Mobile Platform Development Reports

Daryl McAllister

S1222204

Github Repository: <https://github.com/mcllsti/Earthquake-Application-Android>

Github Video Direct: <https://github.com/mcllsti/Earthquake-Application-Android/blob/master/Video%20and%20documentation/ScreencastOMatic_S1222204.mp4>

Github Video Folder: <https://github.com/mcllsti/Earthquake-Application-Android/tree/master/Video%20and%20documentation>

Contents

[Design Report 2](#_Toc5049979)

[Testing Report 6](#_Toc5049980)

[Formal Testing Methods 6](#_Toc5049981)

[White Box Testing 6](#_Toc5049982)

[System and Feature Testing 6](#_Toc5049983)

[Testing Areas 6](#_Toc5049984)

[Testing Method 7](#_Toc5049985)

[Informal Testing 7](#_Toc5049986)

[Unit Test 7](#_Toc5049987)

[Integration testing 7](#_Toc5049988)

[Test Documentation 8](#_Toc5049989)

[Informal Junit Tests 8](#_Toc5049990)

[Source Code: 8](#_Toc5049991)

[Outcome: 8](#_Toc5049992)

[Formal Testing 9](#_Toc5049993)

[Test Cases 9](#_Toc5049994)

[Test Outcomes 9](#_Toc5049995)

[Test Data 11](#_Toc5049996)

[Test Figures 14](#_Toc5049997)

# Design Report

This report outlines the design choices undertaken for the request application. The application has the requirements of allowing users to view Earthquake data received from the British Geological Survey data feed as well as the ability to filter them and refine by date. Design was a large focus point for this project as with any mobile device, a seamless interface should be provided to the user to ensure they have the best possible experience within the application.

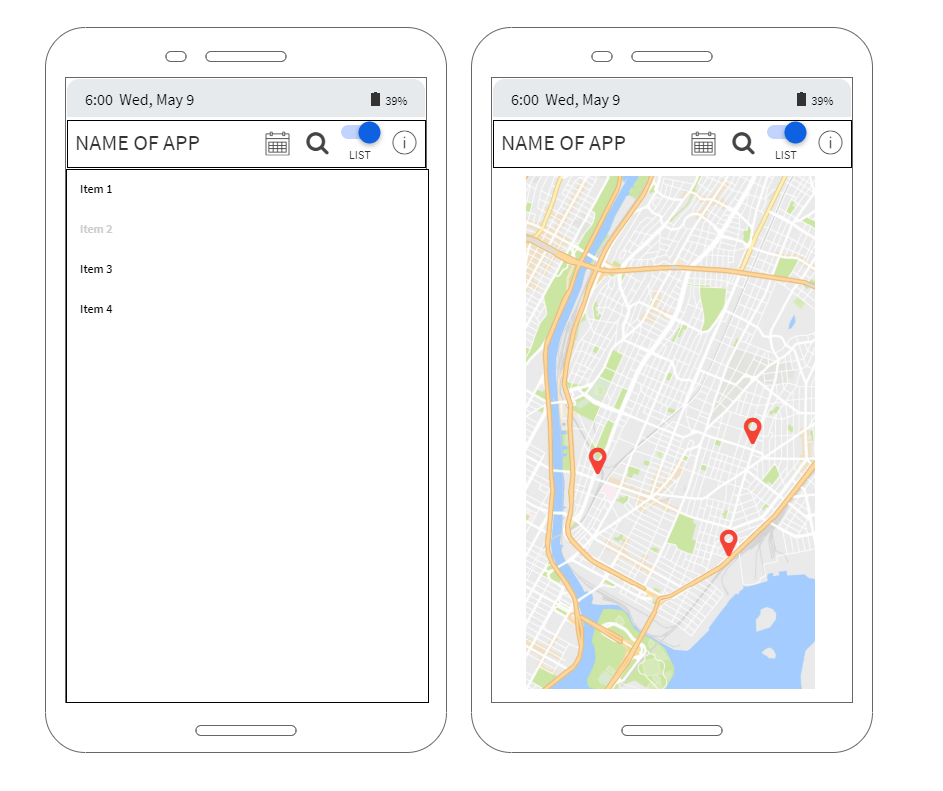
The main overarching decision in terms of design for the application was to “keep it simple” which was chosen for a number of reasons. Firstly the project brief did not outline a potential target audience that would be using the application so by ensuring simplicity made it so that the application could be used by individuals of all tech skill levels and would not require a learning curve to use. Shneiderman's golden rules of design also where a factor as the suggests to strive for consistency and reduce short term memory load could be easily achieved by ensuring the application did not have many activities and menus for users to get lost in or to have to ensure where similar. The best approach decided to keep the application simple but also provide required functionality was to implement an interface that allowed users to seamlessly switch views and data while maintain a constant page not dissimilar to a single page web application. This design was outlined in wireframes to start as seen in figure 1. 

Figure 1 - Wireframes of main activity

Figure 1 shows the intention for users to be able to switch quickly between a list view of earthquakes and a map view of earthquakes which also became part of the final produced product. A quick and seamless ability to change views was also important as at different times users may be more interested in map based location of earthquakes or structured earthquake data and the ability of the application to provide both was important. The final design does not make any large changes from the wireframe designs. The only real change was using custom icons on the switch button to make it fit in with the Action Bar icons more and made it uniform.

