**PUNE INSTITUTE OF COMPUTER TECHNOLOGY,**

**DHANKAWADI, PUNE**

**A MINI-PROJECT REPORT**

***ON***

**DIGILIB APP**

SUBMITTED BY

**NAME: NACHIKET ERLEKAR**

**ROLL NO.: 31436**

**CLASS: TE 4**

GUIDED BY

**Prof. R.V. Bidwe**

**COMPUTER ENGINEERING DEPARTMENT**

**Academic Year : 2019-20**

1. **ABSTRACT :**

For any institute a library is the single most important source of all information.

Library of PICT is well known everywhere for its richest source of information. Currently it has a trove of over 35000 books and almost an equivalent amount of journal journals periodicals and standards.

Currently the Library has a website on which information about the latest status of library is visible. It shows the availability of books and other information. PresentlyStudents and faculty of PICT have to submit a demand slip for a book physically in the library on a page. They also have to pay a fine for late return physically in college accounting department.

We propose an innovative android application called “Diglib” for PICT library. Through this app Students and Faculty will have all information of latest status of library in their hand. Using this app they can find out the availability of any book in the library. They can submit demand slips for books that are all issued. They can also request for a book in the library through app.

The app keeps track of book return date. It automatically calculates the fine if the book is not returned in time and notifies the user accordingly. The app support payment through popular payment apps. User can pay fine digitally. This app reduces the effort of students and faculty members in searching for a book and issuing it. It also reduces the hassle in paying the fine. It also reduces paperwork and effort of library management staff.

**II. TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| **SR. NO.** | **CONTENT** | **PAGE NO.** |
| 1. | Introduction | 4 |
| 2. | Design and Implementation | 5 |
| 3. | Testing / Result and analysis | 12 |
| 4. | Screenshots | 14 |
| 5. | Conclusion and Future Enhancements | 22 |
| 6. | References | 23 |

**III. LIST OF TABLES**

|  |  |  |
| --- | --- | --- |
| **SR. NO.** | **TABLE NAME** | **PAGE NO.** |
| 1. | Test cases | 12 |

**IV. LIST OF DIAGRAMS**

|  |  |  |
| --- | --- | --- |
| **SR. NO.** | **DIAGRAM NAME** | **PAGE NO.** |
| 1. | Architecture diagram | 7 |
| 2. | Activity Diagram | 8 |
| 3. | Use case diagram | 9 |
| 4. | Data flow diagram | 10 |

**CHAPTER 1**

**INTRODUCTION**

1. PURPOSE

The purpose of this document is to present a detailed description of the DIGILIB APP (Digital Library App System). It will explain the purpose and features of the system, the interfaces of the system, what the system will do, the constraints under which it must operate and how the system will react to external stimuli. This document is intended for both the stakeholders and the developers of the system and will be proposed to the SDL lab teacher for its approval.

2. SCOPE

This software system will be a Digital Library App for students of Pune Institute of Computer Technology (Pune).This system will be designed to ease the life of students when it comes to issue books from PICT library.

This app will curb the need for students to be physically visiting the library and even more.

Currently students have to physically go to the library to check the availability of specific book. Through our app this need is completely eliminated.

This app can be used in the libraries of various schools and colleges to convert traditional libraries into smart l ibraries.

3. OVERALL DESCRIPTION

Students can check the availability of any book in the library directly on their app. They can also receive a notification when the book will be available in the library. Additionally the system will inform the students of fine on currently issued books. Students will also be able to demand a book not present in PICT library inventory. This app use Firebase (Cloud Firestore) for Database Management. This app however does not allow facility to issue a book directly. Students will still have to physically issue a book from the Library. As a new feature we have implemented payment gateway to pay fines on books directly without visiting the college accounts office.

4. MOTIVATION

Digitizing PICT library for students and teachers.

Need of physical presence in the library for every activity is a hassle.

Ease of access, checking the status of books, generation of demand slip for books, fine system for late submission of the issued book.

Making the process of issuing, fine and request of book is made completely paperless and process is taken online.

**CHAPTER 2**

**DESIGN AND IMPLEMENTATION**

2.1 PRODUCT PERSPECTIVE

The proposed Library Management System will take care of the current book detail at any point of time. The book issue, book return will update the current book details automatically so that user will get the update current book details.

2.2. REQUIREMENT ANALYSIS

Hardware required for developing the app:

* Computer with intel i5 or greater processor.

