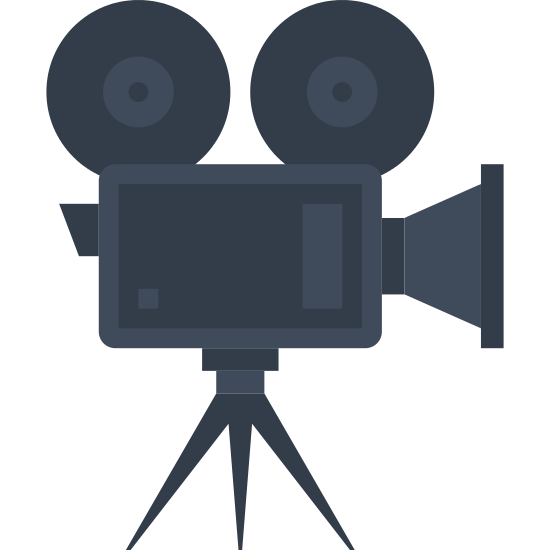
Project Report on Movie Searching Android App



Members:

|  |  |
| --- | --- |
| ***Bidisha Das*** | ***25*** |
| ***Swagata P Das*** | ***26*** |
| ***Swarnabha Das*** | ***27*** |
| ***Vishal Das*** | ***28*** |

**About the Project**

The project aim is to make an application which would search for related information of Movies that are given as an input by the user. On providing the input in the search bar, the related movie list, the respective posters and the plot of the movie will be displayed adjacent to each other in a list by the Application.

On further click, the Application would display the plot in the detailed description, the ratings and the cast, too.  
This would provide us knowledge about the movie and its ratings, what the movie is all about.

This android application is to be realized on Android Software Development Kit using OMDb API to obtain the necessary movie information, all content and images. Since access to OMDb API is limited per user, an API key is obtained to keep allowing access to the OMDb API.

**Technologies Involved**

Technologies that are involved are as follows:

1. Android SDK(Android 9)
2. OMDB API
3. Programming Language: JAVA
4. Volley for Networking or sending JSON Object Request to API

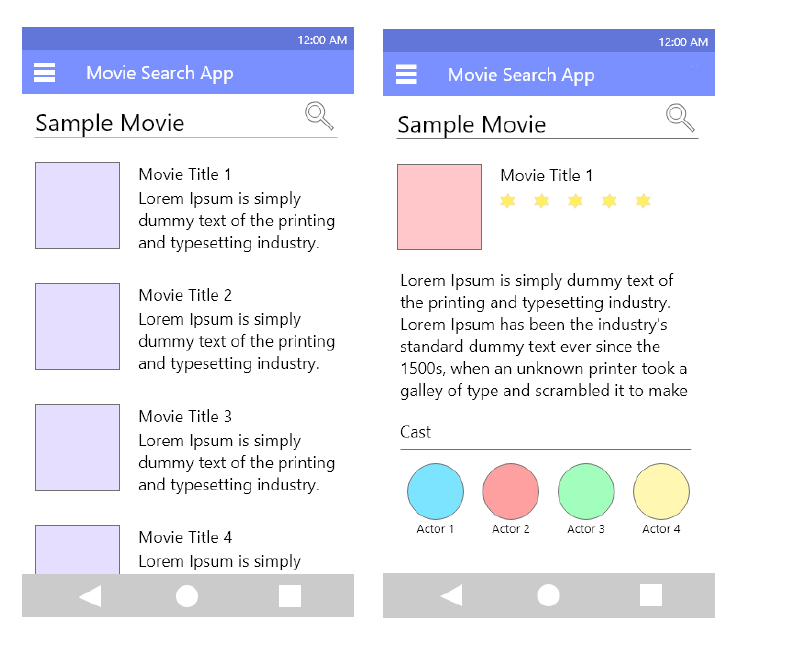
Android Studio and SDK tools has been used to develop the concerned application. OMDb or Open Movie Database is an openly maintained Database of movies and related content contributed by the users. The OMDb API allows accessing data from this database for use in the project.

The programming language in use is JAVA. Volley is an HTTP library that helps Android applications to network easily and quickly, in short, a networking library for Android. JSON Object Request, on specifying a URL, API in this case, fetches the movie information as a JSON object in response.

**Methodologies Involved**

#### Inception - An application for the User to provide a movie title and obtain related information instantly.

#### Design - Since User experience depends on the user interface, it needs to be simple yet accomodate all requirements. The app needs to be linked with an API to the OMDb to acquire the list of movies and their details. The user can scroll down to view the entire list of movies.



#### Development - The development of this project involves- setting up the user interface, fetching data using JSON objects from a movie database using API and displaying the required information on the app for the end user.