The components used to be able to switch views seamlessly was a Frame Layout which contained a list fragment and a map fragment. The use of the Frame Layout was only implemented to ensure the area required for both fragments continues to stay blocked out even throughout fragment changes. The fragments act in a similar manner to partial views for websites in that when shown they fill the Frame Layout with the appropriate view. The fragments are switched at will by the user who does so by clicking the Switch button in the action bar. This more custom design was chosen over an existing element such as the View Flipper for distinct reasons. It was important to give users a clear tactile way of switching the view instead of an arbitrary swipe and the ability to show which view they are looking at now is also important feedback which is again in line with Shneiderman's golden rules. The View Flipper also made it more bulky in terms of code to maintain a user’s current filters and date refine when switching from a list to map or vice versa. The ability to change the fragments from the Action bar was also important as the Action bar was intended to be a driving force for users to manipulate the data they saw on screen and their views so to break up the user from using the Action bar and using the screen swipe was not adhering to constancy.

The list fragment was displayed with a custom adapter for a custom layout. This custom layout left a much better impression for users and also allowed them to quickly see a colour coded magnitude and small relevant earthquake data rather than all data in an unreadable list. The same data cherry picking was used in the Map fragment where positions where not included in the marker dialogues as they were easily viewable on the map. Ensuring that users where not presented with all the data at once was important as it could have been poorly displayed and became much better when given its own Activity to be viewed upon.

The Action bar usage for the application when dealing with the main activity was an important factor as any ability of the user to filter results, search by date or switch view is contained in this bar. This was done so to give users a consistent area of control and acts almost as a nav bar on a website where a user can return to time and time again while being constant. As seen in figure 2, the Action bar contains icons to ensure that users have better visual clues to the actions and keep the bar as small as possible. The icons give the user a quick reference to what is proposed.



Figure 2 - Final Design Action Bar

The more detailed view of an earthquake stayed true to its initial wireframe concept outlined in figure 3. The concept was to combine both elements previously layed out, a map and detailed view, in one to give the user an overview of everything they require on the one activity. This one done in this section as a user should be able to get the overall picture of the specific element they have chosen easily. Unlike the list and map views on main activity which retain a similar layout when transferring from portrait to landscape, the earthquake specific view switches the layout to give a much easier layout to navigate. The use of a dialogue box for users to choose dates was chosen so that a user was not taken out of their current context to choose and instead could quickly choose a date and be right back at the main activity with the earthquakes they wanted to see. This helps improve a user’s experience and stops unnecessary jumping between screens. The refinement option is represented by a search icon which is carried over from early designs of having the ability to have a text search. This was decided against as it may not be clear to users on how to search or what to search for so instead a spinner with options was decided upon with the most popular search refinement options populating it. While this is more limiting for a user, it should make for a better experience and makes the code base less bulky as input handling is not as needed.

