Mobile Application Development Coursework Report

Lea Whitelaw
40279124@live.napier.ac.uk
Edinburgh Napier University - Module Title (SET08114)

Intro

Keywords – mobile, app, android, java, androidstudio, studio, development, software

1 Introduction

Movie_Guide is the app I have created for this coursework for the Mobile Applications Development module.
The brief for this coursework was to design and implement an android app prototype using the Android Studio SDK platform. There were no specifications for the type of app or any necessary functionality that was required for the app. This left the brief quite vague. To aim to get the most out of this coursework I knew that I had to cover functions in my code that weren't taught to us in the workbook, as it would show my ability to learn and develop code without clear instructions. Something that was not taught to us was how to use an API to pull information from the internet to use in our app. This was something I was sure that I wanted to learn for my app. The inspiration to do a movie guide came from the fact that there are a lack of apps which show local cinema times in all cinemas in your area, most apps available just show times available for one company, i.e.- cineworld, odeon or vue. My original idea was to build an app which showed relevant movies with their plots, ratings and times in your local cinemas.

2 Software Design

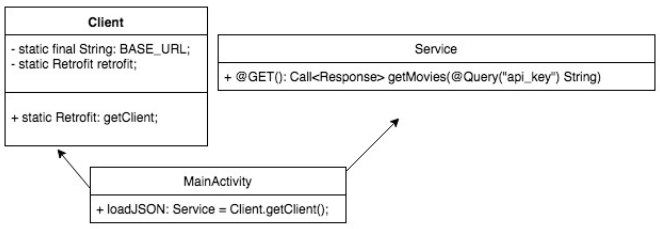

Figure 1: **Retrofit Class Diagram** - My software design of the retrofit method for getting API data

To begin my software design and developing my app, I needed to do some research on how I was going to get the API to feed data to the app. A method that I came across was the 'retrofit' method, which uses client and service classes to create methods which pull the API data from the HTML header with your personalized API key, which come together in a function which loads the JSON file onto your app. This website was very helpful in learning this process; http://mobilesiri.com/retrofit-tutorial-android-studio/

Shown in Figure 1 is the class diagram I created to show how the software would come together in terms of pulling the API. I learned that I had to register on the website which I wanted an API from to get my own personalised API key which I am not allowed to post anywhere. Some of the API keys were not free but I managed to find one on theMovieDB.org.

My design idea's for my app were pretty simple, I wanted it to have a main home page which scrolled through a lot of different movies and once you clicked on them they opened their own page with more details about the movies and local cinema times. I struggled more to find info about how to pull an API about cinema times and how I would display them in correlation to their individual film titles. I decided to move forward with building the basic app that just showed the films first and come back to the cinema times later.

The colour theme for my app was based on different shades of red. I used color-hex.com to find an appropriate colour scheme. The colour red was chosen because it is a colour that is often associated with cinemas because of the old style red velet theatre curtains and the red and white popcorn boxes. I thought the colour scheme would be fitting with the purpose of the app and more aesthetically pleasing than the basic android themes provided in android studio.

3 Implementation

On top of involving APIs in my app I wanted to use some other techniques that hadn't been taught to us in the lectures or workbook. I came across the option to use RecyclerView as a more advanced version of the ListView that we had been taught in the workbooks. Although quite complex to understand, after watching enough tutorials on YouTube and looking through tutorial websites online, I managed to implement the recylerView and implement my home screen as a grid view instead of a list view, as shown in Figure 2.

Another element I implemented was the use of the Glide function in android studio as a way to display my thumbnails and placeholder images for the loading movie pictures. The use of glide was simple to learn through this tutorial online; https://inthecheesefactory.com/blog/get-to-know-glide-recommended-by-google/en it was an efficient way to display all my images.

Overall the implementation process was more challenging than I thought it would be, and due to this it meant I had

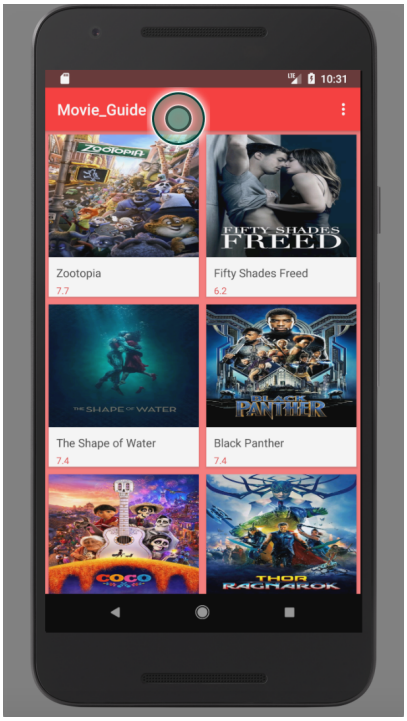

Figure 2: **Grid RecyclerView** - The layout of the homepage of my app

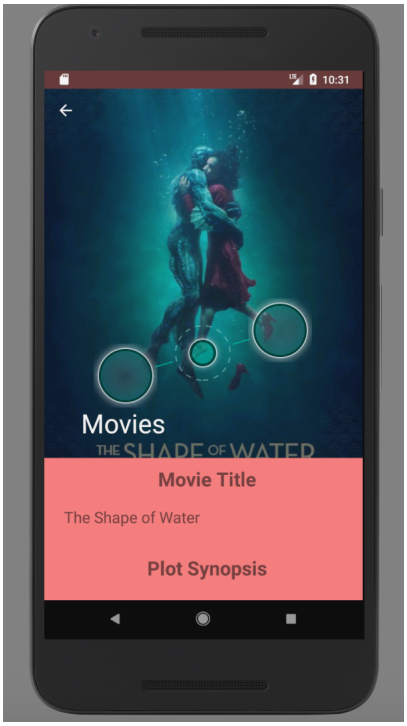

Figure 3: **Clicked on** - Page view once a film is clicked on by user.

