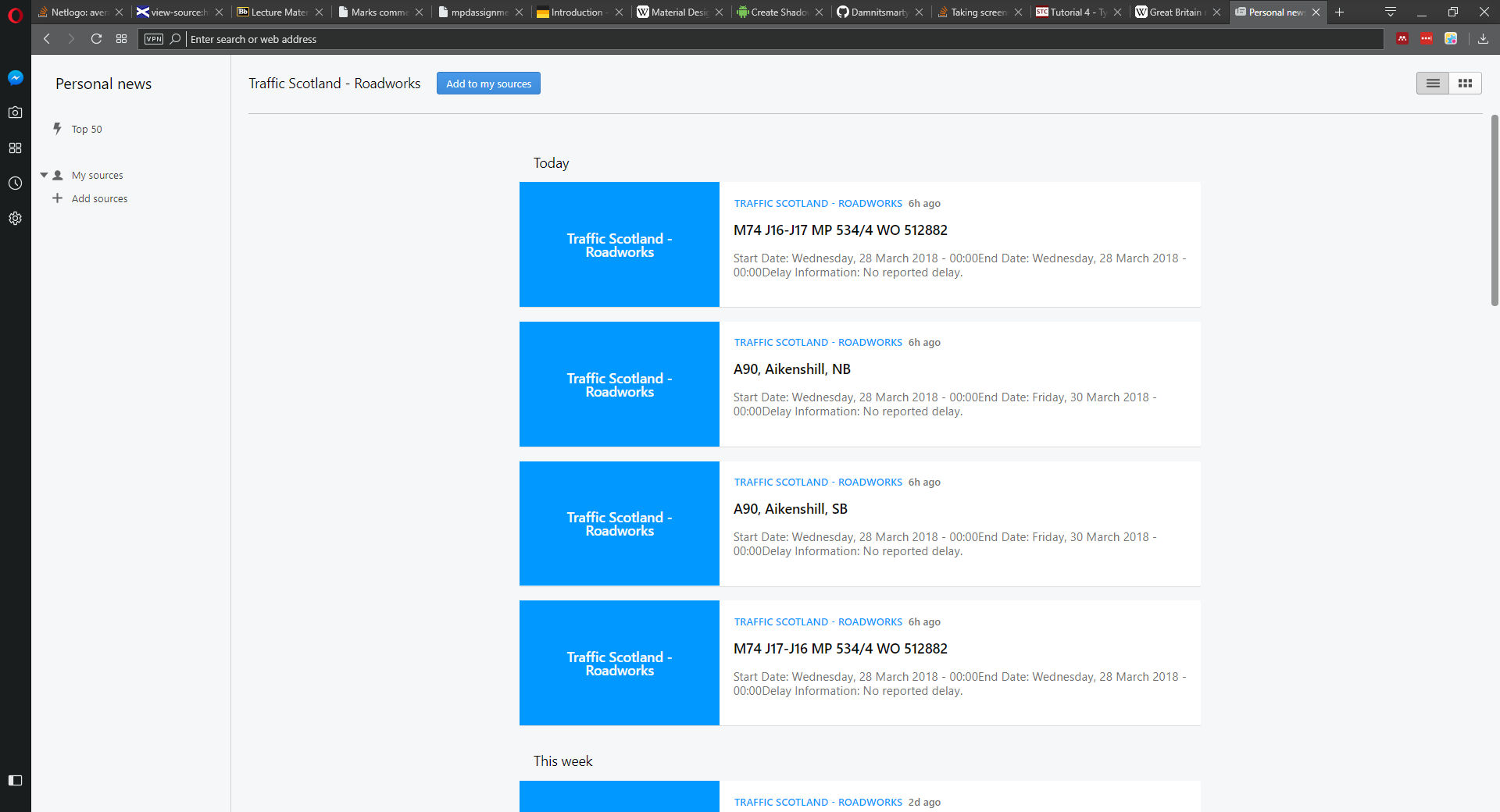


Figure Opera's RSS Feed Viewer

The layout of each item was largely inspired by the RSS Viewer of major web browser Opera. Using that layout as a base, the description of each item wsas removed, and the left “accent” slot was changed to include the Road number. Furthermore, the background colour of each item’s accent slot was colour-coded to reflect the remaining construction time – Green for less than a week, Orange for less than two weeks and Red for more than two weeks remaining. (Note that since the Incidents RSS Feed does not offer end times, every item from it has been coloured in red)

The design of the layout of list items strictly followed the appropriate Material Design Guidelines (available at: <https://material.io/guidelines/components/lists.html>) where applicable.

### Details Activity

When the user wishes to see the rest of the information for a specific item, he can tap on the item in order to launch the Details Activity.

The Details Activity contains several pieces of information that need to be conveyed properly.

The toolbar has several changes when compared to the one in Main Activity. Most notably, the item is located roughly at the vertical centre of the screen, below a MapFragment element. This change of position, while in contrast with Android design guidelines, is actually encouraged by Material Design for pages displaying detailed information, when there is a more important element that should be positioned on top. (for example, Google uses this technique to display Contact Profile images in their Contacts app) The toolbar also retains the colour-coding of the item from Main Activity using it as its background. Finally, the back-arrow icon is displayed on the left of the Title, leading the user back to Main Activity if pressed.

To display the geographical location of the Incident/Roadwork in a natural way, the application employs the use of the Google Maps API and MapFragment element. To display the description, a multiline TextView is used below the toolbar, which can be scrolled to reveal more content.

A different layout was included for Landscape orientation in order to better position the MapFragment. In this mode, the MapFragment instead fills the entire left half of the screen, while the details are displayed on the right. The toolbar is also positioned at the top of the screen.

## Further work

The main activity was also able to hide the toolbar and TabLayout when scrolling in order to maximize the number of displayed items in the list. This was done through the use of CoordinatorLayout and AppBarLayout. This functionality was later removed due to rendering errors causing the Fragments containing the RecyclerView to become unpopulated and unusable.

Further research is also needed to implement ‘Hero’ transitions between Main Activity and Details Activity. Hero transitions involve transitioning between two activities while retaining and transforming some key Views present in both activities.