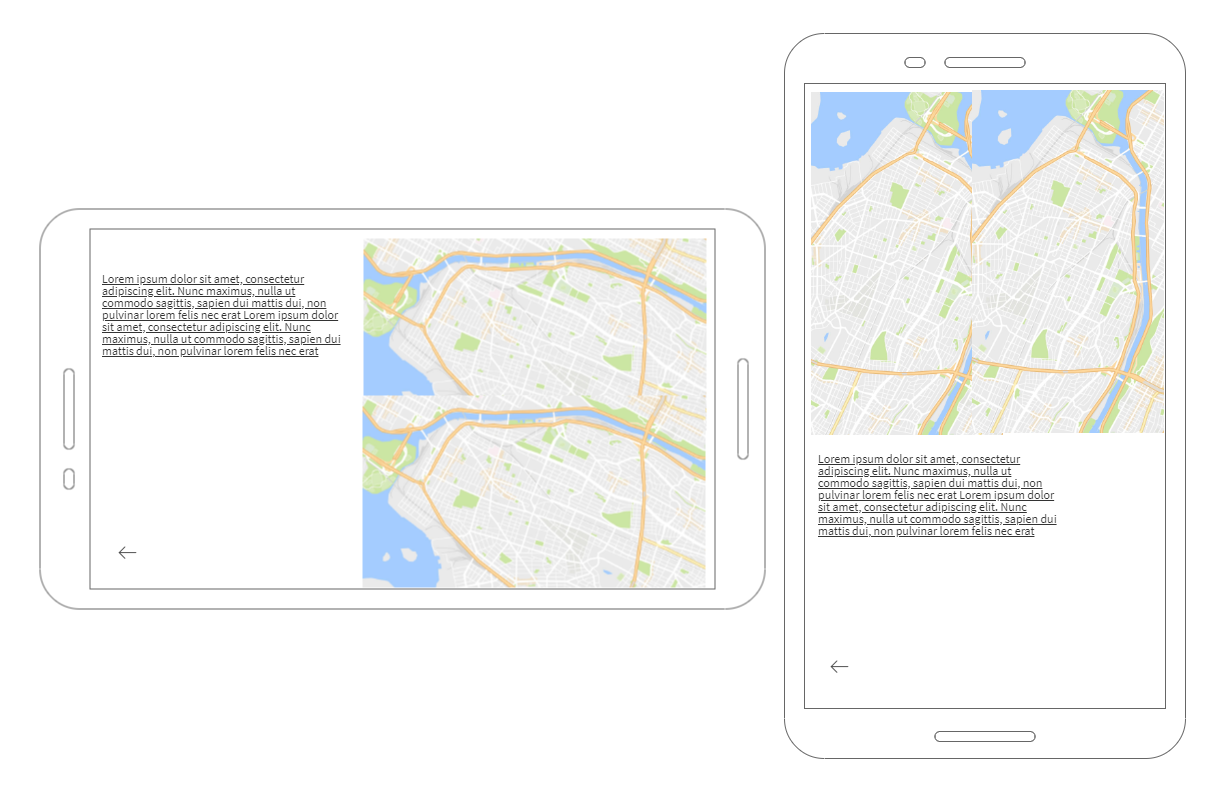


Figure 3 - Wireframes of Earthquake View

In portrait it was envisioned that the map and then data was linear to fit the narrow long profile of portrait but this did not translate well to a horizontal plane. The natural progression was to make it horizontally shifted with the data on the left and map now on the right. This stops the user having to scroll down the screen by having it all available to view. The earthquake specific activity final product differs slightly from the design in that a back button was removed. This was done as all phones have a back button on them and to have an on screen app back button would be counter intuitive to how users use their phone and take up screen space unnecessarily. The text data is displayed using a simple Text View which has the details of the earthquake set programically. The map is instead another Map Fragment which again has the earthquake set as a marker for a user to see a map position for it.

Colour was also a challenging concept to deal with for the application. It was decided that it was important to keep the application largely neutral with whites, blacks and greys proving the bulk of colour. This was done to help users get a quick and clear view and done to also assist any user that may have visual impairments. The neutral colours were also chosen to even further highlight the areas of bright colour which is the magnitude colour coding and map views. These areas where intended to stand out as they were decidedly the most relevant areas that users would want to view so having their eyes naturally caught by them would help users find them and allow them to stand out to a user.

Interaction from users comes almost exclusively in the form of clicks/touches. This was a deliberate decision to once again keep consistency in the application and maintain simplicity. This was also a factor in removing a View Flipper and using the switch instead of change the fragments. Users can click on the spinner, switch, date dialogue, markers and list items to interact with them and since click is a natural instinct for a user, means the application has virtually no learning curve. The only area that uses a scroll is the list view which also was intended to create a sense of many endless earthquakes as no other part of the application can scroll.

Overall the design of the application is what was intended and provides a clear and fluid user experience for individuals. The applications goal to be simplistic to use was achieved in that it is only composed of two activities that still provide a user with all the data and refinement options that they would require. The neutral colours and easy to use Action Bar also help users to quickly find what they require and refine the large list of earthquakes quickly to see more specific aspects. While large parts of the user interface and design are positive, there are areas for improvement in that feedback is provided in Toast messages and to make the feedback more meaningful; dialogue boxes could be used instead. The application also could have utilized colour in more areas as well to help draw eyes to them more such as the icons in the Action Bar which could also be an improvement in the future. Despite these shortcomings, the application fits the purpose and does so with a more than acceptable design.

# Testing Report

This report outlines the decisions undertaken on how to test the Earthquake application as well as the reasons for the decisions. The testing is made up of formal white box testing as well as equally important informal testing that occurred throughout implementation. These two forms of tested are outlined as they both occurred to ensure that application has minimal bugs.

## Formal Testing Methods

### White Box Testing

The main method of the formal testing for the application comes in the form of white box testing which is where the tester has knowledge of the implementation of the program. This method was done due to the limited resources during development and testing that made it not possible to have independent users test the application. This does not limit or bias the test however as the program was tested throughout development in white box method so that correct bugs could be ascertained and the formal white box testing at the end was primarily as a find all to find bugs that where missed in implementation so they could be fixed immediately. The white box testing was done by primarily testing features and requirements.

### System and Feature Testing

The formal testing that took place focused on both system testing where the project is tested as a whole in different environments ( in this case mobile phones ) as well as feature testing where individual features where tested to ensure they met the requirements as laid out in the requirements specification. Both the system and feature testing was undertaken to ensure full robustness of the system and to be exhaustive in the testing process as security is a driving development trend in both android applications but also all software.

### Testing Areas

The initial tests done where basic usability testing was done where the navigation and online/offline handling as these where essential to the applications use. The main requirements where then tested such as the searching features (largest magnitude, smallest magnitude etc.), date refinement and the ability to switch map and list views. The last portion of tests where to check that correct earthquake specifics where shown when chosen by the users.

These areas where tested to ensure the correct refinements in the data that they user specified where both functional and correct. In terms of the largest magnitude, it was tested to ensure that the earthquake shown was the earthquake with the largest magnitude. The date refinement was tested to ensure that non-relevant dates did not return a blank view and made users aware that no earthquakes existed on this day. Aspects that where more difficult to test such as the map where just as important as bugs may have gotten through that could not be spotted and the easiest way to test was to use multiple devices to ensure the output was consistent across each device.

### Testing Method

The testing for the specific areas was done by determining what the expected output for the test and comparing it against the actual output which would then show whether the test passed or failed. This was done for each of the test cases outlined before the main testing begain.

## Informal Testing

Testing occurred throughout development and out with formal testing in many forms such as unit tests or integration tests. These where important as they ensured that the program worked before continuing development and are considered an important testing phase.

