Mobile Platform Development – Coursework Submission 1

Gavin Walter Jaap

S1512251

Github: <https://github.com/Gavinja/MPDCWK1512251>

.Apk File: <https://github.com/Gavinja/MPDCWK1512251/blob/master/app-debug.apk>

Github Video Link: <https://github.com/Gavinja/MPDCWK1512251/blob/master/GavinJaapMPDCoursework.mp4>

Contents

[Design Report 3](#_Toc5963487)

[Introduction 3](#_Toc5963488)

[Design 4](#_Toc5963489)

[Main view (Application Startup) 4](#_Toc5963490)

[List view 4](#_Toc5963491)

[Maps view 5](#_Toc5963492)

[Landscape and Portrait Layouts 6](#_Toc5963493)

[User Interaction 7](#_Toc5963494)

[Conclusion 8](#_Toc5963495)

[Testing Report 9](#_Toc5963496)

[Introduction 9](#_Toc5963497)

[Functional testing 9](#_Toc5963498)

[Functional testing outcomes 9](#_Toc5963499)

[Functional Testing Results 10](#_Toc5963500)

[User Acceptance Test 10](#_Toc5963501)

[Results from User Acceptance Test 10](#_Toc5963502)

[Test Strategy 11](#_Toc5963503)

[Test: Main Activity View 11](#_Toc5963504)

[Test: List Activity View 11](#_Toc5963505)

[Test: Map Activity View 13](#_Toc5963506)

[User Acceptance Testing 13](#_Toc5963507)

[Feedback 14](#_Toc5963508)

[Figure 1 mpd Main Activity View 4](#_Toc5956217)

[Figure 2 mpd List Activity View 5](#_Toc5956218)

[Figure 3 mpd Map Activity View 6](#_Toc5956219)

[Figure 4 mpd landscape Main Activity View 6](#_Toc5956220)

[Figure 5 mpd Landscape List Activity View 7](#_Toc5956221)

[Figure 6 mpd Landscape Map Activity View 7](#_Toc5956222)

# Design Report

## Introduction

The British Geological Survey (BGS) is an organization that aims to advance geoscientific knowledge of the United Kingdom’s land by surveying, monitoring and researching. Their website has given out an RSS feed that contains data about earthquakes that have occurred within the United Kingdom.

Using this RSS feed the goal of the project was to develop a responsive and easy to use android application that will parse the RSS feed and turn it into information that is useful and easy to understand. Another requirement is to add a way to filter through the list of earthquakes to find specific information.

The project also requires that both landscape and portrait layouts are formatted and designed with the user experience in mind.

## Design

### Main view (Application Startup)

One of the main primary goals was to ensure that the user would be able to easily navigate and use the application, to achieve this the main activity that will launch once the application is opened and the user will be able to see a navigation bar at the bottom.

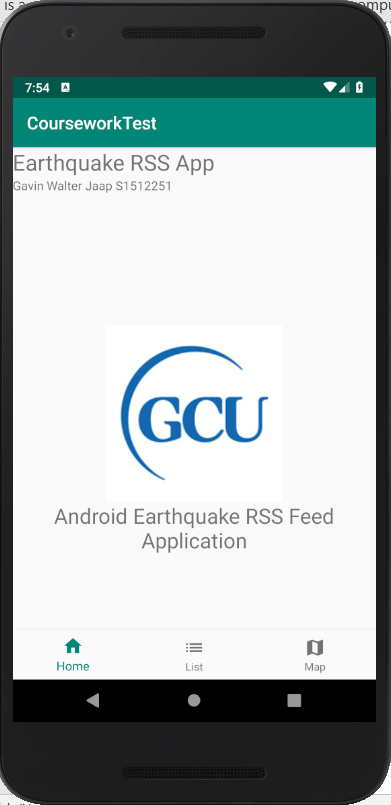


Figure mpd Main Activity View

The design of the navigation bar was chosen as it provides three options at a glance and allows the user to understand where they are within the application and where they can go. Another reason the navigation bar was chosen is because it is small and scales well while changing from portrait to landscape.