Software required for developing the app:

* Android Studio by IntelliJ
* Java SDK
* Firebase account

Requirements for running the app:

* A smartphone
* Android lollipop or greater
* Working internet connection

2.3 DEPENDENCIES

* Working internet connection
* Continuous updation of database required
* Database hosted by Google
* Third party payment API

2.4 PRODUCT FUNCTIONS

* Users have their own account with appropriate account security.
* User can view the list of books in the library and their availability.
* User can fill a demand form for those books which are not available and get a notification when the book is available.
* Users can file an issue request for a book right from the app and avoid the hassle of finding the book in the library.
* User will also be able to track the fine for the book, if any.

2.5 FUNCTIONAL REQUIREMENT

1. REGISTRATION

Description : First the user will have to register/sign up. There are two different types of users.

The library manager/head

Regular person/student : The user has to provide details about his/her name, address, phone number, email id.

1.1 SIGN UP

Input: Detail about the user as mentioned in the description.

Output: Confirmation of registration status and a username and password will be set.

1.2 LOGIN

Input: Enter the username and password provided

Output : User will be able to use the features of the software.

2. MANAGE BOOKS BY USER

2.1 SEARCH

Input : Enter the name of the desired book.

Output : A list of books related to the keyword.

2.2 ISSUE BOOKS

State : Searched the book user wants to issues.

Input : Enter the book name.

Output : conformation for book issue and apology for failure in issue.

Processing : if selected book is available then book will be issued else error will be displayed.

2.3 DEMAND SLIP

Input: Enter the book name..

Output: Submit details regarding demand slips and view all the demand slips.

Processing: The slip is sent to admin level interface for further proceedings.

2.4 FINE SYSTEM

Input : Check for the fines.

Output : Details about fines on different books issued by the user. Total fine to be paid.

Processing : The fine will be calculated, if it crossed the date of return and the user did not renewed if then fine will be applied by Rs 10 per day.

2.5 PAYMENT

Input : User choice of payment.

Output : Redirection to desired payment option.

Processing : Fine is calculated by the application and passed to the payment API.