### Unit Test

Unit tests where used in the application in order to test the model class “Earthquake” that is used to store a sterilized earthquake. This was utilized for the Earthquake class as it is an important class for the manipulation of earthquakes so the use of unit testing to test methods as they are added was beneficial to quickly integrating them into the project without having to worry about extensive testing on their addition. The methods that make up the Earthquake class are used to parse data from the large data strings and so they have to do so accurately which was an important part being tested. The unit test class for Earthquake did so by having a dummy earthquake set up that would be tested with the known outputs already known and tested against.

This proved effective and allowed for controlled unit testing to be accomplished and could be constantly re-ran if any changes were made. The largest downside to the unit testing is that the project was severely limited with what unit tests could be called and the Earthquake class being the only class that could be independently tested as other aspects such as the activities could not be effectively done so.

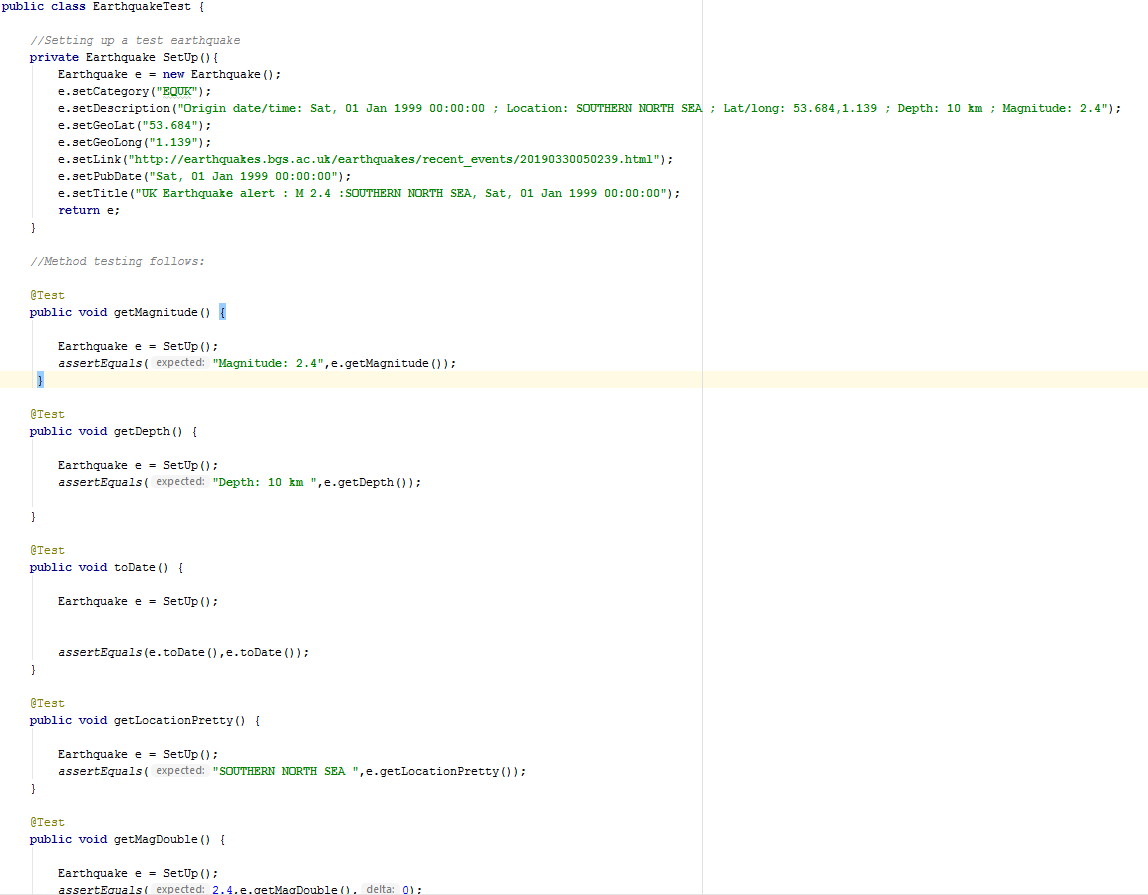
### Integration testing

The application was developed in segments with different features being developed one by one before then being pieced together. An important aspect for the application was to ensure these features worked together when integrated before moving on so that bugs did not pile up and this is why informal integration testing occurred throughout development. This testing was done by completing a development area and then running the application while testing that it was easily transitional to the features and they the program still worked overall. Often at times this helped find bugs during implementation that where then rectified before the project’s completion.

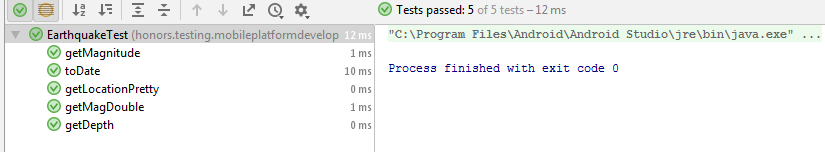
# Test Documentation

## Informal Junit Tests

### Source Code:



### Outcome:



## Formal Testing

### Test Cases

|  |  |
| --- | --- |
| **Test Case Number** | **Testing** |
| 1 | Online/Offline Handling |
| 2 | Basic Navigation |
| 3 | Orientation Handling |
| 4 | Date Refinement |
| 5 | Data Filter |
| 6 | Unfiltering |
| 7 | Other Device Testing |

