University of
Science and Technology of Hanoi

Mobile Application Development
Final Project Report

Open Library client

Group 1 - BI10 | ICT

November 2021

Contents

1 Introduction

1.1 Overview

For thousands of years, humanity have been using books as a way to save or improve knowledge, and the habit of reading books is still very popular among all ages. But books in physical form are sometimes inconvenient, they are heavy and hard to carry around. We are living in the era of science, and with all the aids from technology, why bother carrying stacks of papers that might weigh kilograms when we can contain hundreds of bookshelves in just a light, half-centimeter thin smartphone?

Realizing the fact that more and more people switch from physical books to E-books, Open Library is an online project intended to create "one web page for every book ever published". However, to improve user experiences, it is better to have a mobile app version of the web. Our project is to build the Open Book Library mobile application on Android to allow users to browse and search for open books.

1.2 Group members

Nguyễn Ngọc Anh BI10-010

Trần Ngọc Hiếu Nam BI10-124

Võ Chí Đạt BI9-066

Tạ Quang Hiếu BI10-065

Dương Đăng Hưng BI10-073

2 Methods

2.1 UML Diagrams

Firstly, before starting to code anything, we design some UML diagrams so that we can specify, visualize, construct and document the components of the application clearly.

2.1.1 Use Case Diagram

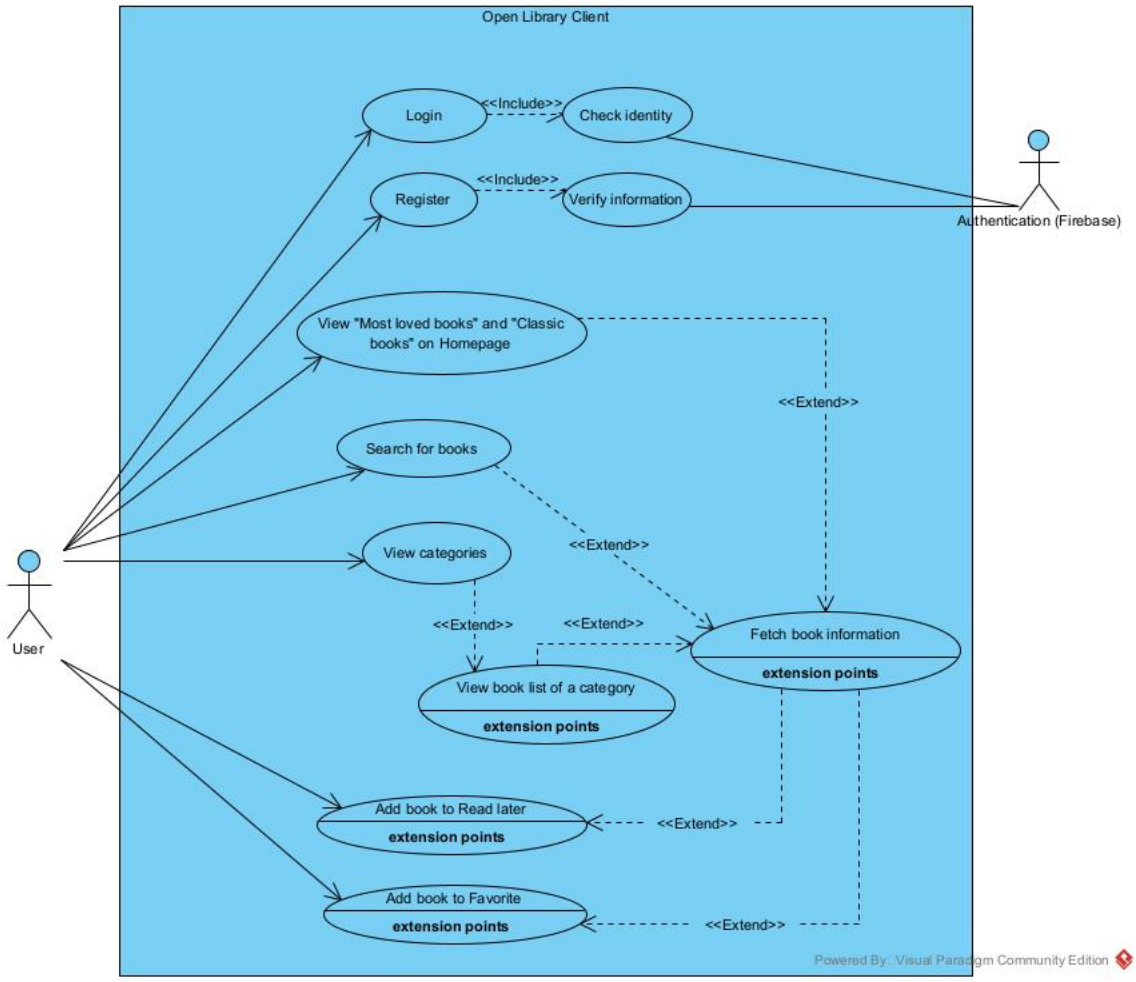

Figure 1: Use Case Diagram

2.1.2 Activity Diagrams

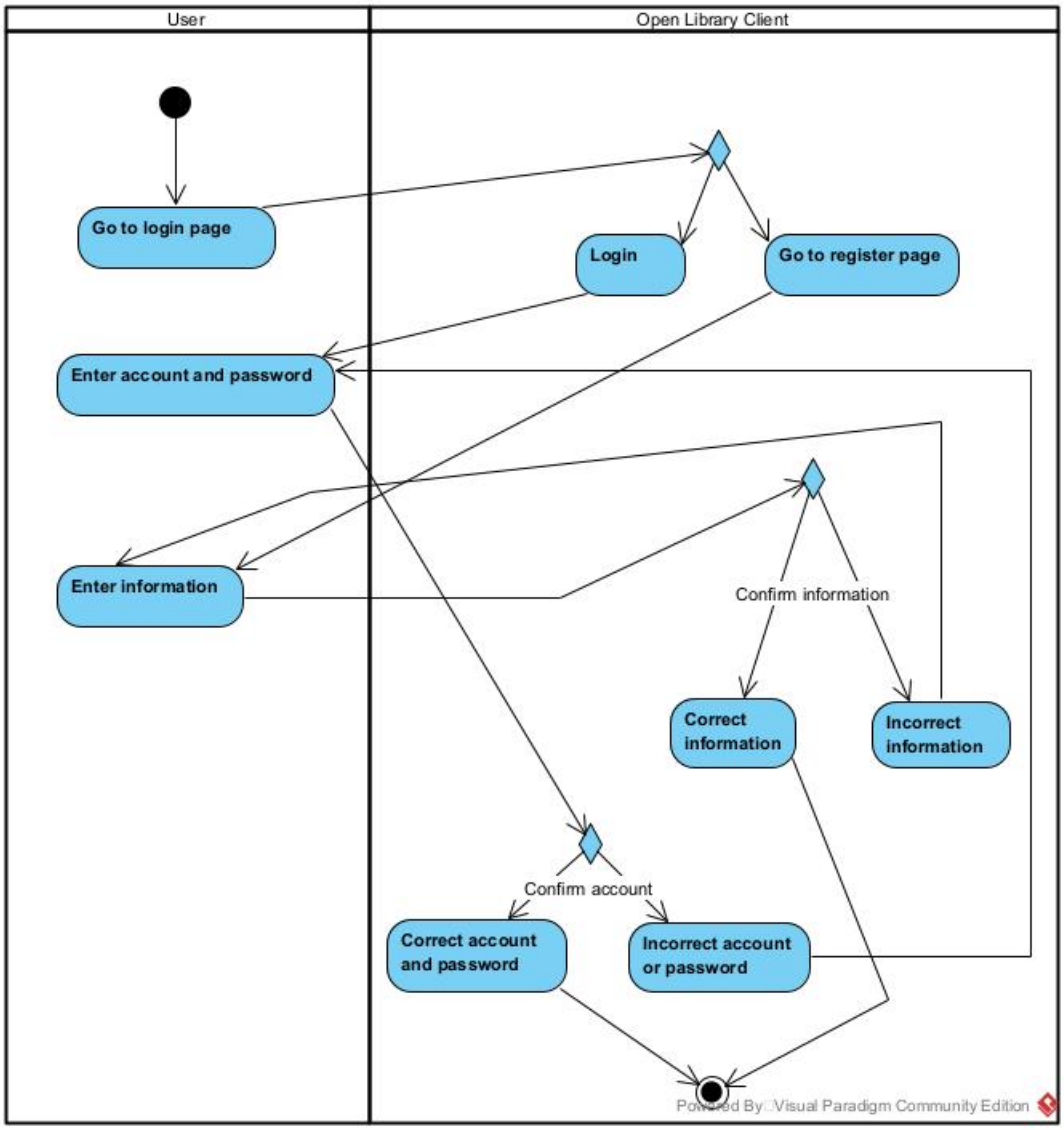

Figure 2: Login Activity Diagram

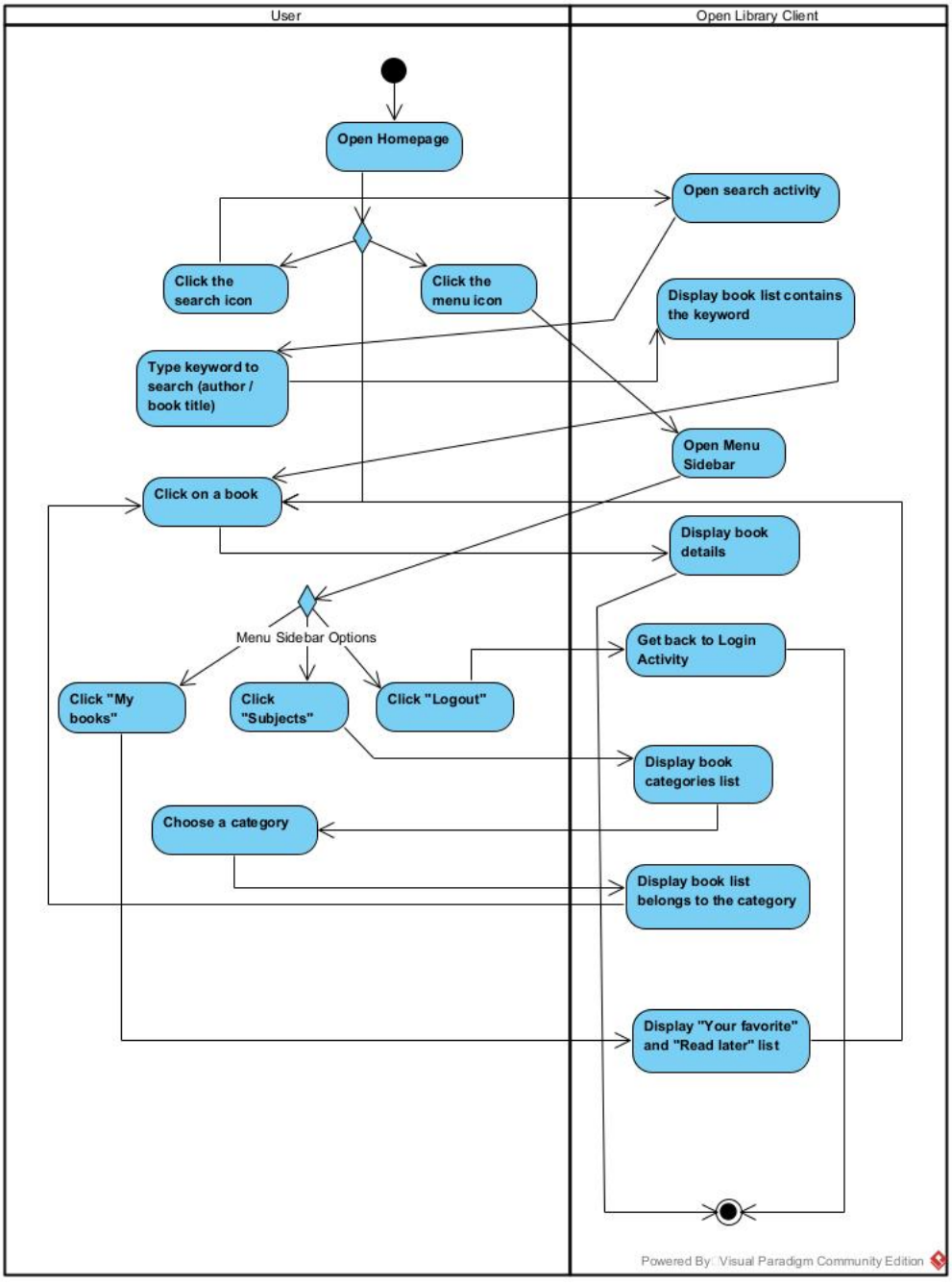

Figure 3: Main Activity Diagram

2.1.3 Class Diagram

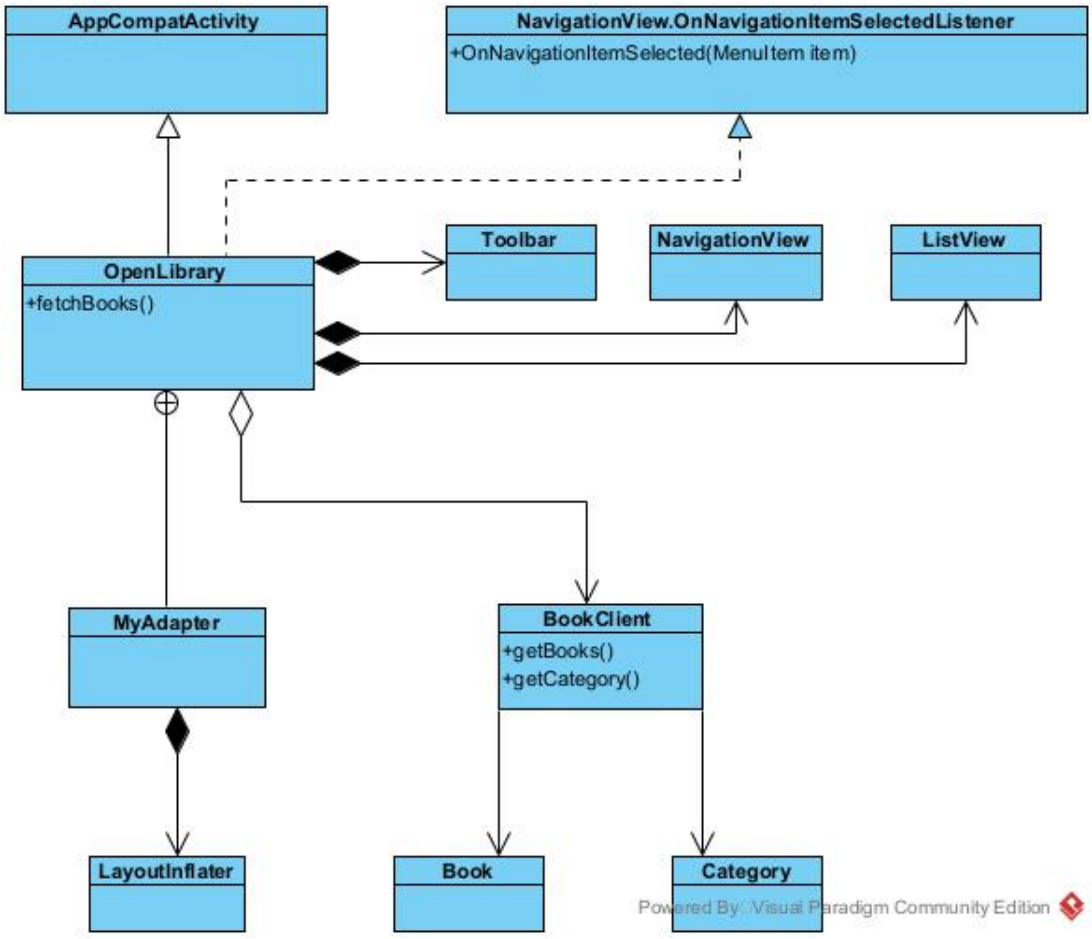

Figure 4: Class Diagram

2.2 Libraries

- Basic Android libraries
- com.loopj.android.http: For networking instead of Volley
  - Simple syntax
  - HTTP requests automatically happen outside the UI thread
  - Make asynchronous HTTP requests, handle responses in anonymous callbacks
  - Can also be used in Service or background thread
  - Tiny size, only 90kb for everything
- org.json: Mainly for handling JSON objects

2.3 API

We use the API provided by Open Library: https://openlibrary.org/developers/api

2.4 Important codes

To request for data from the API, firstly we created a class named BookClient.

```
public class BookClient {
```

Inside this class, we first define the base URL of the API, an AsyncHttpClient instance, and its constructor:

```
    private static final String API_BASE_URL =
        "http://openlibrary.org/";
    private AsyncHttpClient client;

    public BookClient() {
        this.client = new AsyncHttpClient();
    }
```

We also want to create a method to construct API URL depends on what specific data we want to fetch, e.g., book title, category, author names). To do this, we use the base URL of the API and a relative URL:

```
    private String getApiUrl(String relativeUrl) {
        return API_BASE_URL + relativeUrl;
    }
```

Finally, we created methods such as `getBooks()` and `getCategory()` to request data from the API:

```
    public void getBooks(final String query,
        JsonHttpResponseHandler handler) {
        try {
            String url = getApiUrl("search.json?q=");
            Log.i("query", url + URLEncoder.encode(query, "utf-8"));
            client.get(url + URLEncoder.encode(query, "utf-8"),
                handler);
        }
        catch (UnsupportedEncodingException e) {
            e.printStackTrace();
            Log.e("MYAPP", "exception", e);
        }
    }

    public void getCategory(final String query,
        JsonHttpResponseHandler handler) {
        try {
            String url = getApiUrl(query+".json?limit=12&offset=12");
            Log.v("category",
                "subjects/"+query+".json?limit=12&offset=12");
            client.get(url , handler);
        }
        catch (Exception e) {
            e.printStackTrace();
            Log.e("MYAPP", "exception", e);
        }
    }
}
```

After fetching data successfully, we would want to display it on the screen. To do this, in SearchActivity, or OpenLibrary, or CategoryBookActivity, we have a function named `fetchBooks()`:

```
public void fetchBooks(String query, boolean search) {
    this.search = search;
    client = new BookClient();
    ListView listView = (ListView)
        findViewById(R.id.search_items);

    client.getBooks(query, new JsonHttpResponseHandler() {
        public void onSuccess(int statusCode, Header[] headers,
            JSONObject response) {
            Log.v("fetch", query);
            try {
            //unimportant codes here are removed
            for (Book book : books) {
                mTitle.add(book.getTitle());
                mAuthor.add(book.getAuthor());
                mPublisher.add(book.getPublisher());
            }

            //convert ArrayList to String arrays
            String[] aTitle = new String[mTitle.size()];
            String[] aAuthor = new String[mAuthor.size()];

            aTitle = mTitle.toArray(aTitle);
            aAuthor = mAuthor.toArray(aAuthor);

            //set Adapter for the ListView
            MyAdapter adapter = new
                MyAdapter(SearchActivity.this, aTitle, aAuthor);
            listView.setAdapter(adapter);

            //set item on-click listener for the ListView
            listView.setOnItemClickListener(new
                AdapterView.OnItemClickListener() {
                @Override
                public void onItemClick(AdapterView<?>
                    adapterView, View view, int i, long l) {
                    //on-click codes here
                }
            });
```

3 Results

3.1 Most basic functionalities

Using Open Library Client, the user is capable of:

- Login / Logout
- Search for books using name
- View book categories
- View books that belong to a category
- Fetch book details

3.2 User Interface Structure

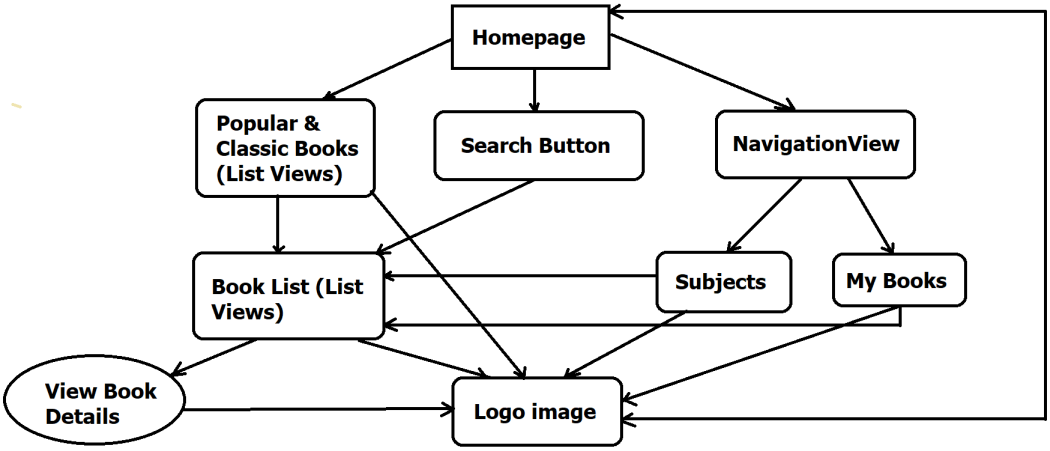

Figure 5: User Interface Structure

3.3 User Interface

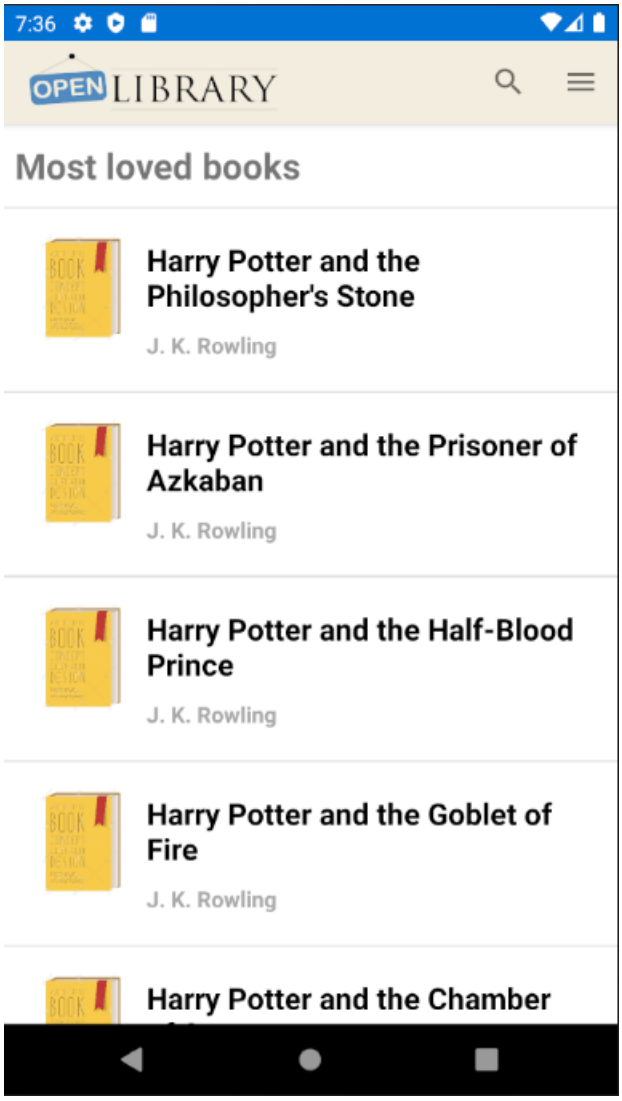

Figure 6: Homepage

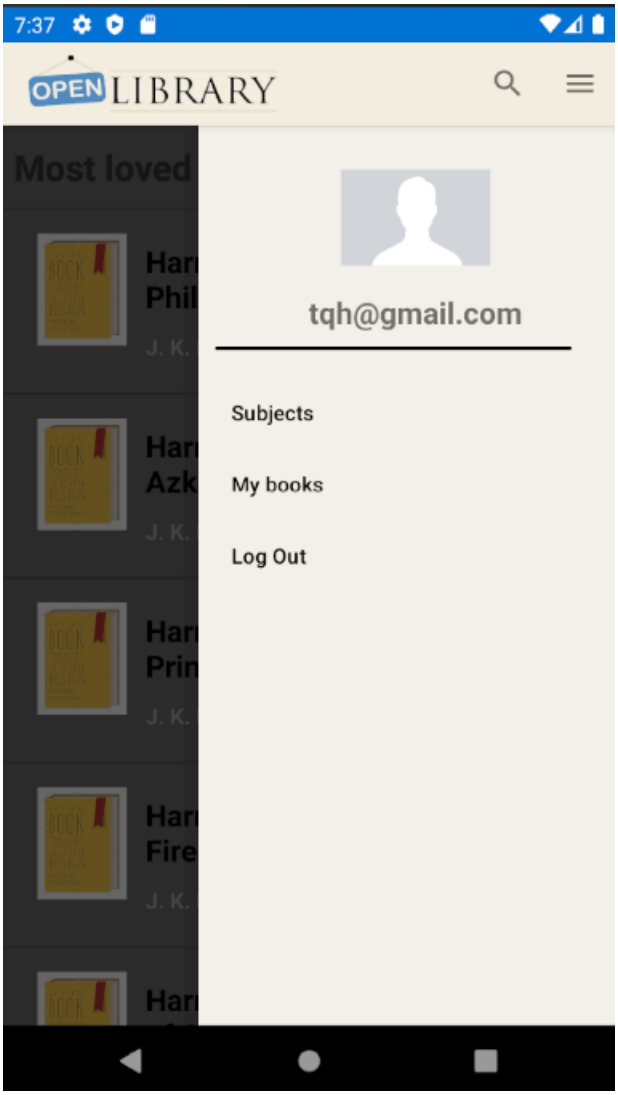

Figure 7: Menu Sidebar

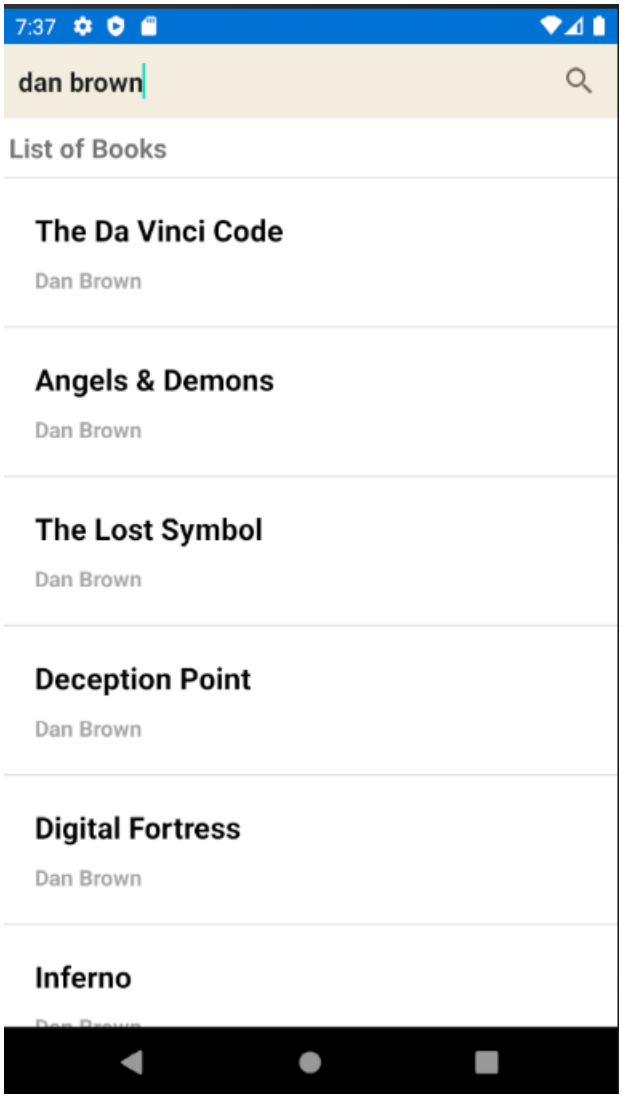

Figure 8: Search Window

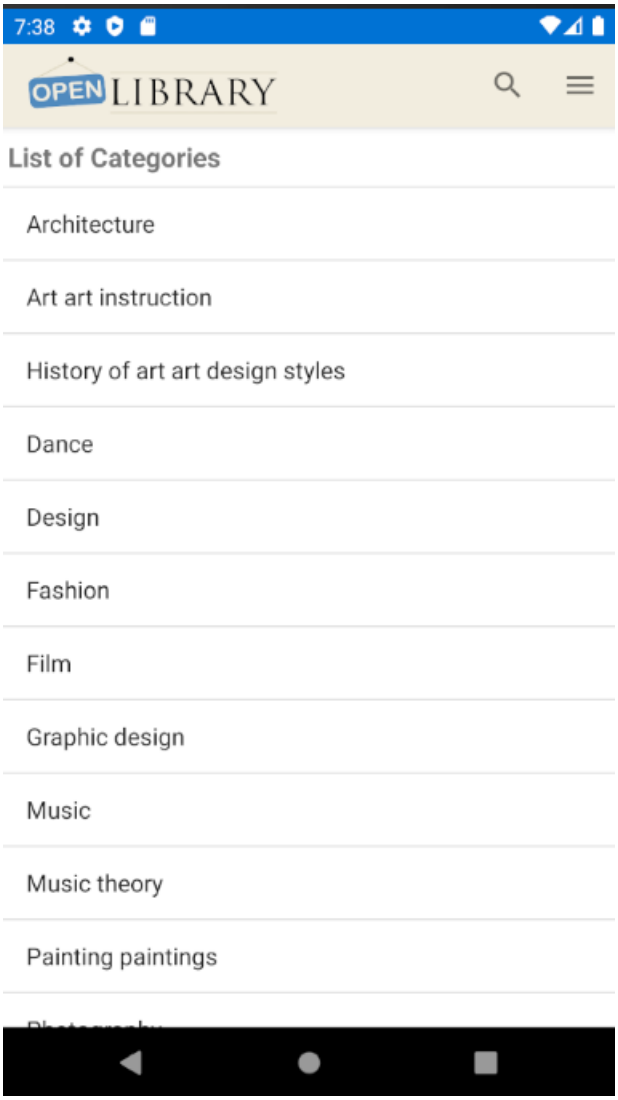

Figure 9: Categories list

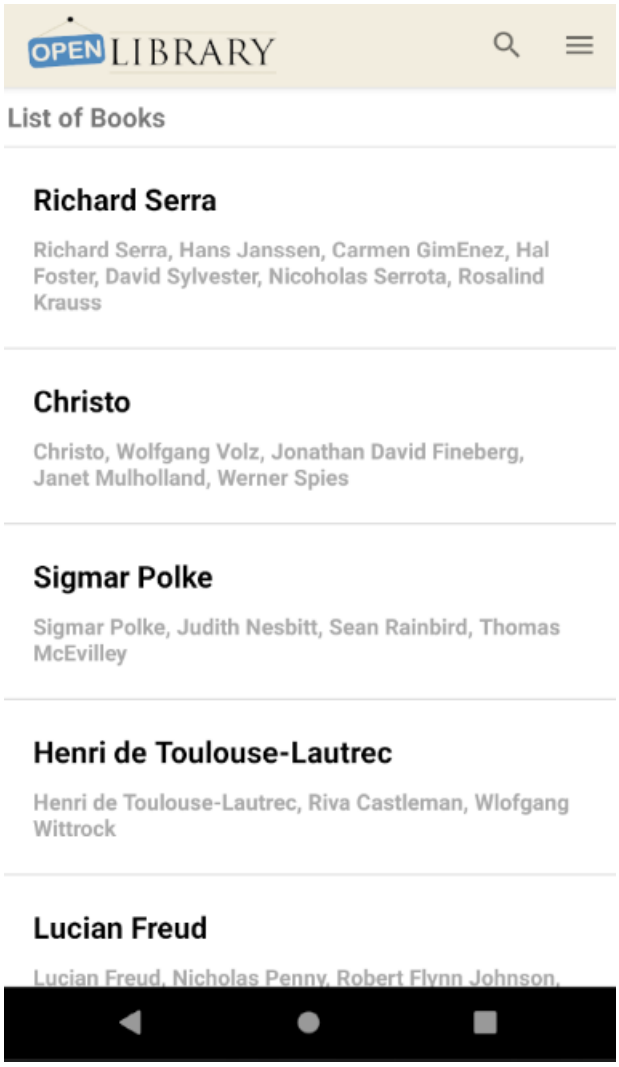

Figure 10: Category books list

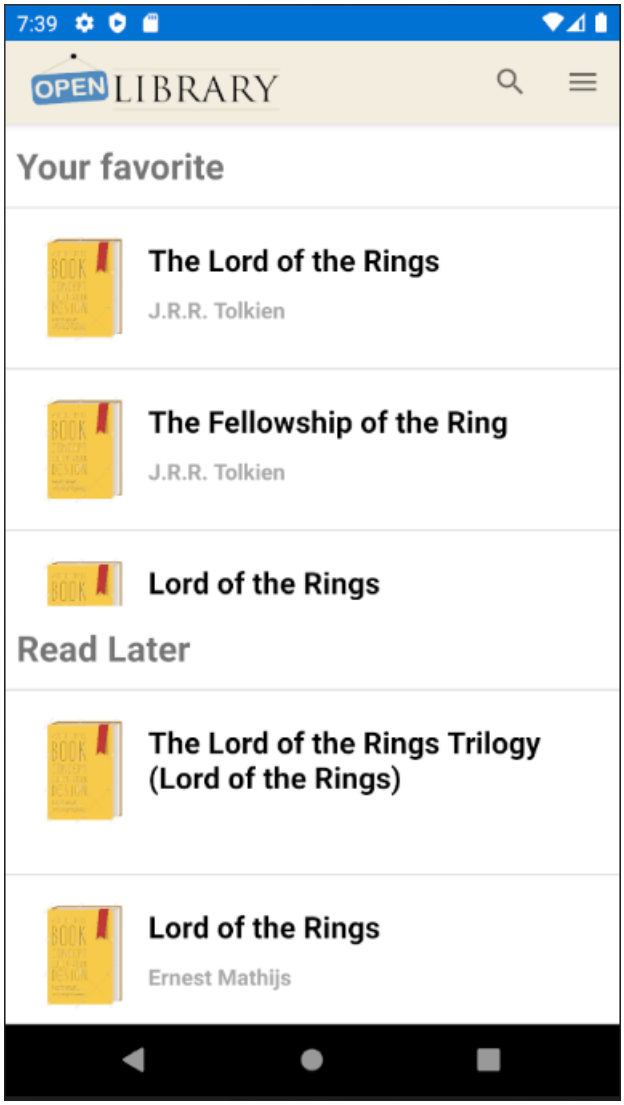

Figure 11: My books

4 Conclusion

4.1 Tasks completed

- Activities and Layouts
- Connect to Network
- Request and Fetch responses from OpenLibrary API
- Fully optimized with multi-threads and AsyncTask
- Basic features of an E-Book client app

4.2 Future improvements

- Implement more features, such as "Add to Favorites" and "Read later"
- API requests are still a little bit slow, try to use pagination
- Also load images into ListView