Mobile Platform Development

Ryan Graham Cobb – S1630846

Complete Coursework Submission

# Project Design

## Project Overview

Per requirement of the Mobile Platform Development coursework a Mobile application compatible with Android Marshmallow(6.0) is to be developed to visualize the Traffic Scotland XML API for Incidents and Roadworks from the site.

This document details several factors of the Mobile Development Project and are listed as follows;

* Project Overview—Details of the Project summarized and the layout of this document for navigation.
* XML Feeds—details of the feeds being utilized inside the project and what they contain.
* Design templates—Details the overall design of the project with example layouts and planned features for all visual aspects.
* Coding templates—Details the background code structure with what resources are intended to be used and the layout of the code blocks.
* Testing Report—Details the testing methods planned for the project as well as actual Testing implementations in the finished project.
* Summary and Short-Comings—This section will summarize the development of the project and details the overall shortcomings of the development which has limited the overall development.

### Current Incidents Feed (“http://trafficscotland.org/rss/feeds/currentincidents.aspx”);

The Current incidents feed displays a list of XML Items pertaining to the current incidents recorded by the Site, these Items relevant to the project contain details as to the incident in the following format;

* Title (Affected road name, Secondary Affected road name, status of road)
* Description (Further details pertaining to incident)
* Georss:Point (Latitude and Longitude coordinates)
* PubDate (Publish Date)

This Feed will be parsed to display the above details in this order and disregard the rest of the variables. Visualising this data will utilize several components of Android Studio.

### Planned Road Works (“http://trafficscotland.org/rss/feeds/plannedroadworks.aspx”);

The planned road works feed details all planned construction on roads logged from the Traffic Scotland, these can detail from road maintenance tot third party construction or other services. These details relevant to the project as listed as follows;

* Title (Affected road name, Secondary Affected road name, status of road)
* Description (SubTags: Start Date, End Date, Works)
* Georss:Point (Latitude and Longitude coordinates)
* PubDate (Publish Date)

This feed will be parsed to display the above details in this order and disregard the rest of the variables. Visualising this data will utilize several components of Android Studio.

## Design Templates

### Overview

Design Templates section will detail the overall design this development aims to achieve, firstly the Developer Platform utilized will be Android Studio 3.0 for Developer Home usage. The project build will be constructed to Android Marshmallow(6.0) per requirements of the development.

This development will utilize the Android Studio’s activity design to develop separate interfaces for each element of the project. These elements include the Home Session, Current Incidents Session, Planned Road Works Session and the Locate Maps session.

### Home Session

The Home session interface will have little functionality and will serve as the Main Menu to switch between other interfaces. This Interface will link to all other interfaces and possess returns on the other interfaces to return to this page. The core functionality of the Home Session is navigation.

The home session will provide Navigation through buttons, the “Current Incidents” will relocate the user to the Current Incidents Page to view the visualized pulled data on the interface.

The “Planned Road Works” will relocate the user to the Planned Road works Interface which will display the visualized pull data on the interface.

The “Locate on Map” will relocate the user to the Google Maps function which when data is clicked from the other Interfaces will display their location from there Georss Points in the XML.

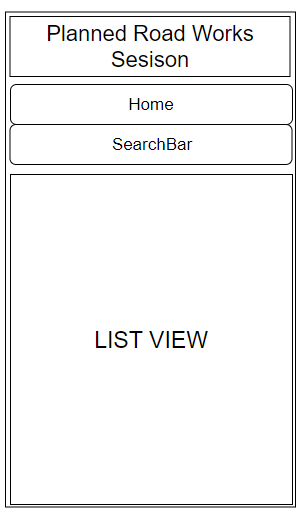
The “Exit App” will close the application with a dialogue message for reference.

*Figure 1: Home Session Design*

The primary reasoning for separate Activities with a main menu is that is follows a conventional practice of navigation making it simple to use and learn, it is also a great help in development to separate code blocks to allow for more modular construction. A series of text relevant buttons will also be more useful than labelling the whole interface to reduce screen clutter and the incredible slow emulator load times on Android Studio.

### Planned Road Works Session

The Planned Road Works session will visualize the pulled XML data into a list for the user to view which when clicked should load the Google Maps data in the other session. A Home button will also exist in order to return to the Home Session to resume navigation. There will also be a search function to shift through the data.

The “Home” Button will return the user to the Main Home Session to resume Navigation of the other material.