### Test Outcomes

|  |  |
| --- | --- |
| **1.1** | **PASS** |
| **1.2** | **PASS** |
| **1.3** | **PASS** |
| **2.1** | **PASS** |
| **2.2** | **PASS** |
| **2.3** | **PASS** |
| **2.4** | **PASS** |
| **2.5** | **PASS** |
| **2.6** | **PASS** |
| **2.7** | **PASS** |
| **3.1** | **PASS** |
| **3.2** | **PASS** |
| **3.3** | **PASS** |
| **4.1** | **PASS** |
| **4.2** | **PASS** |
| **4.3** | **PASS** |
| **4.4** | **PASS** |
| **4.5** | **PASS** |
| **5.1** | **PASS** |
| **5.2** | **PASS** |
| **5.3** | **PASS** |
| **5.4** | **PASS** |
| **5.5** | **PASS** |
| **5.6** | **PASS** |
| **5.7** | **PASS** |
| **5.8** | **PASS** |
| **5.9** | **PASS** |
| **6.1** | **PASS** |
| **6.2** | **PASS** |
| **7.1** | **PASS** |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Data | | | | | |
| **Test Case** | **Input Data** | **Expected output** | **Actual Output** | **Pass/Fail** | **Evidence** |
| **1 – Online/Offline Handling** | | | | | |
| 1.1 | Load application on airplane mode | Application displays Toast message to say connection cannot be established | Application displays Toast message to say connection cannot be established | Pass | Test Figure 1 |
| 1.2 | Turn airplane mode off while application is on and had not loaded | Application loads list of earthquakes automatically | Application loads list of earthquakes automatically | Pass | Test Figure 2 |
| 1.3 | Turn airplane mode on while application is on | Earthquakes loaded remain, toast message displays | Earthquakes loaded remain, toast message displays | Pass | Test Figure 3 |
| **2 – Basic Navigation** | | | | | |
| 2.1 | Scroll List | Scrolls without issue | Scrolls without issue | Pass |  |
| 2.2 | List/Map Switch | Switches to map view | Switches to map view | Pass | Test Figure 4 |
| 2.3 | Spinner | Drop Down List drops | Drop Down List drops | Pass | Test Figure 5 |
| 2.4 | Datepicker | Datepicker shows | Datepicker shows | Pass | Test Figure 6 |
| 2.5 | Click earthquake on list | Earthquake View Shows for specific earthquake | Earthquake View Shows for specific earthquake | Pass | Test Figure 7 |
| 2.6 | Click earthquake on map | Marker box opens | Marker box opens | Pass | Test Figure 8 |
| 2.7 | Click marker box | Earthquake View Shows for specific earthquake | Earthquake View Shows for specific earthquake | Pass |  |
| **3 – Orientation Handling** | | | | | |
| 3.1 | Scroll list horizontal | Scroll list displays | Scroll list displays | Pass | Test Figure 9 |
| 3.2 | Map view horizontal | Map view displays | Map view displays | Pass | Test Figure 10 |
| 3.3 | Earthquake view horizonal | Earthquake view displays | Earthquake view displays | Pass | Test Figure 11 |
| **4 – Date Refinement** | | | | | |
| 4.1 | 1/4/19 chosen as date | No earthquakes to display message displayed | No earthquakes to display message displayed | Pass | Test Figure 12 |
| 4.2 | 3/3/19 chosen as date | Earthquake on the 3/3 displays | Earthquake on the 3/3 displays | Pass | Figure 13 |
| 4.3 | Cancel chosen on date screen | Clears date refinement | Clears date refinement | Pass |  |
| 4.4 | 8/3/19 chosen on date screen while “English Earthquakes” filter is chosen | Only shows English earthquakes with chosen date | Only shows English earthquakes with chosen date | Pass | Figure 14 |
| 4.5 | 20/3/20 chosen as date | No earthquakes to display message displayed | No earthquakes to display message displayed | Pass |  |
| **5 – Data Filter** | | | | | |
| 5.1 | Largest Magnitude chosen | Largest Magnitude earthquake displayed | Largest Magnitude earthquake displayed | Pass | Figure 15 |
| 5.2 | Smallest Magnitude chosen | Smallest Magnitude displayed | Smallest Magnitude displayed | Pass | Figure 16 |
| 5.3 | Most northern chosen | Most northern displayed | Most northern displayed | Pass | Figure 17 |
| 5.4 | Most southern chosen | Most southern displayed | Most southern displayed | Pass | Figure 18 |
| 5.5 | Most eastern chosen | Most eastern displayed | Most eastern displayed | Pass | Figure 19 |
| 5.6 | Most western chosen | Most western displayed | Most western displayed | Pass | Figure 20 |
| 5.7 | Scottish Earthquakes chosen | Only Scottish earthquakes displayed | Only Scottish earthquakes displayed | Pass | Figure 21 |
| 5.8 | English Earthquakes Chosen | English Earthquakes displayed | English Earthquakes displayed | Pass | Figure 22 |
| 5.9 | Welsh Earthquakes Chosen | Welsh Earthquakes displayed | Welsh Earthquakes displayed | Pass | Figure 23 |
| **6 - Unfiltering** | | | | | |
| 6.1 | All earthquakes chosen | All earthquakes displayed | All earthquakes displayed | Pass | Figure 24 |
| 6.2 | Scottish earthquakes chosen on map view and view changed to list | Scottish earthquakes are still the only earthquakes displayed | Scottish earthquakes are still the only earthquakes displayed | Pass | Figure 25 |
| **7 - Other Device Testing** | | | | | |
| 7.1 | Test on Google Pixel 2 XL | Works appropriately | Works appropriately | Pass | Figures 26 |

