|  |  |
| --- | --- |
| Video Link | https://github.com/podonn203/mpdCoursework/blob/master/MPD\_Video\_podonn203.mp4 |
| .apk Link | https://github.com/podonn203/mpdCoursework/blob/master/app-debug.apk |
| Android Studio Project Link | https://github.com/podonn203/mpdCoursework/tree/master/podonn203 |

Mobile Platform Development Report

Patrick O’Donnell: S1714595

Contents

[Design Report: 1](#_Toc5903968)

[Start Page: 1](#_Toc5903969)

[Display Info Page: 2](#_Toc5903970)

[Calendar Page: 2](#_Toc5903971)

[Future Recommendations: 3](#_Toc5903972)

[Conclusion: 3](#_Toc5903973)

[Testing Report: 3](#_Toc5903974)

[Black Box Testing: 4](#_Toc5903975)

[White Box Testing: 4](#_Toc5903976)

[Conclusion 4](#_Toc5903977)

[Test Logs: 5](#_Toc5903978)

[Screenshots: 5](#_Toc5903979)

# Design Report:

The outcome for this project was to create a mobile application to display the most recent instances of earthquakes with in the United Kingdome. The application was program in the Java programming language using Android Studio. The application parses an Extensible Mark-up Language (XML) feed that will provide information about earthquakes that happened within the past 100 days from bgs.ac.uk. In this design report I will discuss the Earthquake application, the pages that where designed, how they were designed and how the user of the app will be able to understand the mobile application.

## Start Page:

The start page of the application is the main page of the application that user will view and interact with. When the page is loaded it will display a list of all the recent instances of the Earthquakes, up to 100 of them. For each of these earthquakes instances, they were parsed from the XML feed in to the application. For each of these instances displays there is a horizontal line to separate them. The data that is displayed on the start page from the XML feed is the location of where the earthquake occurred, the date and time for when the earthquake happened and the magnitude level of that earthquake. For the magnitude level displayed there is a colour indicator that displays to show the level in a colour format. For the colour format is goes as follows;

* Green – For earthquakes with a negative magnitude
* Blue – For earthquakes with magnitude levels from 0 to 4
* Orange – For earthquake with magnitude levels between 4 and 5
* Red – For earthquake with magnitude levels of 5 and above

There is also a different gradient of colours of the blue and red levels. For levels from 0 and 2 it is a darker blue. For levels between 2 and 3 it is a lighter blue and levels between 3 and 4 a very light blue. For each of the levels above 5 to 10 the shades of red change for each level and for any magnitude level above 10 will be the darkest shade of red. These colours are displayed in a drawable of a circle shape.

The start page is designed to be as user friendly as possible and simple to understand. For the view of the application, it can be displayed portrait and landscape. For when the phone is set to landscape the view of the application adjusts to suit the new view. The display of the content for displaying landscape is the same as it is from portrait and the information from the XLM feeds is still laid out in a presentable format. The start page also includes a scroll view, as this view will allow users to easily scroll view all of the earthquakes stored and displayed from the xml document, as without the scroll view users would only be able to view the first five instances of earthquakes when portrait and the first three when landscape.

The colour scheme for the application was kept simple with a white background. I experimented with different background colours but most of them would either strain the eyes or would not look nice displayed. The text was kept black to keep it simple and to compliment the white background. The header colour was made green, just to give the application an “earthy” feel so suit the application. For the font sizes the location was set to 20 font size, as this acts as the title for the instance to make it stand out to the user. The font size for the date and time was set to 15, just to act as a sub-heading. For the magnitude levels, the font size was set to 11 so it would fit nicely inside the circle, while also being readable for user. The font colour for the magnitude was also set to white to compliment the background colour of the circle.

## Display Info Page:

The info page of the application is where the further information of a single earthquake instance is displayed. It is displayed on a second page. This second page will appear when a user clicks one of the single earthquake instances on the main page. When this page is loaded, it takes data from the XML feed, which is taken from the Pull Parser class and displayed the information to the user. The layout for the page is in the form of a List View to display the information and description of that earthquake from the XML feed. This is achieved from using the Earth Adapter class. For the information that is displayed on these single pages it will first show the Location, the Date and time of when that earthquake instance is publish and the category. The reason for displaying the location, date and time again is so that the user knows the have selected the right earthquake page, as it will match the one they have selected on the main page. In addition more information on the earthquake itself is shown. This includes the magnitude level, which will match the magnitude level on the start page, the depth of that earthquake shown as km (kilometres), the latitude and the longitude of that earthquake to. Similar to the start page, this page can also be viewed in both portrait and landscape. In addition the background and header will also share the colours of the start page. The text size for the data displayed is kept as 20 for each piece of data.

