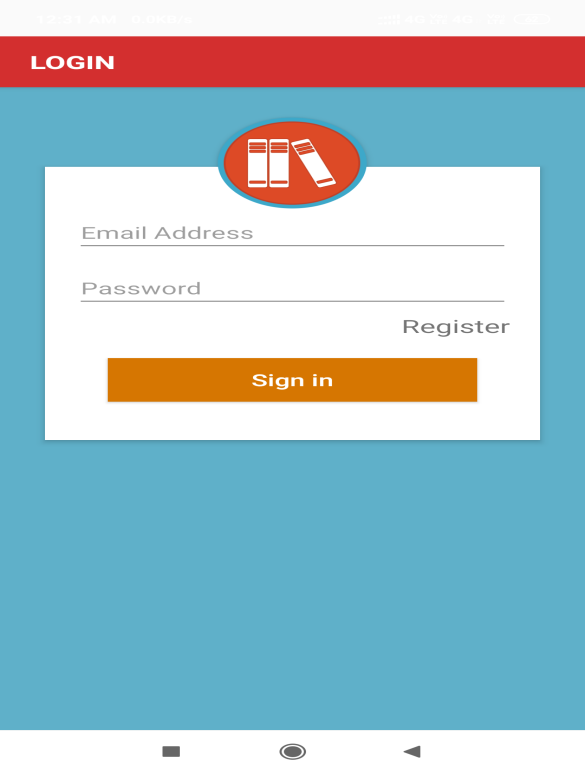
README. SID-17103040 HARSH BANSAL,17103001 PARAS GUPTA

**LIBRARY MANAGEMENT SYSTEM**

Android app that is developed using Android Studio SDK to implement the basic functionalities of a library and also to automate some of these processes. For the server end, Firebase by Google is used to store, retrieve and modify the database in the real time. Here Firebase is based on **NoSql.**

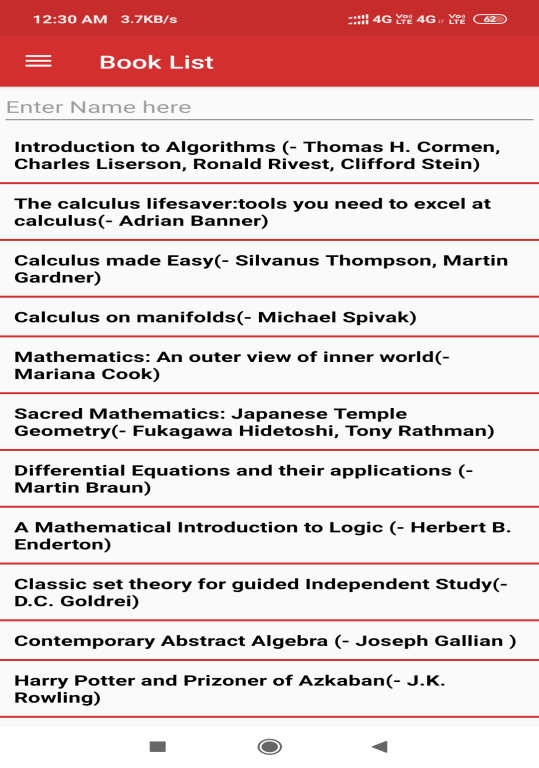
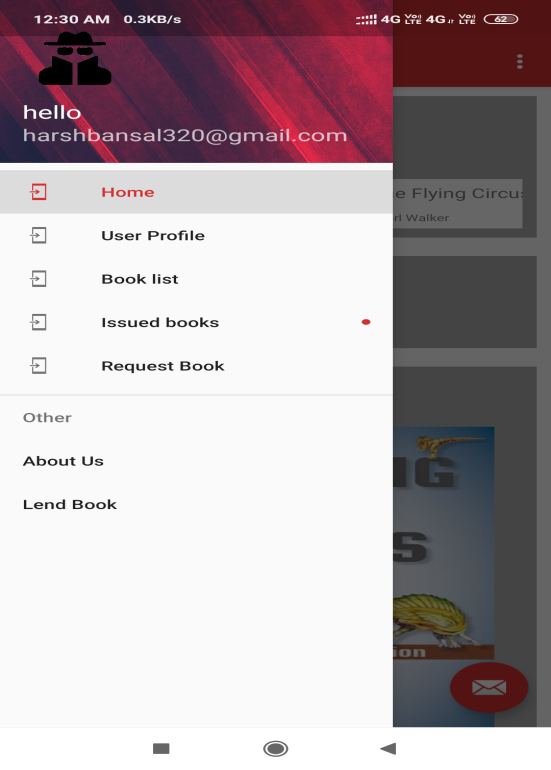
The access to app can be achieved by a username and password. Username will be used by both the student/user and the librarian to enter into the application.