### Test Figures

Figure 1:

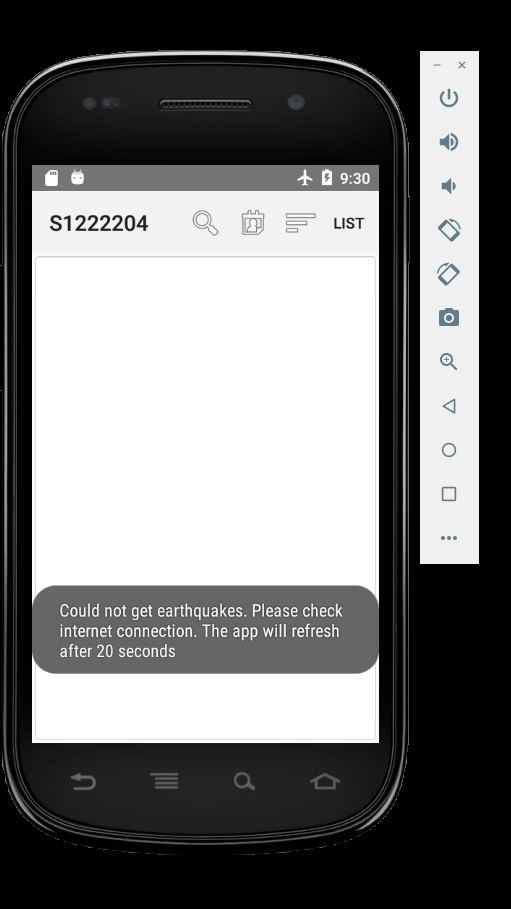


Figure 2:

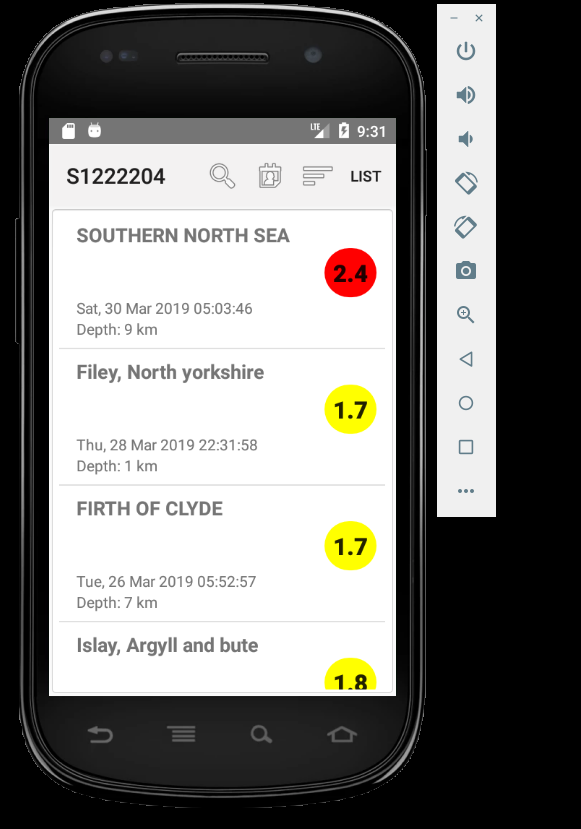


Figure 3:

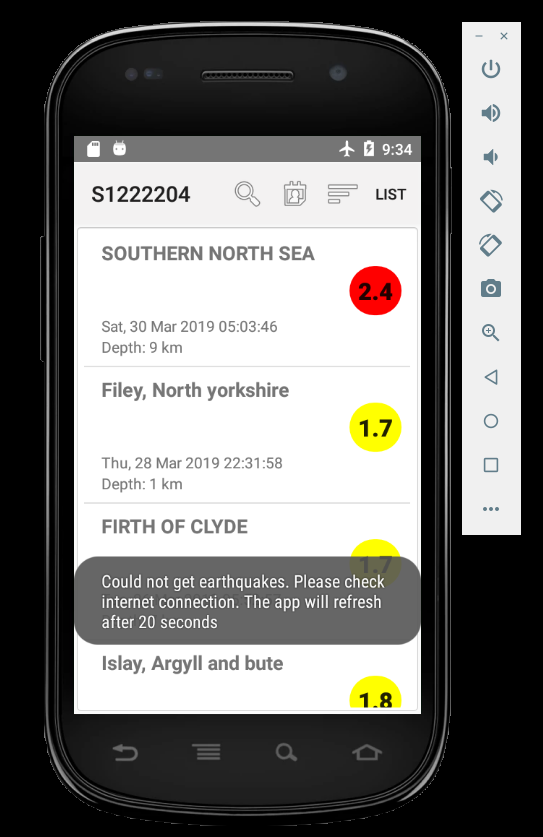


Figure 4

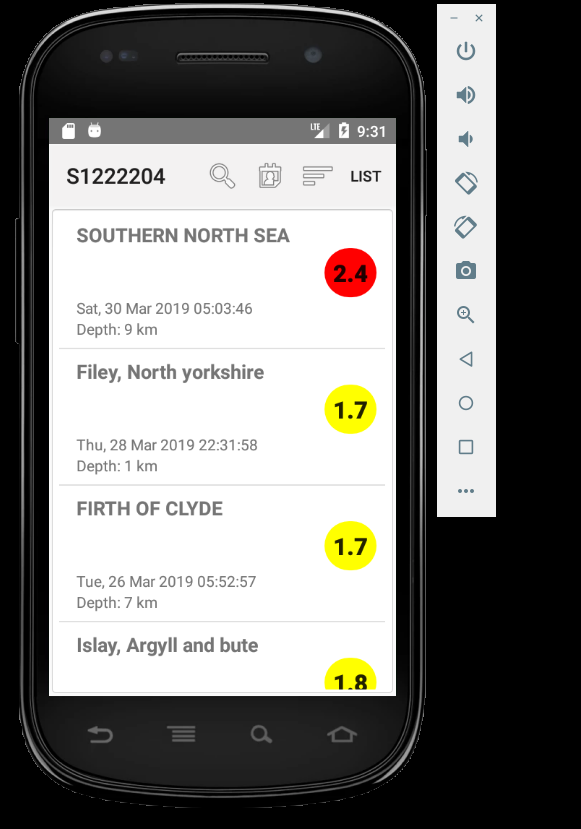


Figure 5



Figure 6

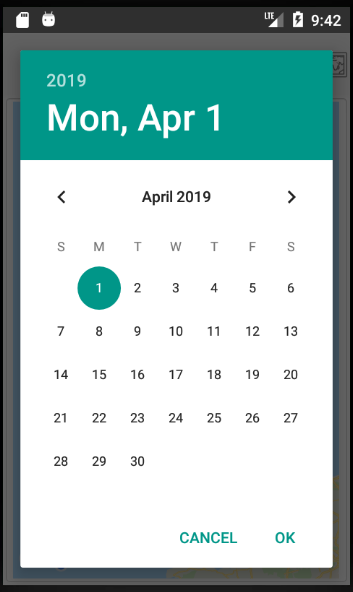


Figure 7

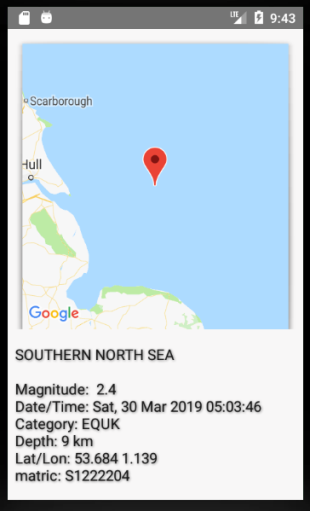


Figure 8



Figure 9

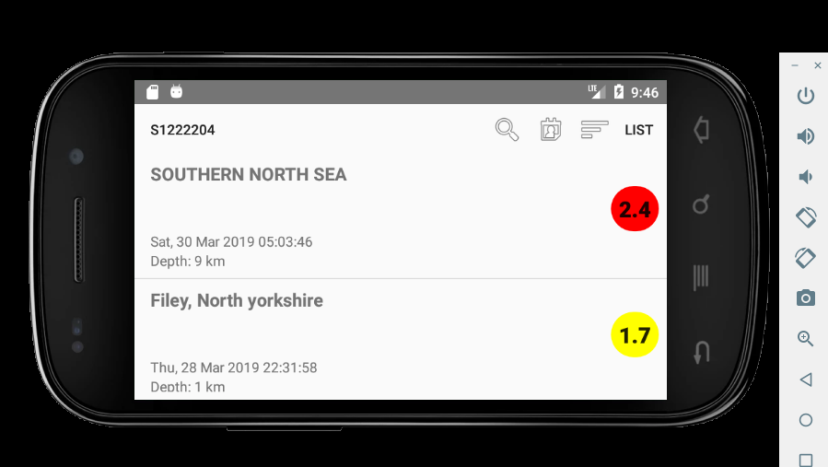


Figure 10



Figure 11

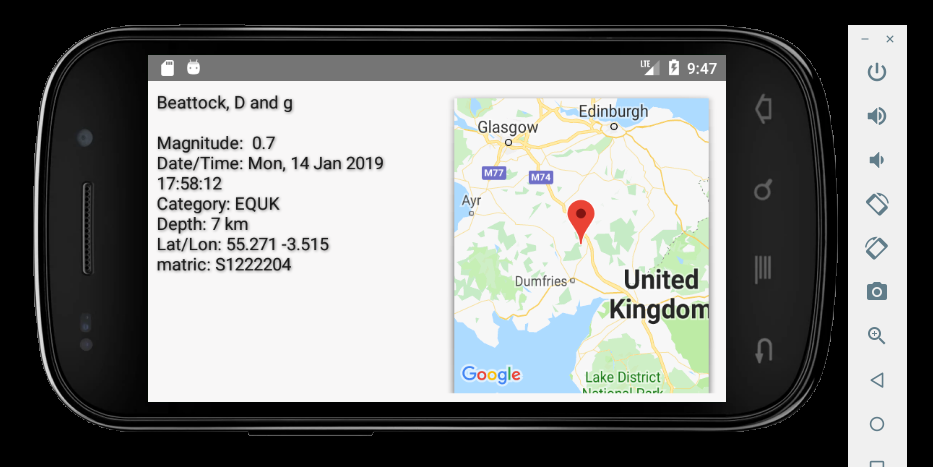


Figure 12:

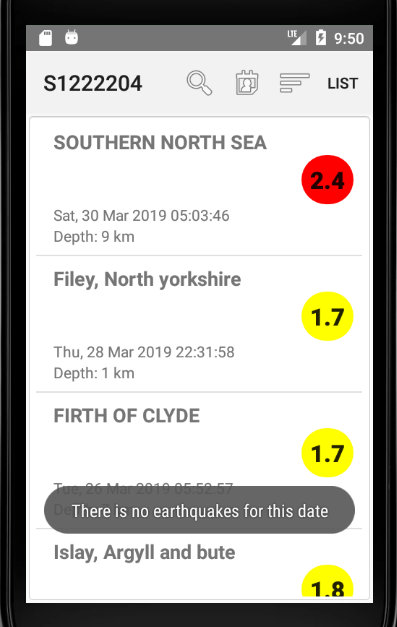


Figure 13

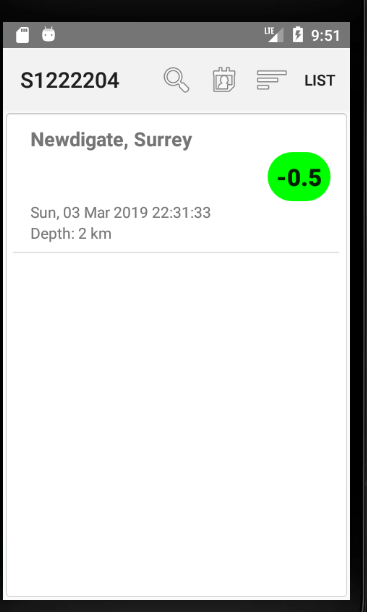


Figure 14

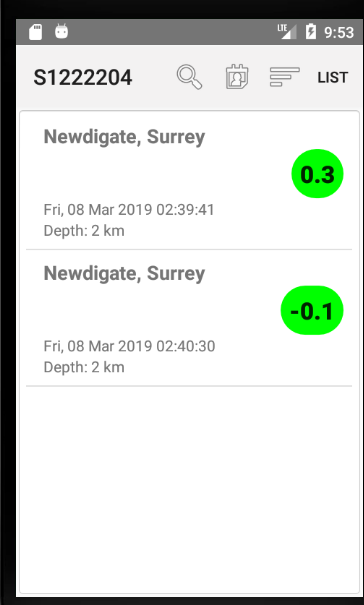


Figure 15

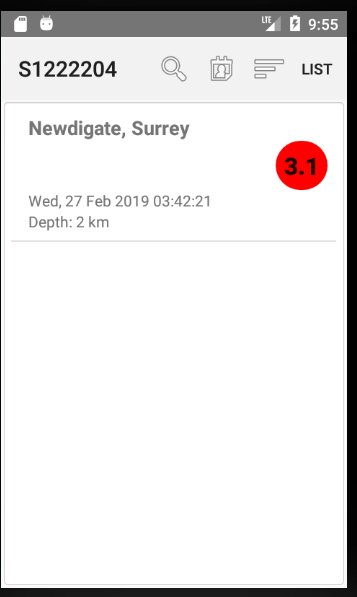


Figure 16

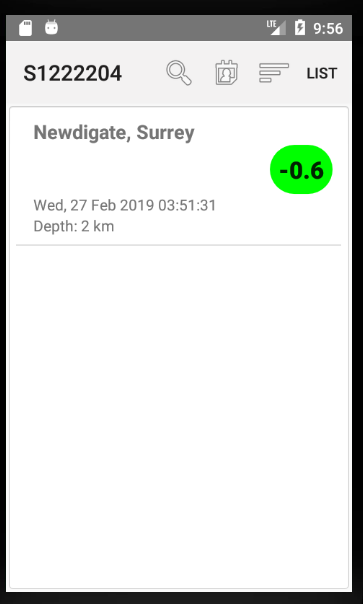


Figure 17

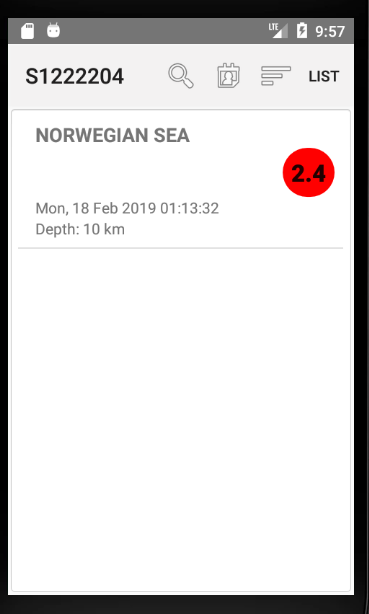
 

Figure 18



Figure 19



Figure 20



Figure 21



Figure 22



Figure 23



Figure 24



Figure 25

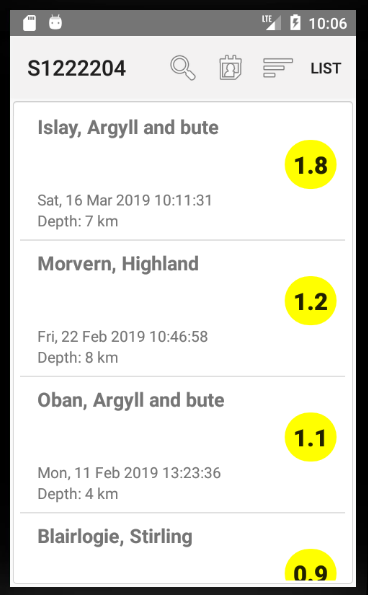
 

Figure 26