The “SearchBar” will allow user submitted data to modify the result of the output data in order to find the relevant parts of the data for the User.

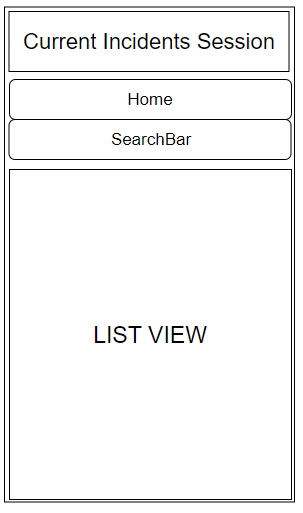
The “ListView” is a display section which will output the Data into a Simple styled List of the pull XML data, this will be setup in a simple fashion so that the Search function will complement the data and not confuse the User. There will also be interactive functions for each item of the list to Send a data packet to the “Locate on Maps” session which will use this packet to display the location of the Planned Roadwork.

*Figure 2: Planned Road Works Interface Design*

The simplicity of the Item display interface is imperative for slower loading times, reducing screen clutter and the amount of assets needed to be loaded will improve usability. The interface design will be shared for the other item list as well to reduce development time and background coding to a minimum for efficient design

### Current Incidents Session

The Current Incidents session will visualize the parsed incidents feed from the website links into a list view for the user to view which when clicked should load the Google Maps data in the other session to view the location of the Incident. A Home Button will also exist to return the user to the Home Session to resume navigation of the app. There will also be a search bar function to sift through the data based upon user Input.

The “Home” Button will return the user to the Main Home Session to resume Navigation of the other material.

The “SearchBar” will allow user submitted data to modify the result of the output data in order to find the relevant parts of the data for the User.

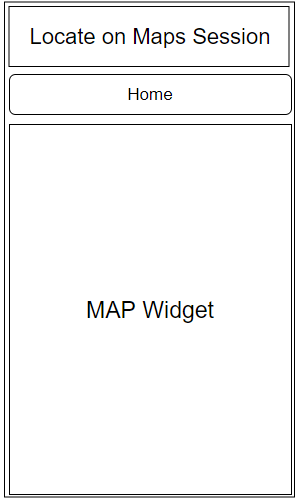
The “ListView” is a display section which will output the Data into a Simple styled List of the pull XML data, this will be setup in a simple fashion so that the Search function will complement the data and not confuse the User. There will also be interactive functions for each item of the list to Send a data packet to the “Locate on Maps” session which will use this packet to display the location of the Planned Roadwork.

*Figure 3: Current Incidents Session Design*

The Current Incidents and Planned Road Works session will share very similar visual attributes in order for a simpler overall design interface which will be better for the user and overall project development time requirement. Although the Visual aspects are similar the background code will differ from which site the ListView data is populated from.

### Locate on Maps Session

The locate on Maps Session is a Google maps interface to resolve Latitude and Logitute from the XML files into a tracked visual position on the map. Both Current Incidents and Planned Road works poccess these Lat/Long values which will allow this app to target the location on the map for the user.



The “Home” button will return the user to the main menu Home session interface to resume navigation of the application.

The “Map Widget” is the Google Map view which will allow the display of the Latitude and Longitude to a visual marker on the google maps. This will be activated by clicking on the List View Item coordinate from the other Interfaces.

*Figure 4: Maps Session Design*

## Coding Templates

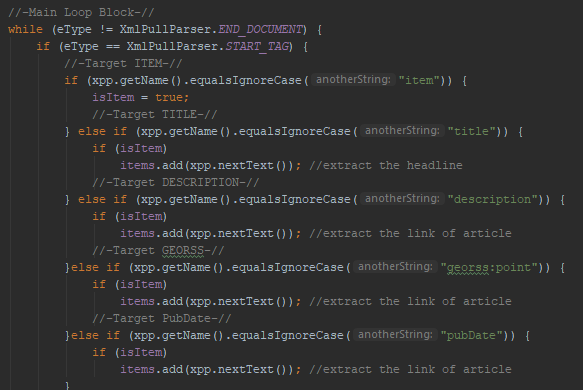
### Overview

The coding templates section will detail the proposed backend functionality of the project, within this development the Android Studio 3.0 package will supplement the bulk of main functionality of visual interfacing with the exception of a Google Maps API import to visualize the mapping functionality in the Locate on Maps Session.

For the Main data Puller, the Project will utilize the Java “XML Pull Parser” (‘https://developer.android.com/reference/org/xmlpull/v1/XmlPullParser.html’) which will pull the XML data into a String Reader which will be stored inside an Array List in order to print the data from the Code to the Interface.

### XML Pull Parser

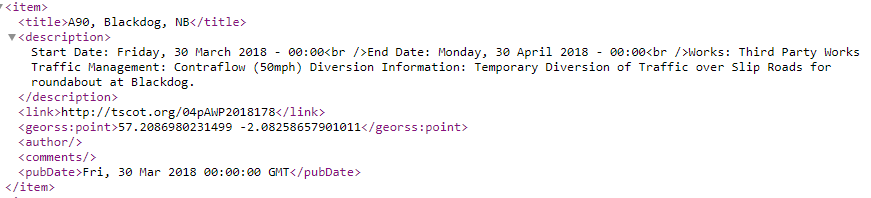
The XML Pull Parser is a powerful tool as it contains a wide variety of tools to parse XML files or Streams into Java Variables. The XML Pull Parser Can Separate the XML code block into its list Items and their Contents, like so:



*Figure 5: XML Puller Extract from Developing Project*

The Pull parser will allow the full break down of the XML data feed into the individual components of the Items tag. The core Item being to create a Java “ArrayList<ObjectClass>” and Array list of objects bound to the items inside the XML. This will allow separation of the items shown in Figure Six, which can develop further detail in the interface section.

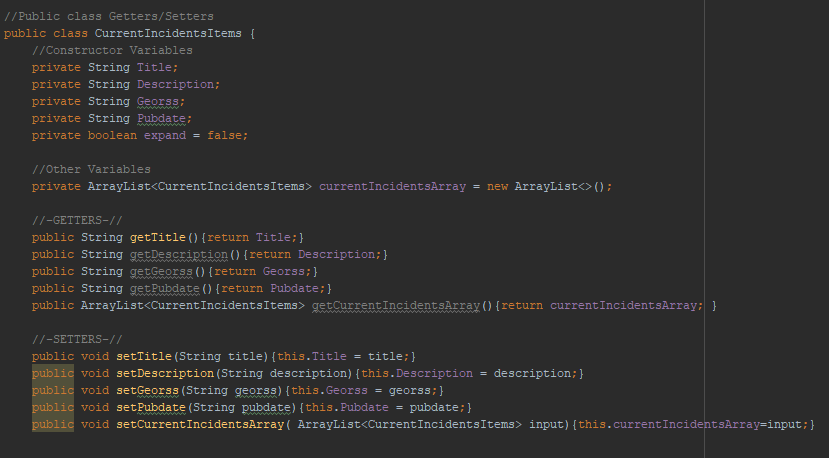
The Object Class will hold variables such as the Title, Description(Sub-tags for Planned Road Works), Georss:Point or Lat/Long Coordinates and finally the Publish Date.



*Figure 6: Extract from Traffic Scotland Feed*

*Src:* *http://trafficscotland.org/rss/feeds/plannedroadworks.aspx*

The Object class example sample shown in Figure 7, demonstrates an example Object class which will be used to store the information of the Pull Parser which will aid in building a List of Objects to output to the List View into the interface. Java Getters and Setters will be defined for every attribute in the XML as shown in the example below.



*Figure 7: Extract of developing code in project, Sample Object Class*

The XML Pull parser section will pull the Object list into the main section to display a list of Objects or items on the interface which should allow for furthering in-depth design elements to the project.

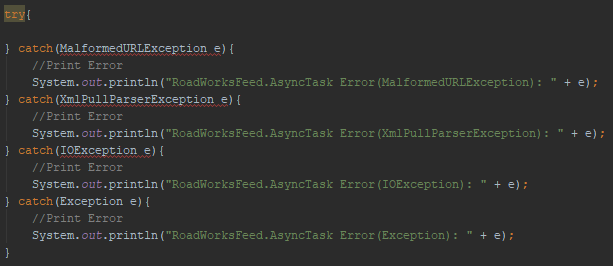
# Testing Report

## Overview

The testing report section will divide the Testing strategy to develop the project and the documented results from certain testing results.