not left myself enough time to be able to add all the features that I wanted into my app. Getting to grips with getting the API to work was difficult, I had to use lots of Log functions to trace my errors which were sometimes HTML errors and sometimes errors in my code. I had to look through many online tutorials to get everything to work in the end. Once I had finally implemented my basic working app which showed the films, their user ratings, release dates and synopsis(Figures 3,4 5), I was aware that I would not have enough time to finish all my ideas for the app, so instead I added a settings menu which allowed the user to change the layout from popular current films to top rated of all times in theMovieDB.org's database. I used the android developer website to learn how to add the settings and preference fragments classes to add the settings feature(Figure 6).

4 Critical Evaluation

My biggest point of criticism for the outcome of this project is that it doesn't do one of the main features that I originally planned it would. There are multiple movie guide apps on the play store which I believe would compete with this one, the edge would have been if I had added the cinema times, a task which I couldn't finish before the submission deadline, but I may carry on to work on in my personal time for my knowledge development of this module. After showing some course-mates my app, the feedback I received was that it had a good colour scheme, it was well laid out, the function to change it from currently popular to top rated was good, and the fact that it refreshed with the APIs online data was good. The negative points people gave were that it was slightly basic functionality wise and that there are probably a lot of similar competitive apps already out there.

If I were to do this project again, I would definitely have given myself more time so that I could work on the features that I have mentioned which I couldn't complete for this project.

5 Personal Evaluation & Conclusion

Overall, I am both pleased and disappointed with this app. Whilst I am proud of myself for managing to get the app to perform the functionality it currently does, I am disappointed that I couldn't get it to do more in time for the deadline.

The learning curve when getting started with Android Studio was tougher than I thought and I am displeased with myself in the way I underestimated this coursework. Learning from this, I will go on to try and finish to project to complete my learning process and in future I will definitely give myself more time for big projects like this.

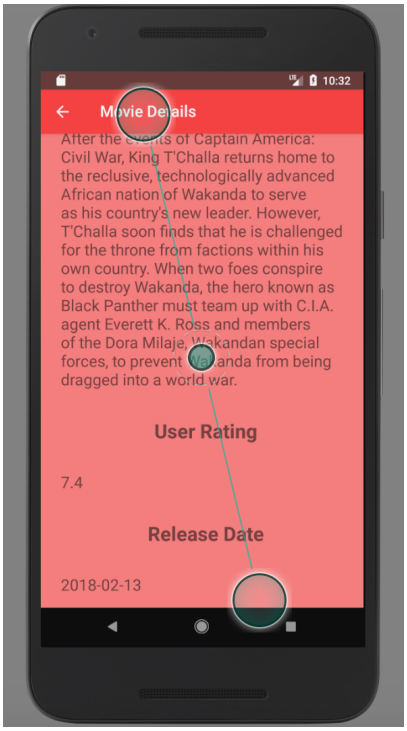

Figure 4: **Synopsis** - Text with plot of film

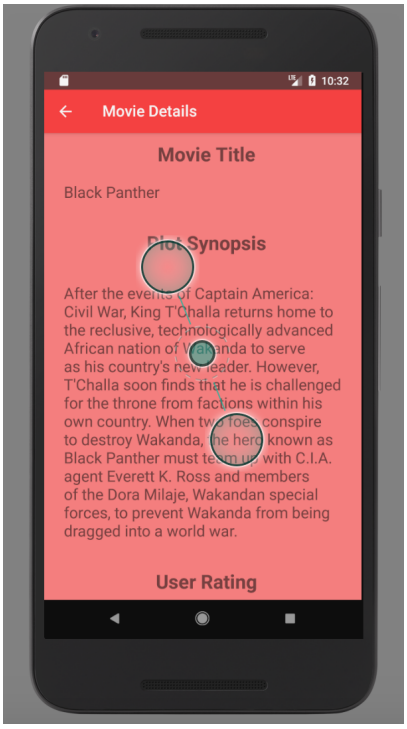

Figure 5: **User Ratings** **Release dates** - Further information on the films

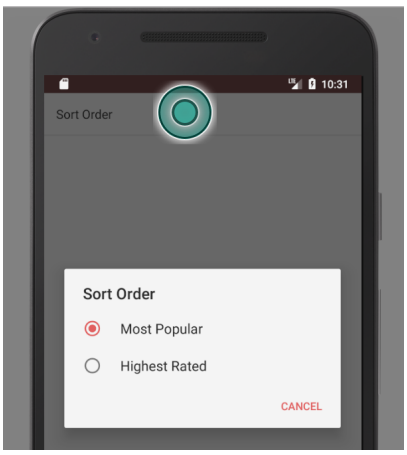

Figure 6: **Preference Order** - The layout of preference order settings menu.