## Calendar Page:

Due to time constraints the calendar page was not implemented however designs where thought up and taken into considerations thought out the project. A wireframe was created to display what the calendar page would look like (See figure 1). This page would have been displayed through a button on the start page. The page itself have display a basic calendar picker, ideally starting with the current date. User could browse the calendar and select a date, this would then bring up a list view of all the earthquake instances that occurred that day. User could then select one of the instances similar to the start page and it would bring them to the Info page as discussed earlier to display that information for that earthquake. It again would function to be displayed in both landscape and portrait.

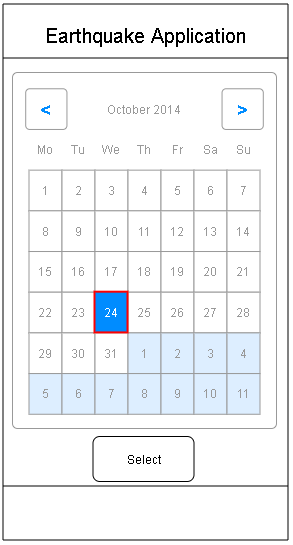


Figure 1: Wireframe for Calendar Page

## Future Recommendations:

For future recommendations for this earthquake application I would recommend implementing the colander date picker page and date selection page as mentioned above, allowing the users to managed the earthquake instances by date as of right now only being displayed as a list view from most recent to least recent. This would help improve the user experience for users. I would also recommend a map functionality that would display where the earthquake happened on the map. I would have also liked to include functionality to help the user find the largest magnitude earthquake and the deepest earthquake from the list of instances from the XLM feed. Also displaying the most northern, southerly, westerly and easterly earthquake. This would be achieved through a simple button for each one.

For what the application has in terms of functionality I would have liked to make the data displayed in a more presentable format. This would include separating the array of both the Date and Time and the Latitude and Longitude, as of right now the application displays them together rather than separately. This help user the user to identify these values easier.

## Conclusion:

The design report for this project was carried out using the brief from the Application Speciation. The goal of the project was to parse and display data from a XML feed in a presentable format for users to view. This was achieved through researching into Java language techniques found online and through the lab exercises handed out and was implemented into the final application. For the data displayed considerations where taken on how to display it. For displaying the magnitude level colours, I took inspirations for risk tables for implemented them. For additional work that could be added, I would refer to the future recommendation section and implement that function that where listed there.

# Testing Report:

In the testing report I will go into detail about the testing methods that I have used to test the Earthquake Application. These testing methods include white box testing, black box testing and created and filling out test logs to show the functionality of the project. Testing the earthquake applications functionalities and code was complete to make sure that the application is robust and that everything is operating as expected before finishing the project. If testing is not carried out, then it is sure that many bugs and unexpected errors will occur. The application was testing using the Android Studio’s emulator – Nexus S API 23 (Marshmallow). When running the application through the emulator sometimes it would crash and not open the application, this was due to either an error with in the XML layout files or with in the Java class files.

## Black Box Testing:

The black box testing method is performed by the tester having no prior knowledge of the internal code of the system of how it was developed. These tests focus on the application itself and the requirements and functionality of the system. This will allow me to check that the application is running as intended and check if any errors occur during the testing process.

The advantages of using black box testing are; that these tests are performed from the user’s point of view and will help indicate the differences that are listed in the specifications. Another advantage is that the user can perform these tests without needing to know any programming language or have previous experience using programming tools. Test cases can also be designed and created as the specifications are complete.

However black box testing does come with its own disadvantages. These include that there is a limited number of inputs that can be tested in the system and that other programming paths will not be tested. Another disadvantage is that some tests can be redundant if a test case has already been performed.

## White Box Testing:

The white box testing method is when testing is carried out where the tester has the knowledge of the internal structure of the application, program code and design. This should be used to help predict the behaviour of the application when options are selected. White box testing includes the testing methods of Unit testing, where different sections of the application are completed in an iterative lifecycle, meaning that one section must be complete before the next section can start. Acceptance testing, this is used to guarantee the application worked as a whole and that all of the elements are working.

The advantages of using white box testing are that testing can be completed at an early stage of development. There is no need to wait for the user interface to be created and made ready. Another advantage is that the tests are tested at the deep end and will cover a majority of the paths.

