

# Coursework Report

## Your Organizer

Marcos Ramirez Aceves  
40284094@live.napier.ac.uk

Edinburgh Napier University - Mobile Applications Development (SET08114)

## Abstract

The intention and goal of this project was to create a functional android mobile application using Android Studio. In the design phase of the project there were two main sources of inspiration. The first one was related to the screen design of the project, taking references from the social media networks Twitter and Facebook. The second one was more related with the actual functionality of the application, taking from reference a basic note taker application. This report will cover the development of this application including design, testing and evaluation.

**Keywords** – App, Organizer, Films, Television, Books, Comics

## 1 Introduction

### 1.1 Scope and content

The main target of the project was to create a mobile application to show the control of the several tools provided by Android Studio as well as the understanding of the different parts of an application, like activities and their life cycle or intents. This application is provided with functionality that allows the user to store three types of lists in internal storage with additional information, those lists being a list of films, a list of television series and a list of books. Regarding to the additional information, the user is only obliged to enter the name of the element(film, series or book) to store, in order to keep track of such element. The reason of making this type of implementation was to allow the user to have more flexibility when using the application and to provide different levels of detail.

### 1.2 Application choice

The main reasons behind the decision of making this type of application were the following:

1. A personal need of having something to keep track of books that I would like to read and television series or films that I would like to watch in an easy and organized way.
2. The thought of that more people may want to use this type of application.
3. The lack of an application that provides this type of functionality.

## 2 Software Design

The project started with a simple design using post-its representing the application layouts. After several iterations and considerations about the visual structure and best location of the different widgets to be used, the design was completed, however the initial design needed to be adapted to the tools provided by android studio.



Figure 1: Post-Its Design

This was followed by the design of a Classes diagram using the Unified Modelling Language (UML), representing the different Java Classes used in the application(the diagram does not include the Java Classes used for the different activities of the application). These diagrams were drawn using the online tool: Draw IO (<https://www.draw.io/>).

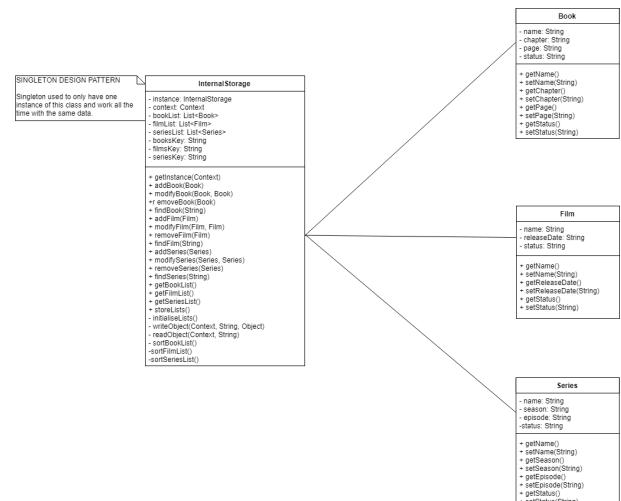


Figure 2: Classes Diagram

# 3 Implementation

## 3.1 Fragments, activities and buttons

The application mainly consist in three fragments, each fragment being responsible of one of the three different lists already mentioned. The visual structure of the three fragments is mostly the same, with the only variation of the attributes shown in each of the table layouts corresponding to the attributes of the objects the list is storing. The three tables are inside of scrollable widgets to make sure that the user can see all the elements of the tables even if there are lots of them or if there is an element with a large name.

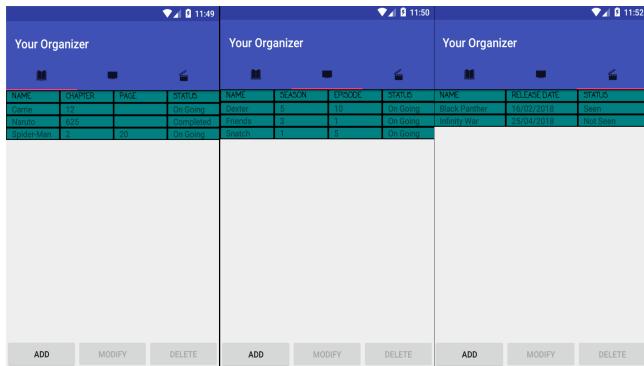


Figure 3: Fragments 1

As can be seen in Figure 3, two of the three buttons are disabled by default. The modify and delete button will be enabled after the user has selected a row(an element) of the table. Also the user is only allowed to select one at a time, having to touch again the selected row to deselect it.