#### Stabilization - The API fetches data only a limited number of times per user. An API key has to be acquired to keep gaining access to the OMDb.

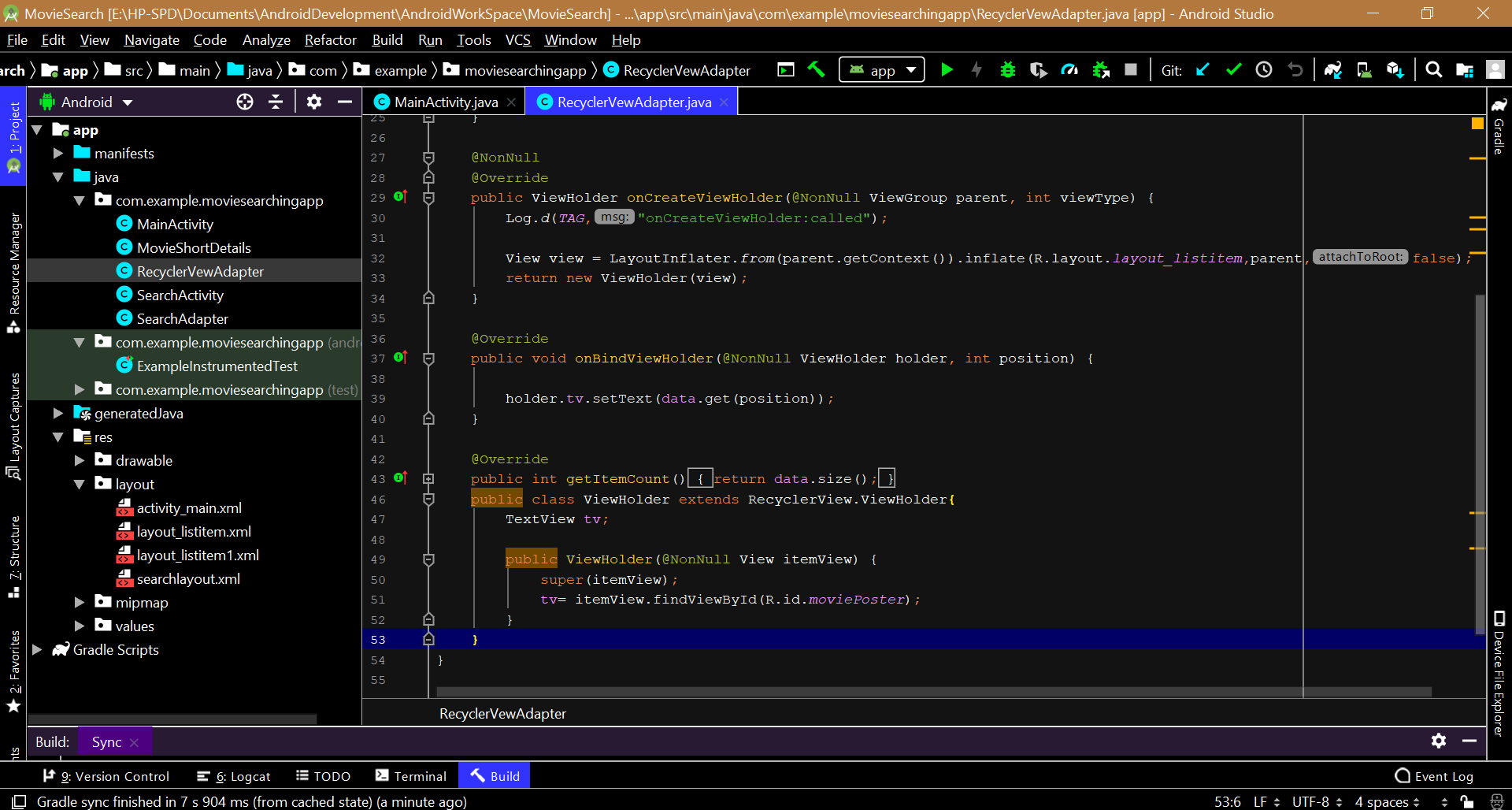
#### Deployment - The deployment essentially deals with the implementation of the project in the large scale factor that deals with the generalised outcome of its implementation on any android phones. As a matter of fact, the scalability itself is so huge that it can be implemented everywhere,ranging from Lollipop, Kitkat and so on.

**System Design**

1. The application starts with the search page which is initially empty.
2. Next, as the user inputs a movie name in the SearchView, the API request are sent to the OMDB API, using the API key, which returns the result in form of a JSON Object.   
   This JSON object is now converted to an Array which is then taken and the required elements like   
   ‘Title’ is extracted.
3. Further, the Movie poster in the form of BitMap is extracted and decoded for each element which occurs in the result of ‘Search’ array.