### List view

The purpose of the list activity screen is to provide the user with the RSS feed along with optional filters that they can use to filter the list. To achieve this a recycler view was used to display a list of values obtained from the British Geological Survey RSS Feed (Available here: <http://quakes.bgs.ac.uk/feeds/MhSeismology.xml>)

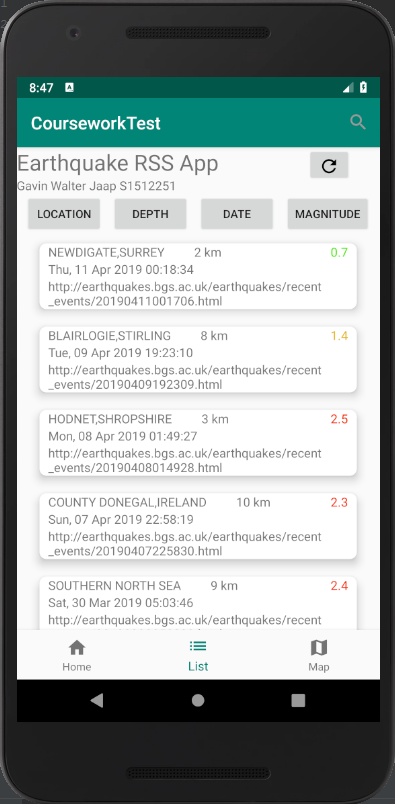


Figure mpd List Activity View

Within the application we can see that the list is there within a recycler view, the primary reason that the recycler view was chosen over a ListView was the ability to develop a ViewHolder pattern which defines which data is passed into the recycler view, this allowed for parsing values from the RSS feed to become easier as a result as data could be picked and chosen from the RSS feed and added to recycler view. Another reason that the recycler view was chosen was due to the OnItemTouchListener which allows for clicks on particular items within the recycler view, with the OnItemTouchListener it gives more options to the user than the OnItemClickListener such as easy movement between different activities and fragments.

The List activity view also contains a refresh button that allows users to easily refresh the earthquake recycler view and receive new data if the RSS feed has been updated.

4 Buttons were also implemented that are to be filter selectors however they have not been programmed to work at the current time, these filters would work by switching what the search bar looks for within the recycler view so if users wanted to find earthquakes by depth they would click on the depth button which would switch what the search button was searching for, by default it would search for location names.

The Items within the recycler view can be clicked and that will display the location of the earthquake on google maps.

The magnitude is colour coded so that if it is less than 1 it is green; if it is greater than 1 but less than 2 it is orange and if the magnitude is greater than 2 then it is red.

### Maps view

The maps view allows users to see on google maps where an earthquake has occurred using the latitude and longitude from the RSS feed.

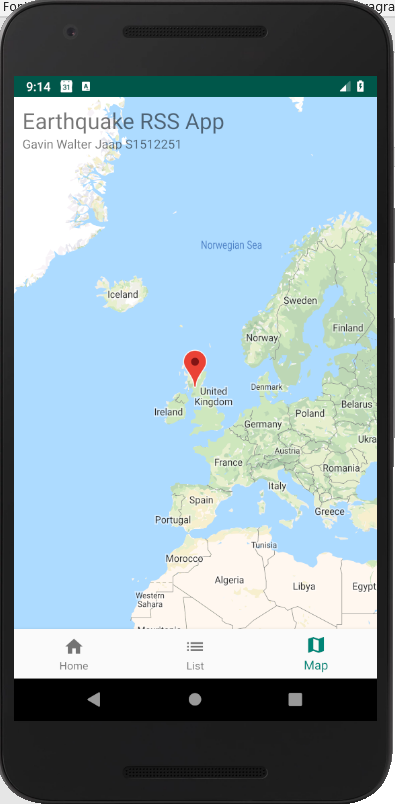


Figure mpd Map Activity View

Within this a user would be able to see precisely where an earthquake has occurred by displaying a marker where earthquakes occur and then be able to click the marker to get more details, this is not implemented as of this submission.