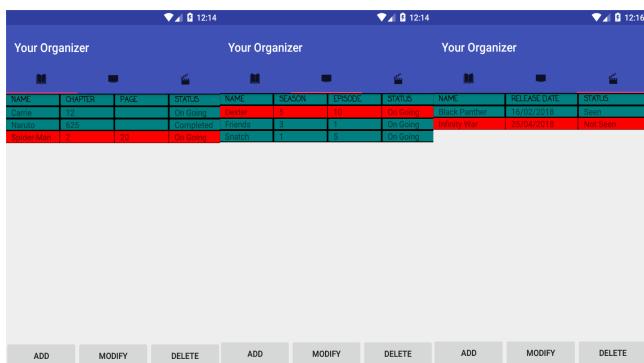


Figure 4: Fragments 2

The delete button will remove the selected row from the table and the corresponding element in the list. The add button will take the user to a new activity, in which the user can enter the details of the element to be added in several fields. The modify button will take the user to the same activity as the add button but in this case the fields will be automatically generated with the attributes of the selected row(element) of the corresponding fragment. Each fragment has his own Add/Modify activity, however in the same way as with the fragments, the visual structure of the three activities is mostly the same, being only difference the fields to be entered, which depends on the attribute of the element to store/modify.

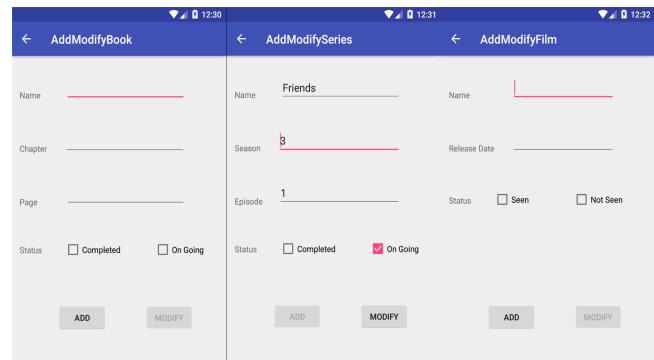


Figure 5: Add/Modify

As can be seen in figure 5, the behaviour of the Add/Modify activity depends if the user wants to add or modify an element.

## 3.2 Storage

Deciding the data model for the storing strategy is a really important part during the development, one of the reasons is that the way in which data is stored can affect the application performance and sometimes storing can be a big problem if it is not implemented correctly. When coming to think about the data model there are also some other considerations to have in mind like the security or the actual hardware of the mobile device. For this project, the security is not really a big deal as the application is not storing any type of sensitive data like personal information or location data. At the early stages of the project one of the decided features to be implemented was persistence of the three lists, and therefore different data models needed to be considered in order to select the most appropriate one. After thinking about the type and the amount of data to be stored, the time and the complexity, internal storage was the model decided to be used in this project.

## 3.3 Testing

Testing the application has been carried out during the complete development process, each time a new feature was added it was tested in an android virtual device to make sure that it worked as expected and if that was not the case, then the code was debugged until the error was found. In this project unit testing was not implemented because the classes used for the application only had getter and setter methods. However, an android test of the InternalStorage class was performed. Further testing should include UI testing.



Figure 6: Testing

## 4 Critical Evaluation

### 4.1 Original concept

The original visual structure of the application is detailed in Figure 1 while the final result is shown in the Implementation section, therefore it can be seen that the final application concept is mostly the same as the original with some variations, at least when it comes to the visual structure. About the functionality, the final result is a bit different than the original one, the main reason is that some of the features have not been implemented because of a matter of time, this is discuss in the last part of the section. Overall the final result is highly accurate to the original one.

### 4.2 Comparison and evaluation

By performing a research in the Google play store several applications with similar functionality were found. After analyzing those applications, what can be seen is that the main differences are that most of those applications focus on a single type of element (like keeping track of books or films, but not both at the same time), and the visual structure, while few of the applications found had a simple visual design, most of them included pictures of the films or the books, as well as short descriptions or even rating. When it comes to the functionality, it is similar to this project, but they have included further functionality and details like searching, duration or rating. The application developed in this project provides a simpler visual structure as it tries to be as clear as possible, and facilitates the search by keeping everything in alphabetical order, however it is true that if the data becomes very large a search function would be very useful. When it comes to the images, in this case I think that they would not make things clearer(as intended in the other applications) as they will just inflate the user interface. Apart from that, further details could also be included in the elements stored as shown in those applications.