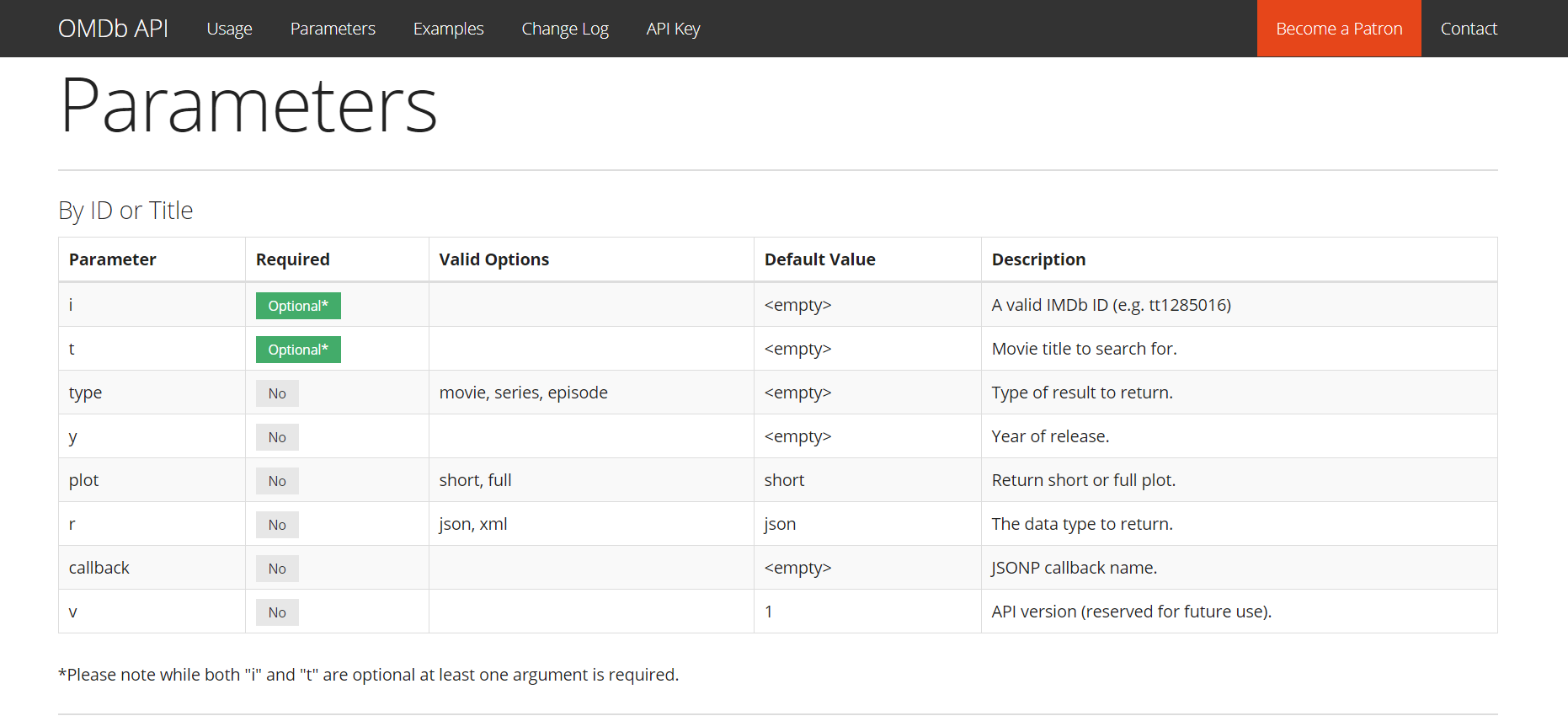


1. For the Description of the movie, the short ‘Plot’ is extracted by using the omdbID from the Search Array and sending another API request.
2. These are populated on the layout, which in turn is connected by the Adapter to the data.   
   The RecyclerView in the application to recycle the views that are out of visibility of the user.
3. The OnBind() method helps in binding the data with the recyclerview , and then inflate It to the user.   
   The results are connected to the XML layout that has been created and hence produces the output.



**Database Schemas**

No database is required like Realm because the data is fetched directly from OMDB API.



**Self Contribution**

1. Bidisha Das (25) : API calling and alignment of the result with the XML layout,

Handling JSON Object and JSON array,  
 Using Volley for connectivity.

1. Vishal Das (28): API calling and analysis of results.
2. Swagata P. Das(26): Creating XML Layout
3. Swarnabha Das(27): Formulating the logic and the algorithm for the project.

**Snapshots of the Application**