Overview:

For Users:

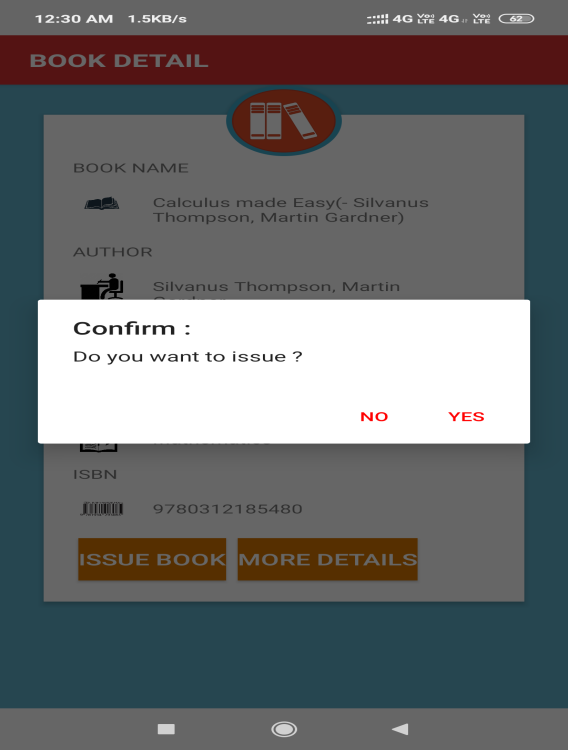
1. Search for a book in the library (either by book name and author name).

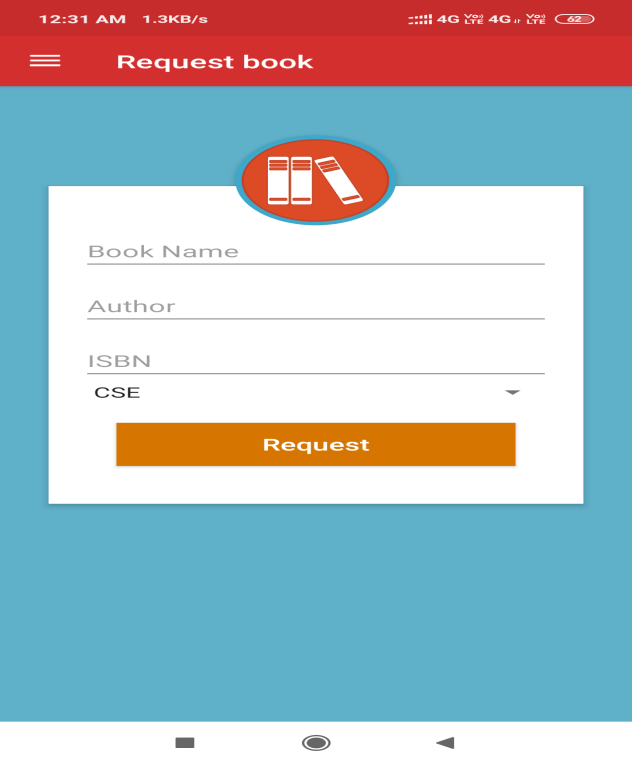
1. Display the details of searched book in a new request.