### Landscape and Portrait Layouts

All views have got two different designs, a portrait view and a landscape view. This is to ensure that users can view all the information in all views figures 4,5 and 6 show the landscape varients of the portrait views shown in figures 1,2 and 3

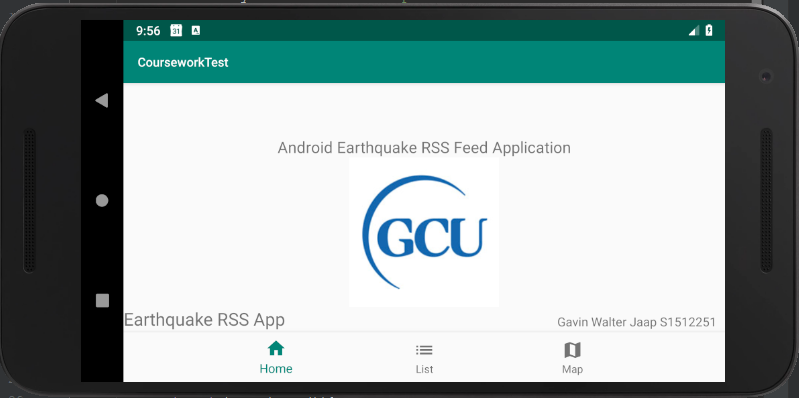


Figure mpd landscape Main Activity View

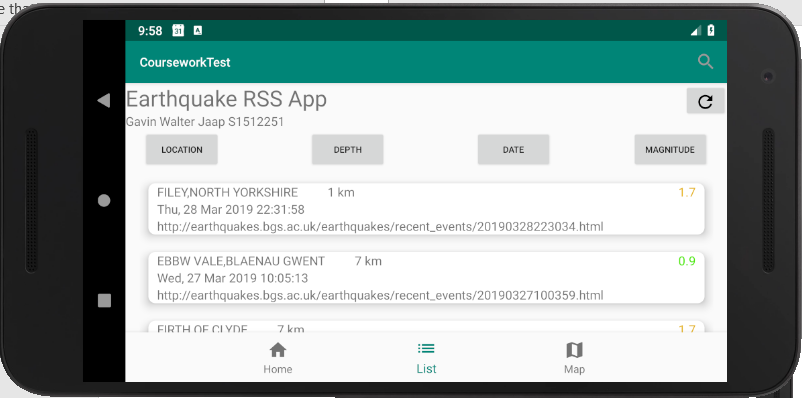


Figure mpd Landscape List Activity View

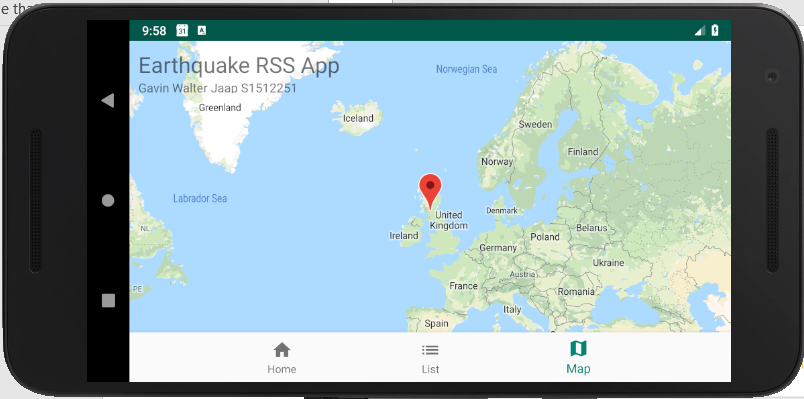


Figure mpd Landscape Map Activity View

### User Interaction

Within the application users should be able to look at the application and understand how the application works and not be confused by an oversaturation of information as well as giving users the ability to navigate throughout the application with ease.