2.6 ARCHITECTURE DIAGRAM OF DIGILIB APP:



DIAGRAM 1

2.7 ACTIVITY DIAGRAM OF DIGILIB APP:

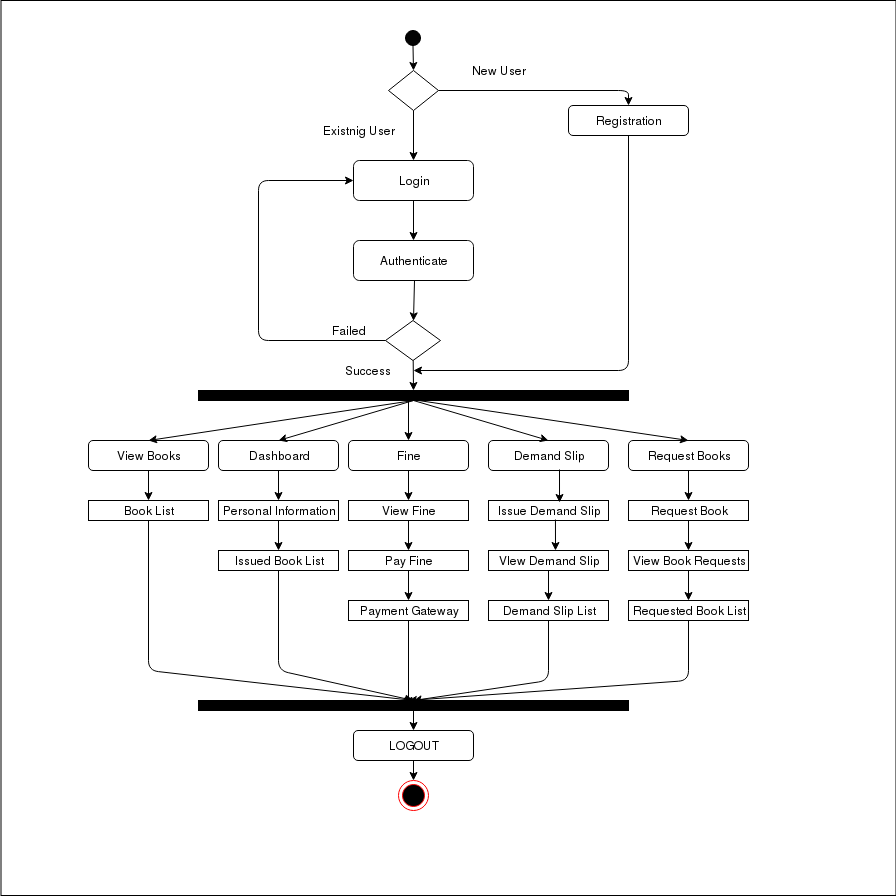


DIAGRAM 2

2.8 USE CASE DIAGRAM

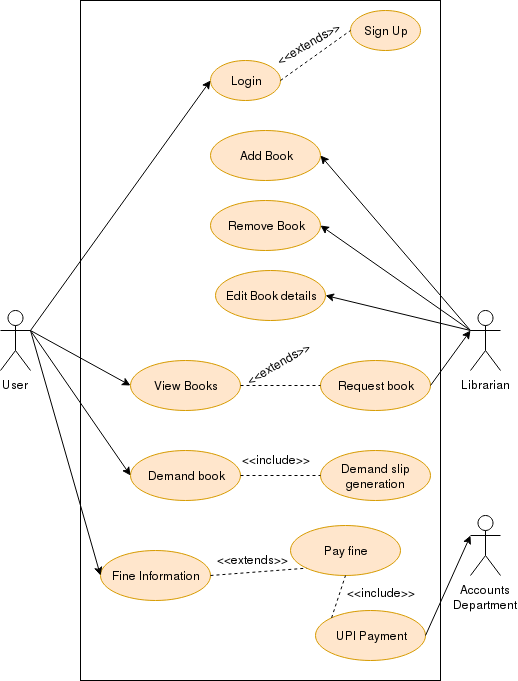


DIAGRAM 3

2.9 DATA FLOW DIAGRAM

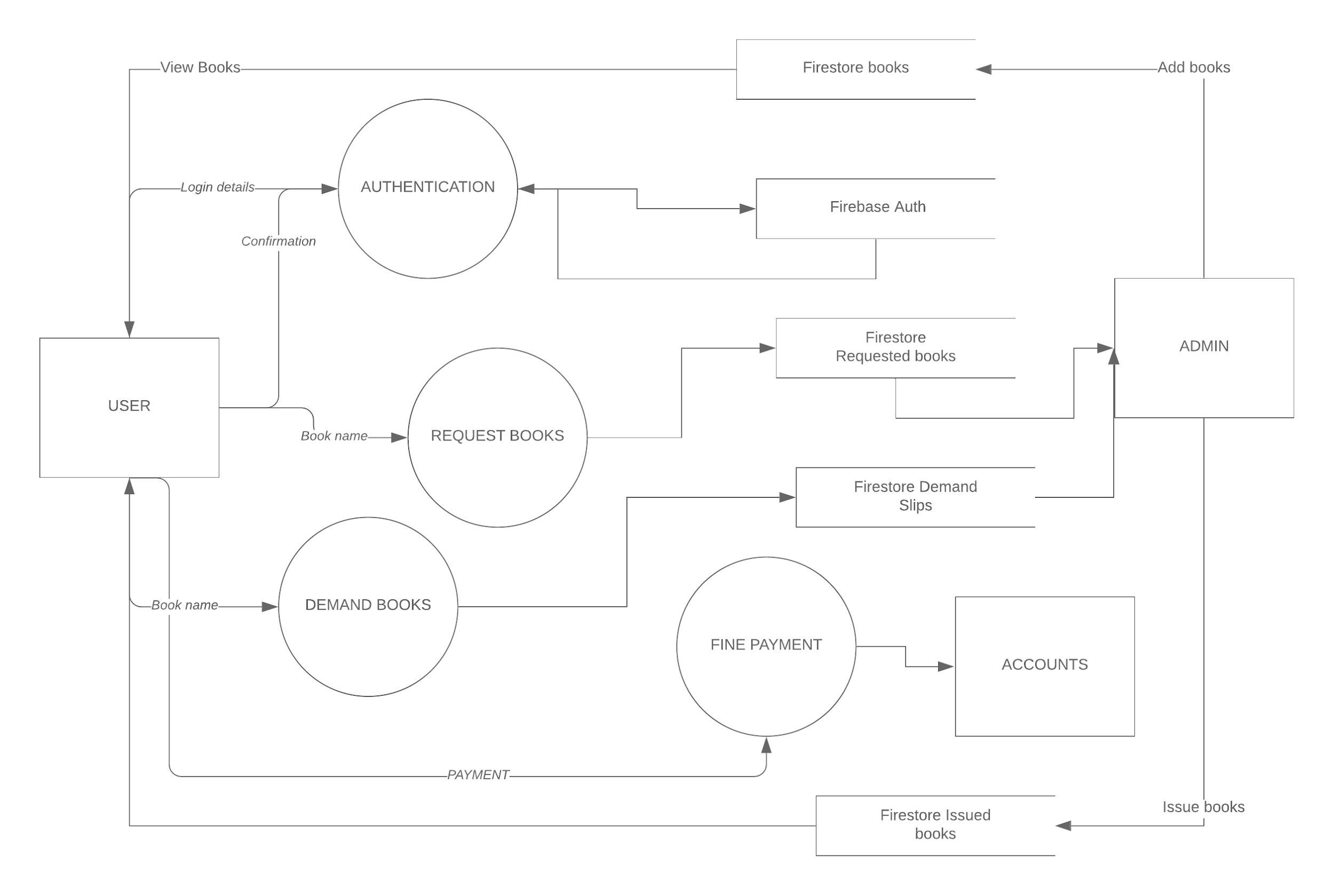


DIAGRAM 4

2.10 IMPLEMENTATION

The Digilib android app was implemented using the Android studio app.

Android Studio is the official integrated development environment for Google's Android operating system, built on JetBrains' IntelliJ IDEA software and designed specifically for Android development. It is available for download on Windows, macOS and Linux based operating systems.

It has a strong editor tool for developing creative UI and emulators for different versions to test and simulate sensors without having actual **Android** devices.

# Installing Android Studio in Ubuntu:

STEP 1:

* We started by installing Oracle JDK 8 with the following commands:

sudo add-apt-repository ppa:webupd8team/java

sudo apt-get update

sudo apt-get install oracle-java8-installer

sudo apt-get install oracle-java8-set-default

STEP 2:

* Download the Android Studio package for Linux and extract it to the home directory.
* To launch Android Studio, open a terminal, navigate to the android-studio/bin/directory, and execute studio.sh
* Select whether you want to import your previous Android Studio settings or not, then click OK.
* Set the ANDROID\_HOME environment variable to the location of your Android SDK installation:

sudo gedit ~/.bashrc

export ANDROID\_HOME=/home/user\_directory/Android/Sdk

export PATH=${PATH}:$ANDROID\_HOME/tools:$ANDROID\_HOME/platform-tools

export JAVA\_HOME=/usr/lib/jvm/java-8-oracle

STEP 3:

Install SDK Platforms

* You need to install an SDK before you can jump into building Android apps.
* Click on Configure -> SDK Manager to open Android SDK Manager.
* Select the latest API in order to test against target build, e.g. API 19 (Android 4.4.2), and the Android Support Library and Android Support Repository packages in Extras. Then install the selected packages.