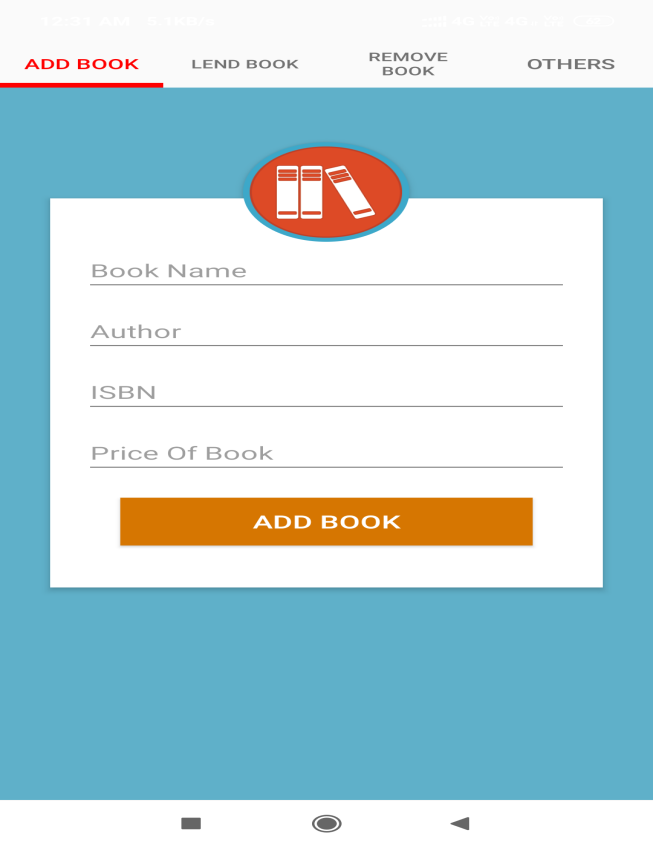
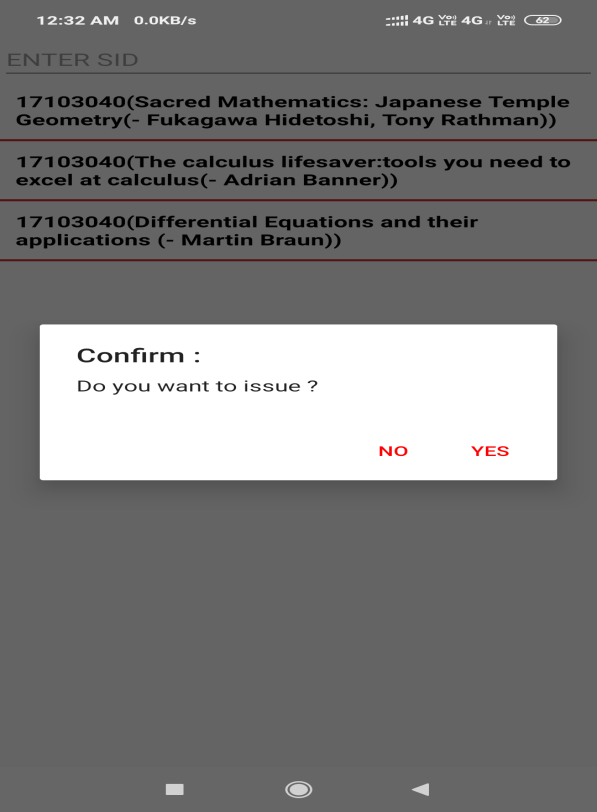
1. Make an issue request for a book on the search result menu. The request will be sent to the backend Firebase server (NoSql).