The user experience with this application should be positive with its simplistic and easy to understand design, this can be seen from the use of a navigation bar that allows users to easily and seamlessly move between different views. Another way that the design is made to promote a good user experience is labelled buttons that explain what it is they do and are placed appropriately within both landscape and portrait view. Some work however needs to be done for the List Activity View Buttons, there are too many of them and they could be adapted to work as a spinner so the user clicks on the spinner and selects what they would like to filter by.

## Conclusion

Overall, this application does provide some functionality outlined within the specification such as getting an RSS feed from the British Geological Survey and parsing it through to a Recycler view and displaying the information in a meaningful way such as colour coding the magnitude to colour in depending on its number. However, a lot of functionality was not implemented within the application such as a meaningful filter and the implementation of the google maps. In retrospect more time should have been dedicated to ensuring the filters were implemented within the list allowing users to easily traverse the list enhancing user experience along with having google maps properly implemented to display information easier.

# Testing Report

## Introduction

Within this Project, 2 methodologies are being implemented to test the application to ensure a wider range of understanding of the parts of the application that work within this project and the parts that don’t.

Methodology 1: Functional testing

To test the applications functionality, white box testing will be used. White box testing involves the tester having knowledge of how the application works. The test for this will take inputs that are designed around the application architecture. This testing method allows for the inputs and outputs of the application to be properly tested and examined.

Methodology 2: User Acceptance Testing

This testing method involves have testers evaluate their overall feelings about the application and discuss what issues they may have with it.

## Functional testing

With in the Functional testing as described above, white box testing will be used to test the applications functionality. To do this a Table will be made for each functionality to check if there is faults or issues with the application.

The table will contain:

* A Id for the test
* A Description of the test
* The input required for the test
* Expected outcome
* Actual outcome

Tests that the expected outcome does not equal the actual outcome will be considered a fault and added to a list containing all faults.

## Functional testing outcomes

1. Home screen

The tests here checked that the navigational bar worked and that the orientation could change from portrait to landscape seamlessly

1. List screen

The tests carried out here checked the navigational bar and the orientation again but included more tests such as

* 1. Search bar: tested by user typing in the search bar, by default location is set to be the filter
  2. Filter selected depth: tested by user selecting depth and typing in the search bar
  3. Filter selected date: tested by user selecting date and typing in the search bar
  4. Filter selected magnitude: tested by user selecting magnitude and typing in the search bar
  5. Refresh: tested by user clicking the button to make sure that the RSS feed refreshes
  6. Navigating to map by clicking on the RSS item: tested by user clicking on the RSS feed

1. Map screen

The tests here were to check the functionality of the map view, the map view would be selected and a user would try and interact with it in different ways such as clicking on a marker

### Functional Testing Results

The results from the functional testing showed that there is a sever lack of functionality in the map view and the list view, all testing on components that were developed worked as expected.

## User Acceptance Test

Within these tests, two testers who are acquaintances of the developer were given tasks and then asked questions based on their experience with the application. Some of the feedback showed some HCI issues with the application as one of the testers has difficulty with eyesight

The tasks the testers were asked to perform was:

1. Get the list view
2. Find earthquake by location
3. Find earthquake by depth
4. Find earthquake on google maps

### Results from User Acceptance Test

The results were overall negative towards the application, lack of functionality makes it difficult for the user to filter through the RSS feed to find information relevant to them and makes it hard for the testers to say much more than, it doesn’t have enough functionality. One concern brought up was the size of the text and buttons being difficult to read for people with poor eyesight.