**CHAPTER 3**

**TESTING/RESULT AND ANALYSIS**

3.1 TEST CASES

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| SR. NO. | TEST CASE DESCRIPTION | EXPECTED RESULT | ACTUAL  RESULT | REMARK |
| REGISTRATION PAGE: | | | | |
| 1. | Enter name, roll no, department, class, username and password and click register. | New account created successfully and logged in. Dashboard page will open. | Same as expected | User shouldn’t be already registered. |
| LOGIN PAGE | | | | |
| 1. | Enter correct username and password | User successfully logged in. Dashboard page will open. | Same as expected |  |
| 2. | Enter incorrect username and password | Toast showing incorrect username or password | Same as expected |  |
| DASHBOARD | | | | |
| 1. | Clicks an option from the navigation drawer | Activity corresponding to that option opens | Same as expected |  |
| 2. | Clicks logout from the navigation drawer | User successfully logged out and login activity opens | Same as expected |  |
| VIEW BOOKS ACTIVITY | | | | |
| 1. | Type book name in search bar | Books corresponding to the keywords and their availability is displayed. | Same as expected |  |
| DEMAND SLIP ACTIVITY | | | | |
| 1. | Enter the book's name and click add. | Toast showing demand slip successfully added. | Same as expected |  |
| 2. | Click view demand slips | List of demand slip displayed | Same as expected |  |
| DEMAND SLIP LIST | | | | |
| 1. | Click the delete icon in front of the slip to be deleted. | Slip is deleted from the list | Same as expected |  |
| FINE SYSTEM | | | | |
| 1. | Click pay fine | Asks for preferred UPI app available in your phone | Same as expected | At least one UP app should be present on the device |
| REQUEST BOOKS | | | | |
| 1. | Enter the book's name and click add. | Toast showing request successfully added. | Same as expected |  |
| 2. | Click view requests | List of requests displayed | Same as expected |  |