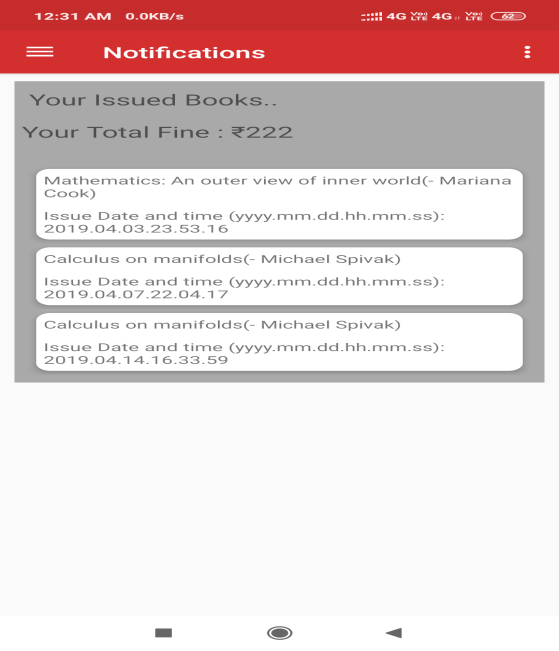
1. Also, if the book is not available, user can request for the book. The request will be stored on the real – time server of Firebase. Also, if a book has been requested multiple times, then this list of requested books is sorted according to the no. of requests made for it (i.e the most requested books is at the top of list, hash map is used here).



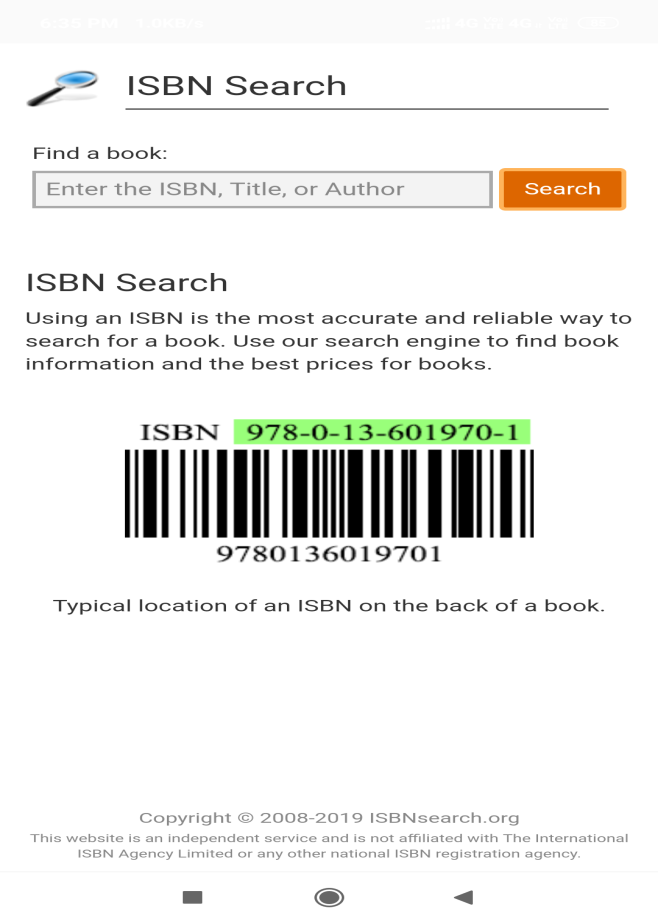
1. The user can also add a list of books he has which the other users can view also. So, here a Lending System has been developed in which one student can lend his book to another student. Also the student which wants to lend a particular book can contact the owner of the book. His/Her mail will be provided for this purpose.

