**Mobile platform Development**

**Craig Higney**

**S1630775**

**https://github.com/chigney/mobile.git**

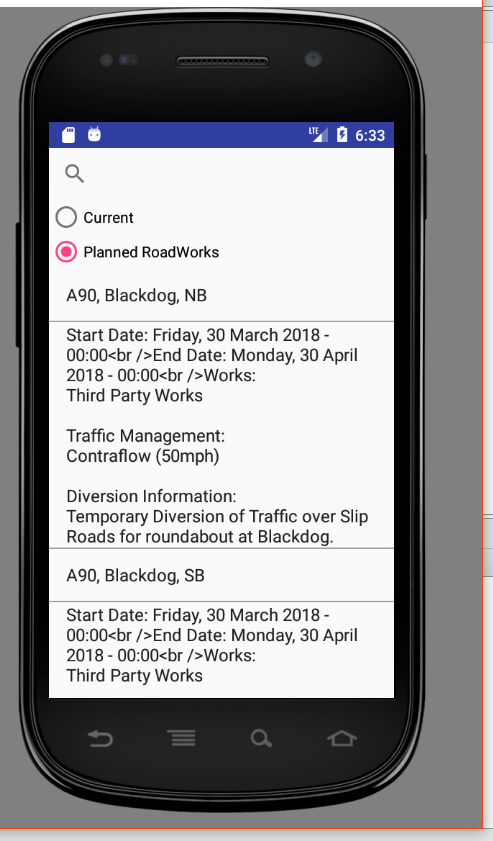
Project Design

Following the requirements specification, a mobile application was created using Android Studio version 2.3.3 designed to be compatible with the android emulation of a phone using Marshmallow release with API level 23 which aims at android 6.0 and makes use of google APIs. The aim of the application was to access the traffic Scotland website and to read the RSS feed containing information about the current incidents that were occurring on the roads in Scotland and all of the planned roadworks that were upcoming. The details that are provided about the current incidents within Scotland at the time of accessing them is the road that the incident has occurred and the area that the road is located in, also of the road has been closed it will state where the closure begins and where it ends and example of this is “M8 J4 (E Whitburn) - J3a (Bathgate) – Closure”. It also provides a description of the incident another example of this is, “The M8 is closed Eastbound at Junction 4 to allow for overnight roadworks. Motorists are advised to use the available signed diversion route and should allow extra time for their journey”, it also provides a link which goes to a page which gives much greater detail about the incident including the start time, if there is a diversion route available and the expected duration/end time. It also provides the longitude and latitude of the incident and the publication date that the incident was inserted into the feed.

All of this data is contained within XML this means that it had to be parsed to make it easy to read for the user, by removing the tags and information that is not required by the user. This is done by making use of XMLPullParser which collects all of the incidents by using the <item> tag which is used to show that it is not part of the structure of the page. It is then used to gather all the information that will be displayed by the application, this consists of the <title> which states the road that the incident is on, the <description> which is a greater detailed narrative of the incident including the reason for the incident. The link to the page which gives even greater detail will also be provided for the user, as well as the publication date so that they are aware of when the incident occurred.

The planned roadworks page provides the same things as the incidents, the information that will be displayed by the application will be the <title>, <description> and <link>. The slight change about the description is that it contains the start date and end date of the roadworks and some other information including who is performing the works and the traffic management information including the speed limit and diversion information.

This application will consist of three pages(activities), the first page will consist of three buttons. The reason that buttons were chosen to be implemented in the application is due to the tidiness that they provide and the ease of use that the user will experience will use in comparison to some other components for example in an earlier build I made use of radiobuttons but after using the application a few times myself I felt that it could lead to it feeling quite clunky and could be confusing for some users which information is being displayed to them. I also felt that making use of them lead to a less aesthetically pleasing experience for the user and another reason that buttons were chosen instead of radiobuttons is because if the requirements of the application were to be updated or changes it would require a greater deal of the code to be changed which would allow for there to be more than two options for the user to select from for example if the application required for current roadworks to be displayed in it. Here is an example of what the application looked like when it made use of the radiobuttons:



The first button performs the parsing of the xml and leads to the page that will display the information about the current incidents, the second button will also perform the parsing for the planned roadworks page and then display all of the upcoming planned roadworks, the third button is an exit button which stops any of the activities that are running and closes the application. The second page that will be loaded up once the user has clicked the first button contains a SearchView which will be used to allow the user to search for a certain road or area that an incident has occurred on to be able to find any incidents that could impact their journey. This was important to implement to ensure that the application met the requirements that were set out which included allowing the user to be able to search for a certain road to find incidents. This page will also contain a ListView is a display area that will be used to show the data in a clean fashion, this was chosen because it allowed for the data to be easily entered into the area in contrast to some other components that could have been used. The way that the list will be populated will be by using an ArrayAdapter in which it will convert an array of strings into view items which can be loaded into the container.

The page that will be displayed after the user has clicked on the button that will gather and parse the data from the feed regarding the planned roadworks will be very similar to the current incidents page making use of the SearchView again to allow the user to search for a certain road or for a date that roadworks could be taking place on. It will also contain a ListView which will be again used to contain the data in a clean fashion and will allow the user to easily scroll through all of the results that are returned from the RSS feed.

When the list is displayed the user should be able to click on the link of each item that is returned from the feed and they should then be taken to the link that is included in the feed which takes them to the page which goes into greater detail for example in the current incident scenario it displays when the incident is expected to finish. This will be of great help to the user as they will now be able to plan their journey time as they will know when a certain road will reopen which could save them time instead of waiting in the diverted traffic.