# Test Strategy

## Test: Main Activity View

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Id | Description | Input | Expected Outcome | Actual Outcome |
| MAI1 | List view clicked on nav bar | Click on list view button | Change to activity list view | Expected outcome |
| MAI2 | Map view clicked on nav bar | Click on map view button | Change to activity map view | Expected outcome |
| MAI3 | List view in landscape | Rotate device from portrait to landscape | Change view from the portrait version to the landscape version | Expected outcome |

## Test: List Activity View

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Id | Description | Input | Expected Outcome | Actual Outcome |
| LIS1 | Home view clicked on nav bar | Click on home view button | Change to activity home view | Expected outcome |
| LIS2 | Map view clicked on nav bar | Click on map view button | Change to activity map view | Expected outcome |
| LIS3 | Get RSS feed from British Geological Survey | Nothing | Displays fetching RSS feed on start and displays it in a recycler view | Expected outcome |
| LIS4 | Refresh button is clicked | Click on refresh button | Refreshes the RSS feed by fetching the feed again | Expected outcome |
| LIS5 | Search button is clicked | Click on search button | Displays a search bar that a user can type in | Expected outcome |
| LIS6 | Search values typed | Type in values within the search bar | Filters the list by locations containing the same values as the ones that are typed | Unexpected outcome: the recycler view does not change to filter the values that are typed in the search |
| LIS7 | Location Button Pressed | Press Location Button | (On by default) Switches the filter for the search bar to location | Unexpected outcome: is not pressed in by default and does not switch filter. Displays a toast message:  “location is not implemented” |
| LIS8 | Depth Button pressed | Press depth button | Switches the search bar filter to depth | Unexpected outcome: toast message: “Depth not implemented” does not switch filter |
| LIS9 | Date button pressed | Click date button | Switches the search bar filter to date | Unexpected outcome: toast message: “Date not implemented” does not switch filter |
| LIS10 | Magnitude button pressed | Press magnitude button | Switches the search bar filter to magnitude | Unexpected outcome: toast message: “Magnitude not implemented” does not switch filter |
| LIS11 | Scroll through list | Click and drag the list | Scrolls list up and down | Expected outcome: |
| LIS12 | Change to map view from clicking an item on the list | Click item on list | Changes view to map view and displays the location of the item on map | Unexpected outcome: the view changes to map but doesn’t go to the particular place the earthquake is |
| LIS13 | List view in landscape | Rotate device from portrait to landscape | Change view from the portrait version to the landscape version | Expected outcome |
| LIS14 | Get more information by holding in a item within the list | Click and hold an item on the list | A view will appear showing additional information | Unexpected outcome: function not implemented, takes user to map |

## Test: Map Activity View

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Id | Description | Input | Expected Outcome | Actual Outcome |
| MAP1 | Home view clicked on nav bar | Click on home view button | Change to activity main view | Expected outcome |
| MAP2 | List view clicked on nav bar | Click on List view button | Change to activity list view | Expected outcome |
| MAP3 | Display all Earthquakes from RSS feed | Nothing | Displays markers at all earthquakes found within RSS feed | Unexpected outcome: No functionality, Displays only a default set marker hardcoded into the app |
| MAP4 | Display more information about earthquakes | Click on marker | Displays the earthquake information in a fragment | Unexpected outcome: No markers added, functionality not included. |
| MAP5 | Map view in landscape | Rotate device from portrait to landscape | Change view from the portrait version to the landscape version | Expected outcome |

## User Acceptance Testing

The tester was given 4 goals to achieve:

1. Get the list view
2. Find earthquake by location
3. Find earthquake by depth
4. Find earthquake on google maps

### Feedback

Tester 1: Hayden Anderson

* Easy enough to navigate
* Application cannot perform many functions
* Too many buttons in the list view
* More colour coding
* Filter buttons should be easier to understand at first glance
* Maps should be zoomed in more towards the UK
* General layout is fine but more functionality would be needed to make it usable

Tester 2: Alexander Jaap

* Difficulty reading some text
* Design is too small
* Would prefer that the navigational bar would shrink to allow a better view of the parsed RSS
* Application doesn’t do much
* Could use more colour