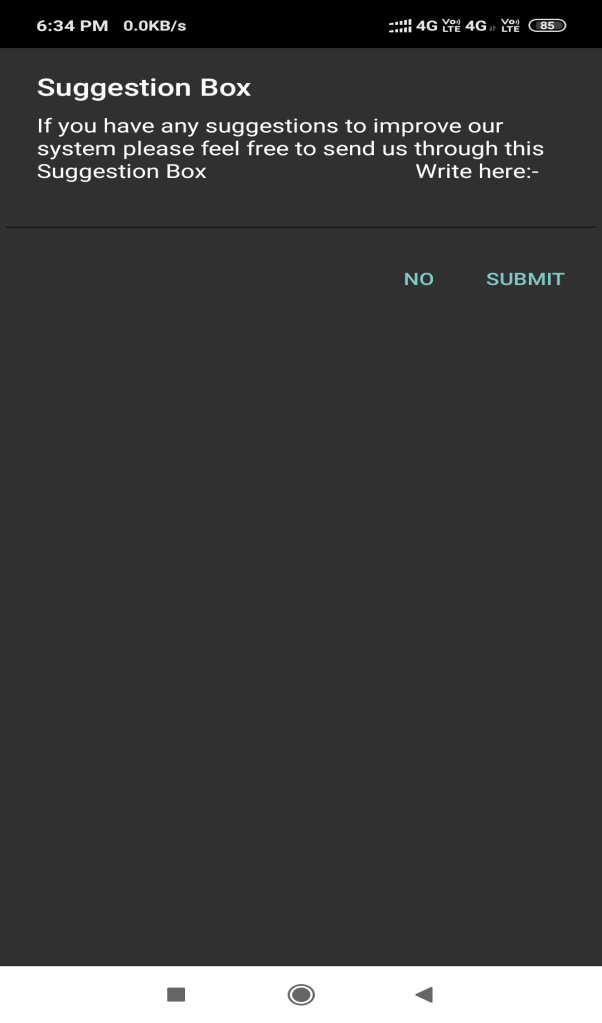
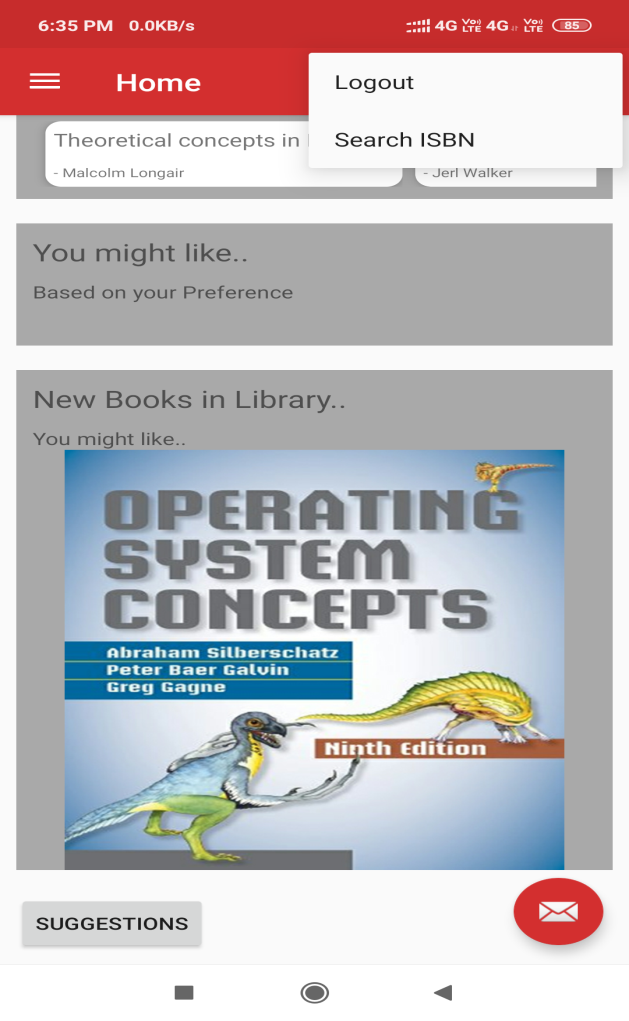
1. The user can also view his issued book list and the fines against them. He can also view his user profile and his/her fines.

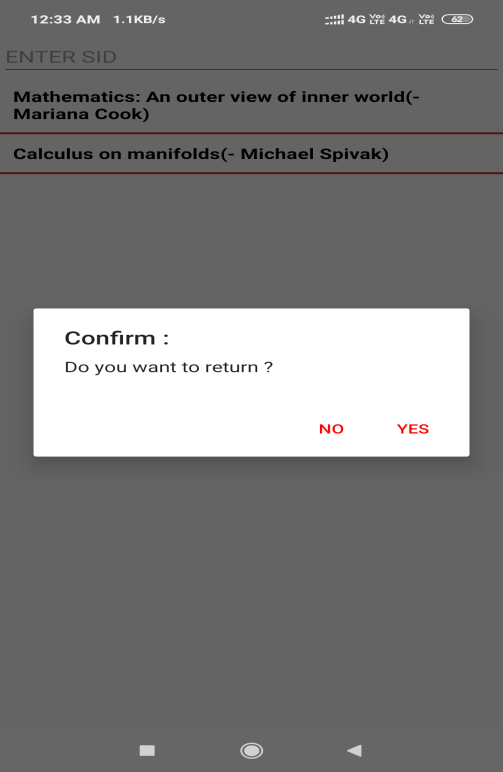
1. The user can also search for a book’s information (like its ISBN no., its amazon-buy link, etc) by entering the name of author of book or book’s name. For this purpose, we have used a third party website [www.isbnsearch.org](http://www.isbnsearch.org). We redirect to this site for this purpose. (We are still on our app, i.e. the website is running in the application environment itself).



1. Also, there is a suggestion box on first user page of the app which the user can use to provide suggestions, criticism etc. This will be stored on the server (Real-time Firebase). So, the librarian can also view these suggestions and can work on them.

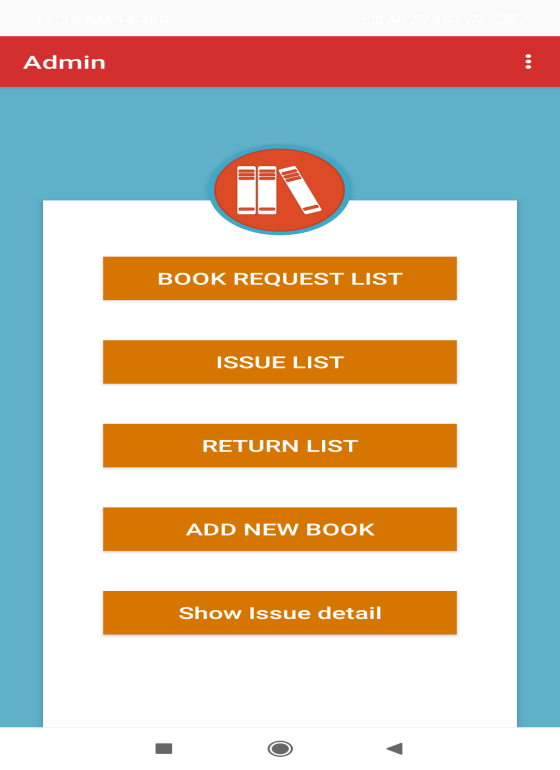
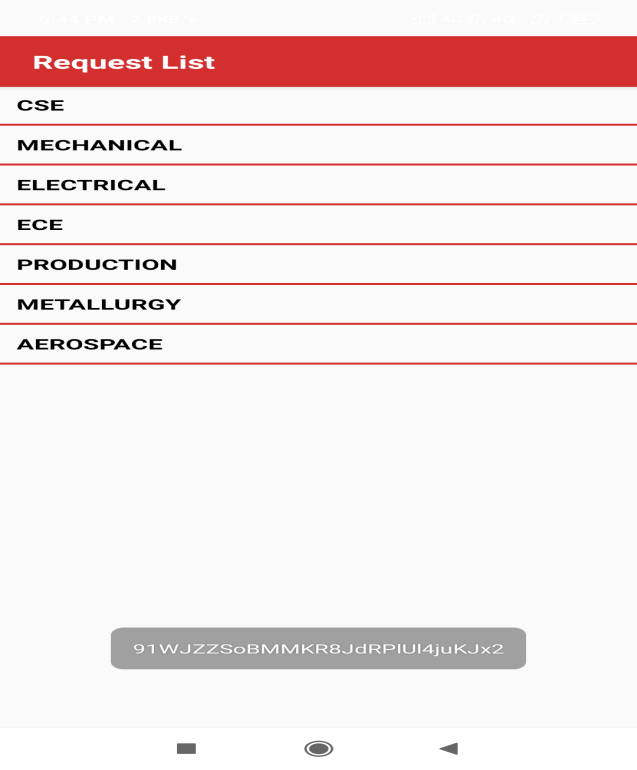
 