The disadvantages of using white box testing are that the testing process can be complex and so experienced testers are needed to perform the task and needing the knowledge of the applications and how it was implemented.

## Conclusion

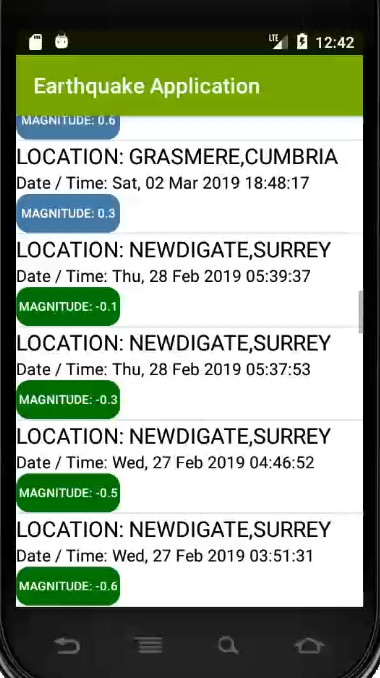
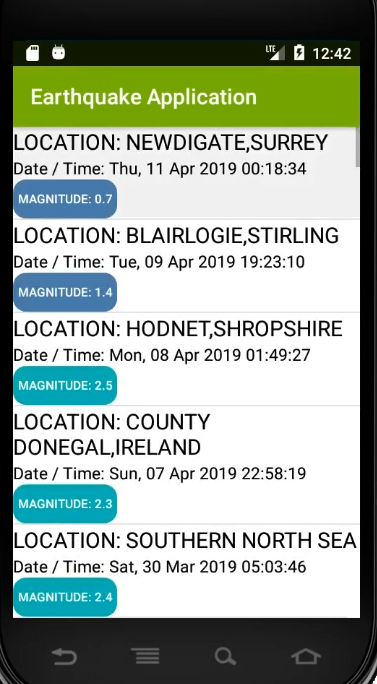
For testing the earthquake application, I decided to mainly focus on using the black box testing method to complete my tests. I found it was best suited for this application because it will help display that all the functions are working as intended for the user. However white box testing will also be implemented a little during the development of the application in the form of test logs. These test logs help identify where the application might be crashing at or help display that the data is being parsed correctly or at all from the XML feed to the application.

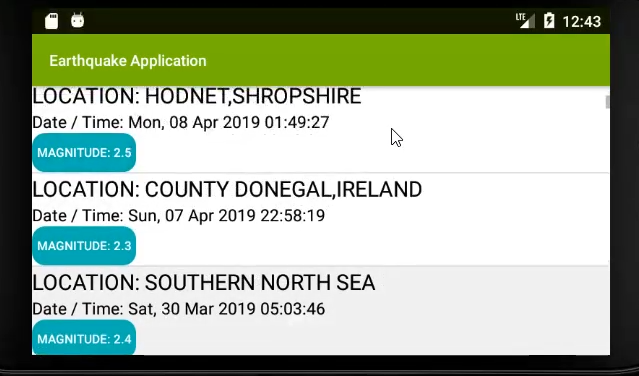
## Test Logs:

|  |  |  |  |
| --- | --- | --- | --- |
| Test Number | Test Case | Test Data | Output |
| 1 | The user runs the applications. | The run “app” option | The application appears on the emulator. |
| 2 | After the application has open the XML data should display | The application has loaded | The page is populated with data from the XML. |
| 3 | The user scrolls down on the start page. | Scroll down the page. | The start page scrolls revealing more list views. |
| 4 | User clicks one of the feeds to show more info on that earthquake instance | onClick on a single earthquake instance | The Info page is displayed. |
| 5 | Displaying the additional information on the Info page. | onClick on a single earthquake instance | A list view is displayed contain data from the XML feed about one earthquake instance. |
| 6 | Return button from Info Page to Start Page | onClick | The start page is displayed. |
| 7 | Magnitude Colours | Magnitude level is from 0 to 2 | The drawable shape is coloured dark blue |
| 8 | Magnitude Colours | Magnitude level is from 2 to 3 | The drawable shape is coloured lighter blue |
| 9 | Magnitude Colours | Magnitude level is from 3 to 4 | The drawable shape is coloured a light blue |
| 10 | Magnitude Colours | Magnitude level is negative | The drawable shape is coloured green |
| 11 | Displaying the Start page as landscape | Emulator is rotated to landscape | Page is displayed as landscape |
| 12 | Displaying the Info page as landscape | Emulator is rotated to landscape | Page is displayed as landscape |

## Screenshots:

Start Page:





Info Page:

