BIRZEIT UNIVERSITY

Advanced Computer Systems Engineering Laboratory – ENCS5150

ToDo NO.2: Library Management System

1 General Description

We would like to implement a simple library management system using Android Studio. Using our system, the users can do some simple operations for borrowing and keep tracking of the books and their statuses.

2 System UML Diagram

Attached is the UML diagram for the Simple Library Management System that we will implement.

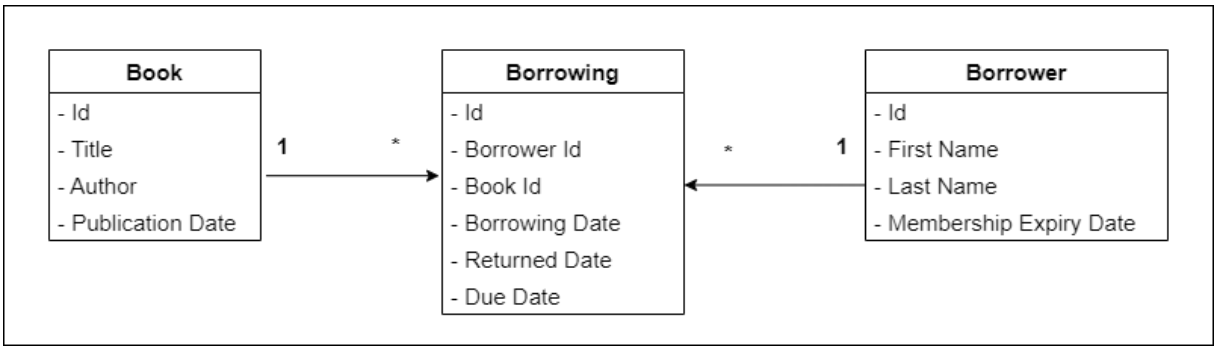

Figure 1: Simple Library Management System UML Diagram

3 Requirements & App Features

The users of our app can view all books that are available for borrowing, view all actively library members, view currently borrowed books by some member, borrow books and return books.

Considering the UML diagram for our system, you are required to:

1. Build the DB schema (tables) for our system using SQLite DB.
2. Specify and use logical constraints that are consistent and suitable for the columns in our database tables, for example Unique constraint and others.
3. Generate in-order auto ids using SQLite.
4. Since we won't be able to use our app to add books and library members, you'll need to seed your database with some consistent and meaningful samples so that you and I can test your app thoroughly. You need to add: 3 books, 3 library members, 9 borrowings considering different books and members. Please, don't add more than these numbers of data to make it easy to test and keep track of the data in your system. And please make your data consistent with the features in your app to help me testing your app correctly. For example:
   (a) Active members: Please make sure you have at least one active member at the time of testing your app. For example set the expiry date to be in the next year.
   (b) Available Book: Please make sure you have a book that's available for borrowing when testing your app. For example, have a book that don't have any borrowings in this year.
   (c) Borrowed Book: Please make sure you have a borrowed book at the time of testing your app. For example, have a book that's borrowed from a date in the past to the next year.

   Note: Please make sure to delete all the other data that you might used in testing to have clean data for your app when being tested by me.

The features of our android app are specified as follows:

1. **Home Page**

   It's the entry point of our app. It includes four main buttons that will enable the next functionalities: List Available Books, List All Actively Library Members, List Currently Borrowed Books and Borrow Book. Clicking any of these buttons will open another page or view to handle the needed functionality as described later.

2. **List Available Books Page**

   This page will be accessible through the **List Available Books** button in the Home page. In this page, the user should be able specify a date range and then click on a button (Search Button) to search for and list all of the books that are currently available for borrowing in the specified date range.

3. **List All Actively Library Members Page**

   This page will be accessible through the **List All Library Members** button in the Home page. In this page, the user should be able to view all the actively library members on the current date (the date of viewing the page).

4. **List All Actively Borrowed Books Page**

   This page will be accessible through the **List Currently Borrowed Books** button in the Home page. In this page, the user should be able to specify the Id of the Library Member to see currently borrowed books for that member. After specifying the library member, the user should be able to click the **Search** button to view all the currently borrowed books with main borrowing details such as, Borrowing Date and Due Date as well as having a button called **Return Book** on each record in he displayed list. Clicking the return book should return the book, that's associated with the return book button. Finally, when returning the book successfully, the displayed list of the currently borrowed books should be refreshed.

5. **Borrow Book Page**

   This page will be accessible through the **Borrow Book** button in the Home page. In this page, the user should be able to enter or select the data needed for borrowing a book. It also has a button called **Borrow** that will enable the user to borrow the book using the specified data if they are valid. If the borrowing done successfully the user will be directed to the **List All Actively Borrowed Books Page**. Otherwise, the user will remain in the same page.

4 UI Design

1. Some UI Components have been specified in the app's feature description, so make sure to to use the specified components.

2. The components used in listing and displaying the details of the listed item in all of the: List Available Books Page, List All Actively Library Members Page, List All Actively Borrowed Books Page and Borrow Book Page are not specified. You can use the components you want and the details as you like as long as the design is clear, user-friendly and consistent with the requirements. Note: Make sure the fields used for entering the date fields are user-friendly.
3. In all pages, other than the Home Page, you need to have a button Called **Home** to return to the home screen.
4. Show toast messages to notify the user about the problem in case of any validation errors in your app.
5. The date fields mentioned in the app feature should be user friendly so I can distinguish the format for the date: YY/MM/DD or other formats.

5 Notes

1. Marks are distributed as follows: 3.35 for UI design, 6.15 for the functionality and the handling logic.
2. You must follow the following format when naming your app: (UNIID_ToDo{ToDo#}). For example, if your university id is: 1180865. Then the app name should be: 1180865_ToDo2.
3. As agreed in the beginning of the lab, we will be using a Pixel 3a XL device with Graphics = Software and API 26, which corresponds to Oreo. So, please make sure to match these requirements to avoid any unexpected issues when I test your app.
4. Please make sure to submit all the requested files in the todo (shown in the next section). Missing any of them means loosing some points.
5. ToDos are individual work, and cheating is strictly prohibited, resulting in a score of 0.

6 What to Submit?

1. **Project .zip file (Size in KB)**: From Android Studio: File → Export to zip file.
2. **APK file**: From Android Studio: Build → Build Bundle(s) / APK(s) → Build APK(s). This file should be submitted as a separate file.
3. **SQL Scripts**: Please submit all the sql scripts that you've used to create the tables and to seed your tables.

7 Deadline

28/11/2023 at 11:59 PM.