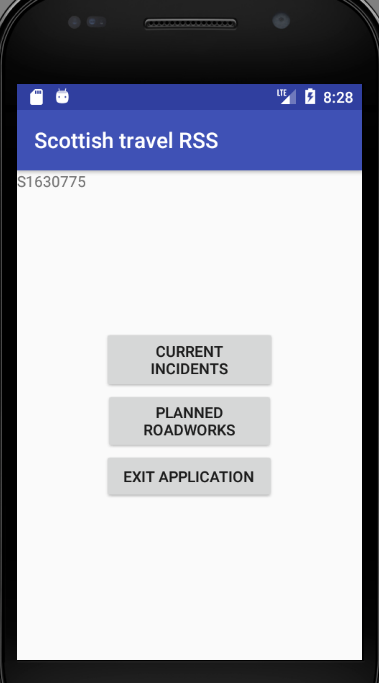
The application as a whole will have a very similar simple theme throughout, the reason for this is so that it is easy for the user to become acquainted with the application and so that it is not required to have extensive computing skills and most users will find it easy to figure out how to make use of the application in a short period of time. Another benefit of having a simple design is that it will not need a lot of processing power to load a lot of components that are unnecessary for the applications needs. It also provides a greater deal of order in comparison to using other components which can end up being messy for example with more than one item on a line, which could be very confusing for the user.

A component that I considered using was a splash screen when I had the radiobuttons implemented the splash screen was useful because the data would load straight away so if it took a while to load making use of a splash screen allows for the data to be loaded in and displayed. So when the design was changed to buttons the splash screen was no longer necessary as it would no longer have to read and parse the data straight away.

Testing Report

Testing is an important part of the software development process and the testing strategies that will be used to test the mobile application that has been developed to ensure that it meets the requirements specified by the user and various other factors such as performing its functions in an acceptable time so that the user is not waiting for the programme to load another reason that testing is important is to ensure that it can be installed and run in its intended environment. It is also used to find bugs or faults within the application that are unseen until the programme has been ran, performing testing can also create bugs within an application.

The types of testing that will be carried out during this testing phase will be smoke testing and black box testing. Smoke testing is testing that is performed to ensure that a program is ready to be executed and there are no faults big enough to stop from releasing a software. There are various questions that performing smoke testing asks for example, “does the user interface load?” is relevant to the mobile application which has been created because if the user interface does not load properly then it is impossible for the rest of the application’s features to function. Smoke testing is very beneficial as it is a quick test and does not need a lot of time to perform or get the results like some other testing methods might. Another way that smoke testing is by making use of try/catch when using the XMLPullParser, where if an error is found it will display the Class name that the error is found, then it will display the error type and message which means that it is possible to find the solution to the error easily if it has been solved before. Here is an example of a smoke test where it is proven that the user interface correctly loads which means that further testing can take place:



Behavioural Testing

Another type of testing that was used nearer the end of the development process was behavioural testing, which is where the user is told a use case to follow and they are also given a set of expected results which should match each action that the user performs. This is also called black box testing because the tester should not care about the code that leads to the answer that they get, they should just care if it is the same as the expected result that they are given and if it is not the same as the expected result then it is assumed that there is something wrong with the code or in the external dataset. Below is the behavioural testing that was performed on the application:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Use Case | Expected Result | Actual Result | Proof | Comments |
| Select Current Incidents button | Changes to the incidents activity screen and display all of the current incidents that are in the rss feed | Successfully changes screen and displays the current incidents parsed from the rss feed. |  | N/A |
| Search for road “A737” | Should only display all the items in the list which contain A737. | Successfully searches for all roads which contain A737 and displays them. |  | N/A |
| Click on the link to be taken to the website which gives greater detail about the incidents. | Didn’t manage to implement properly in time. | Fail | N/a | N/a due to time constraints and difficulties. |
| Select Planned Roadworks button. | Changes to the roadworks activity screen and display all of the current incidents that are in the rss feed | Successfully changes screen and displays the current incidents parsed from the rss feed. |  | N/a |
| Search for road “A82” | Should only display all the items in the list which contain A82. | Successfully searches for all roads which contain A82 and displays them. |  | N/A |
| Click on the link to be taken to the website which gives greater detail about the incidents. | Didn’t manage to implement properly in time. | Fail | N/a | N/a due to time constraints and difficulties. |

Landscape

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Use Case | Expected Result | Actual Result | Proof | Comments |  |  |  |  |  |
| Select Current Incidents button | Changes to the incidents activity screen and display all of the current incidents that are in the rss feed | Successfully changes screen and displays the current incidents parsed from the rss feed. |  | N/A |  |  |  |  |  |
| Search for road “M8” | Should only display all the items in the list which contain M8. | Successfully searches for all roads which contain M8 and displays them. |  | N/A |  |  |  |  |  |
| Click on the link to be taken to the website which gives greater detail about the incidents. | Didn’t manage to implement properly in time. | Fail | N/a | N/a due to time constraints and difficulties. |  |  |  |  |  |
| Select Planned Roadworks button. | Changes to the roadworks activity screen and display all of the current incidents that are in the rss feed | Successfully changes screen and displays the current incidents parsed from the rss feed. |  | N/a |  |  |  |  |  |
| Search for road “A82” | Should only display all the items in the list which contain A82. | Successfully searches for all roads which contain A82 and displays them. |  | N/A |  |  |  |  |  |
| Click on the link to be taken to the website which gives greater detail about the incidents. | Didn’t manage to implement properly in time. | Fail | N/a | N/a due to time constraints and difficulties. |  |  |  |  |  |