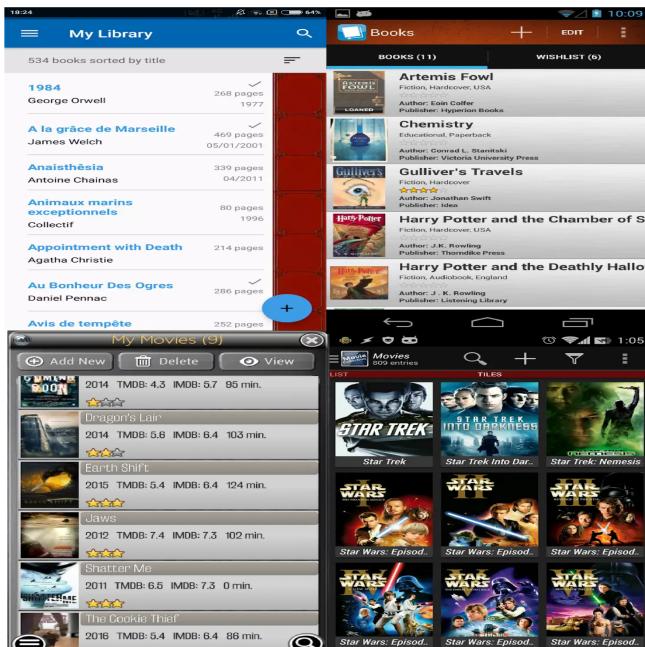


Figure 7: Other Applications

### 4.3 Improvements

Providing further functionality is not always the best way of improving an application however, this project has a solid foundation and for that reason the majority of the improvements consist in adding further functionality to the application. These are some of the improvements that can be provided to this project, starting from what was discussed in the last section, a search function could be included to allow the user to find the elements stored faster. Further functionality that was part of the original idea for the application can also be implemented, like allow the users to increase the chapters/season/episode/page directly instead of manually changing the value of those properties, allow the user to make visibility changes in the properties of the elements stored so they can decide what properties do they want to appear in the table, allow the user to modify the colours for the interface and finally, provide the application with some functionality so it can send a notification to the users about a new episode of a series, or a new film or a new book being released that day.

## 5 Personal Evaluation

### 5.1 Advance features

Working in this project has allowed me to obtain further understanding on the different parts of an application and the way in which they work. The first element added to the application that was not covered in class were the fragments, fragments can be thought as sub-activities and the idea of using them came to me while I was using Twitter, I thought that the navigation shown in Twitter in which you can navigate by switching between different fragments was something that could be very useful to provide an easy and fast navigation for my application. However, at the beginning I had difficulties in understanding and implementing them, for that reason I searched in the android developer website all the information relevant to fragments and also I searched for some tutorials in order to implement them in the best way possible. The second advance feature implemented was making the rows of the tables selectable and highlighting them, I had no problems implementing this part. And the third element added to the application not covered in class were the icons for the application and the tab items using the Image Asset tool provided by Android Studio. Although this was relatively easy to implement, I had some trouble by making the tab icon items to behave in the way I wanted, however with the aid of the information provided by the android developer website I managed to solve it.



Figure 8: Icons

## 5.2 Challenges

During the development of the evaluation I faced different challenges. The first one was designing the layouts, as my application uses constraint layouts for its views I needed to make sure that they looked correctly in different devices with different screen sizes. Android Studio provides in the palette an option to change the device that is being used for the design, this allowed me to change between different screen sizes and make sure that the widgets were in the correct position all the time. Another problem I faced was the addition and modification of table elements, as the application allows the user to just fill the name and leaving everything else in blank I had to be careful with the actual values I stored when adding a new element to the list or auto filling the fields when the user wanted to modify an element, that is the main reason why I am using strings for the different properties. The third challenge I faced was about making the rows selectable and pass the selected element to the modify activity or delete it, I needed to have several things in mind like if more than one row could be selected or what would happen if a user tries to delete or modify multiple rows. After considering those elements, I decided that the best course of action was making sure that the user could only select one row at a time, and in the case that the user wants to select more, a toast shows a message warning about that action not being possible. The last challenge I faced during the development of this project was the tables design. The tables are implemented programmatically and I had several problems making them to behave in the way I wanted. Starting from the text views inside the table rows, I had trouble with making borders for the cells and making sure that their size were appropriate, for that reason I decided to implement a gradient drawable for them, however I still had several issues. Other of the problems I had was making sure that at the beginning, the columns filled the screen width completely. And the final and main issue I had with the tables design was that when adding elements or removing them dynamically strange spaces were generated between the cells. As I had all this problems with the tables I decided to debug the code carefully using logs until I found the reason behind that behaviour and I finally managed to sort it out.

## 5.3 Performance

Overall my performance working in this project was pretty good, I spent a long time working on it and put lots of effort, however I got stuck for a long time with several of the problems already mentioned and that made me lose a lot of time causing that some of the additional features I wanted to implement were not added to the final result, and I think that the main reason behind those problems was my lack of experience developing a mobile application, after all this was my first time working on one. After saying so, I still feel that my application is what I wanted to create and I am happy with it, but as always there are still things that can be improved.

## 6 Conclusion

## References

1. Icons shown in Figure 8.
2. Android Developer Website [developer.android.com](http://developer.android.com)
3. Genymotion <https://www.genymotion.com/>
4. Fragments Tutorial [https://youtu.be/bNpWGI\\_hGGg](https://youtu.be/bNpWGI_hGGg)
5. Constraint Tutorial <https://youtu.be/z53Ed0ddxgM>
6. Stack Overflow <https://stackoverflow.com>