### Testing Strategy

The testing Strategy for a developing project will mostly take form of Black Box and Smoke Box testing in the main code for several reasons. Other testing strategies which operate on stressing user input are not required in this project as the only user enterable section to the project will be the Search Bar which is already fault tolerant to a degree further than the scope of the project, as this section is the only section of user input the code behind the scene must be developed thoroughly to avoid programmer designed errors due to poor or ineffective programming.

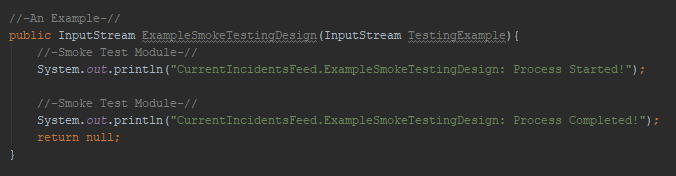
For the Smoke Testing, mostly which will be implemented in the XML Puller Classes inside a Try/Catch statement with extensions in the Catch to diagnose the problem more thoroughly, an example of this below; 

*Figure 8: Example of extensive Testing enforcement in Code Block*

The Try/Catch is among the most popular method for real time error handling as it preventative against applications crashing while in operation and can allow for unexpected errors to be handled and displayed for the programmer to make modifications to resolve the problem.

Displaying the error Block with relevant naming is critical, in this case the (Class Name, Definition Name, Error Type and Error message), variable naming and Error messaging are equally important for development as it reduces the time spent fixing problems and issues as well as improving diagnostic times to identify the problem. Within the project sections of critical code will be embedded inside Try/Catch blocks to ensure continued operation in the event of Android based issues arising.

Further Smoke Testing is implemented in the beginning of each Class and Class operation called to see the working code deploy, this is a visual aid to identify the flow of calls inside the program which will also reduce time spend searching for the cause of the Error. This will be implemented as shown;



*Figure 9: Example of Smoke Testing normal applications*

Further on, the only applicable testing method is Black Box or Behavioural Testing which involves systematically redesigning code tolerance to fault. This is particularly difficult for any developing in Android Studio due to enormous strain of building a emulator for each testing run. This is a problem which limits development.

### Test Casing

|  |  |  |  |
| --- | --- | --- | --- |
| Home Session | | | |
| Casing | Expected Results | Actual Results | Comments |
| Planned Road Works Button Selection | Change Activity page to the Current Planned Road works without Crashing | Successful | Operates as Expected. |
| Current Incidents Selection Button | Change Activity to the Planned Road works without Crashing | Successful | Operates as Expected |
| Locate on Maps |  |  |  |
| Exit Application | Close the APP | Successful | Operates as Expected in test build. |

|  |  |  |  |
| --- | --- | --- | --- |
| Planned Road Works Session | | | |
| Casing | Expected Results | Actual Results | Comments |
| Pressing Home Button | Return to the Home Menu | Successful | Operates as expected |
| Using the Search Bar | Alters the List view to the user specified input, | Successful | Works as intended, however the search looks through the whole text rather from left to right like conventional searches this is to allow the search to adapt to user input more dynamically. |

|  |  |  |  |
| --- | --- | --- | --- |
| Current Incidents Session | | | |
| Casing | Expected Results | Actual Results | Comments |
| Pressing Home Button | Return to the Home Menu | Successful | Operates as expected |
| Using the Search Bar | Alters the List view to the user specified input,  (Unfortunately, at Time of Testing there where no Incidents) | Successful on previous search build. | Works as intended |

|  |  |  |  |
| --- | --- | --- | --- |
| Mas Session | | | |
| Casing | Expected Results | Actual Results | Comments |
| Pressing Home Button | Return to the Home Menu | Successful | Operates as expected |
| Map view to display the Location from the Lat/Long point from Getters/Setters |  | Unsuccessful implementation, despite efforts the Google Map view Does not work as intended… | Project was unable to implement Successfully the MapView despite attempts |

# Summary

Mobile Application development within this project has been extremally challenging mostly due to the difficulties of utilizing Android Studio as opposed to other interface designers. The Project however does not meet several of the more advanced functions, this is due mainly to the dramatically shorted time span to develop the project alongside other works which has made the overall development difficult.

During the development the Object List design was scrapped in favour of String Array due to time constraints alongside development issues which were not recoverable. The Map API function was to dynamically change based upon which Georss Point was clicked which was to be implemented with Getters and Setters in the main application which there is a very primitive design of this inside the main application as of current.