1. For returning a book, the student has to go the library and manually return the book.

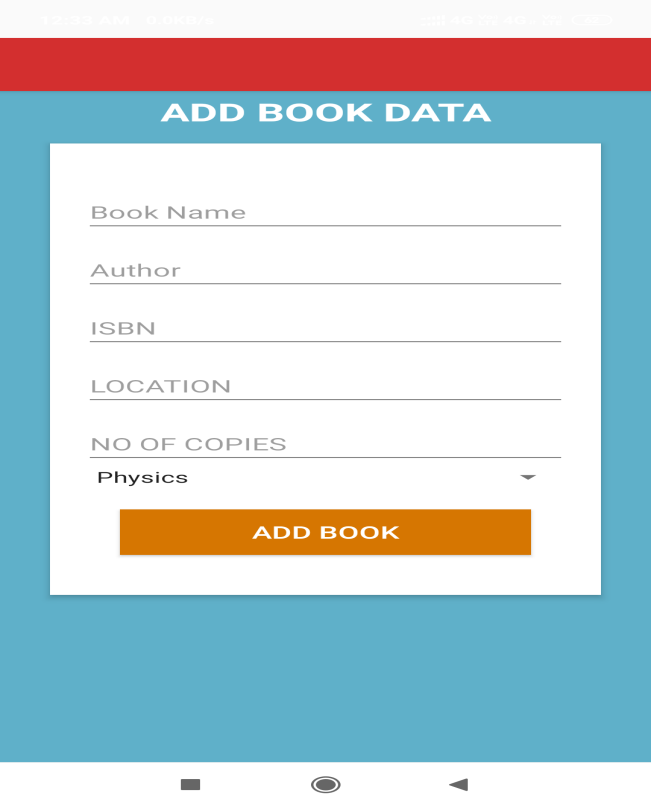


For Admin/Librarian:

1. The librarian can view the requested books made by the users of the app.

1. Also, the librarian can add a book in the database of books at the server. The list is updated instantly at the backend server (NoSql).



1. The librarian will approve the issue requests of books made by students when they come to the library. He will access the issue request list from the real-time server (which is sorted according to time-stamp) and then will update the database of books. (the no. of copies will be updated automatically in the backend server instantly).
2. When the user has come to return the book, the librarian will update the books in the real-time database. (the no. of copies will be updated automatically in the backend server instantly).

To achieve this, there are no. different componets(activities and fragments) in this app:

**/\* The code of these activities and fragments can be found in /app/src/main/java/com/example/navigation \*/**

<https://github.com/harsh320/library-management-system>

Books Fragment: Will display names of books using list view.

BooksDetail activity: Will display details of book activity page.

HomeFragment: Main fragment of application showing home page containing different options like suggestions, etc.

NotificationFragment: Will show the user issue list and fines.

RequestFragment: Contains the fragment page in which book request options are given.