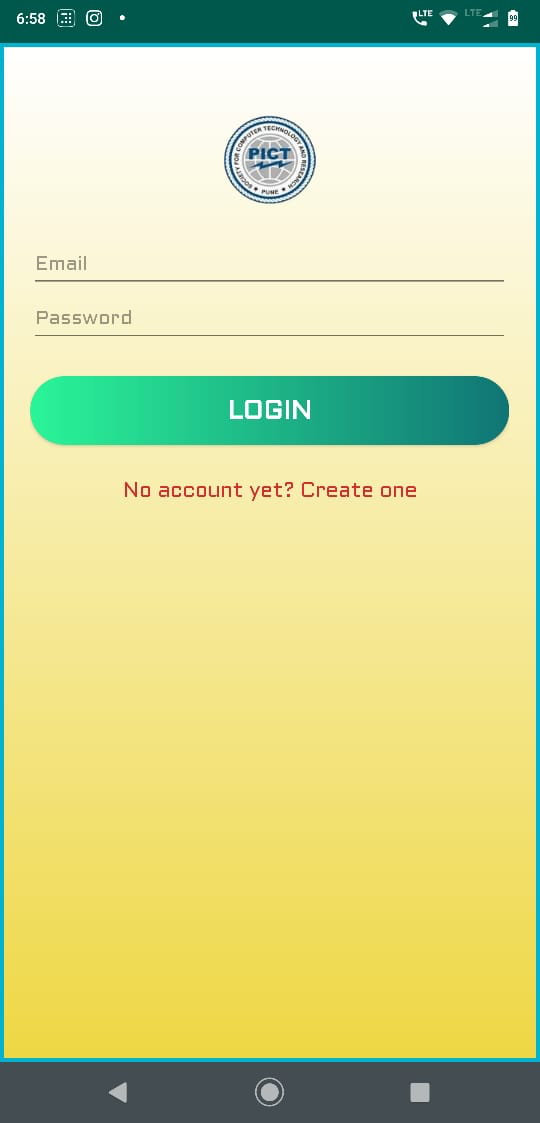
Table no. 1

3.2 RESULTS (Screenshots )

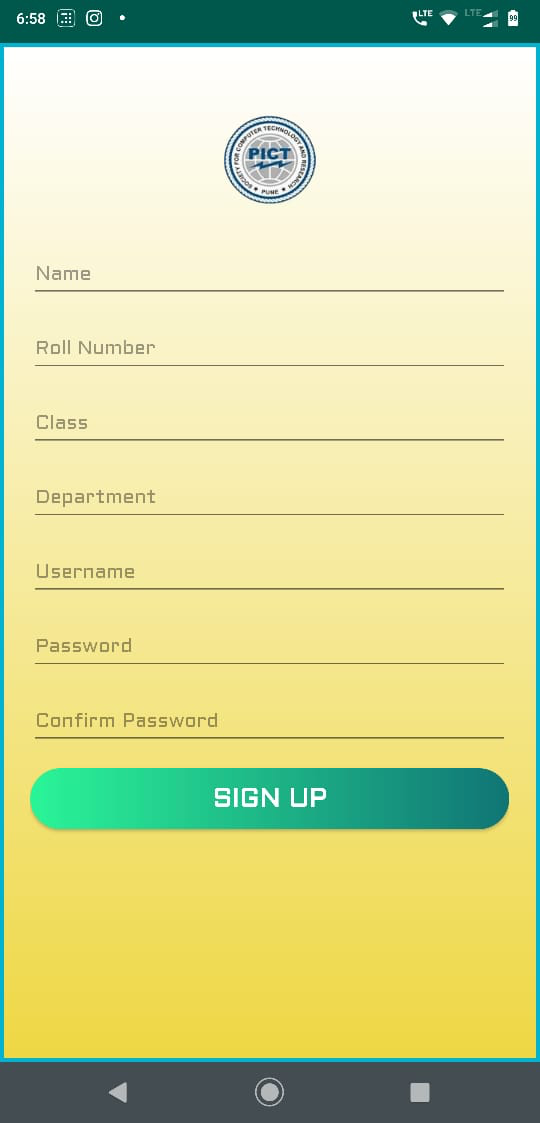
SPLASH SCREEN



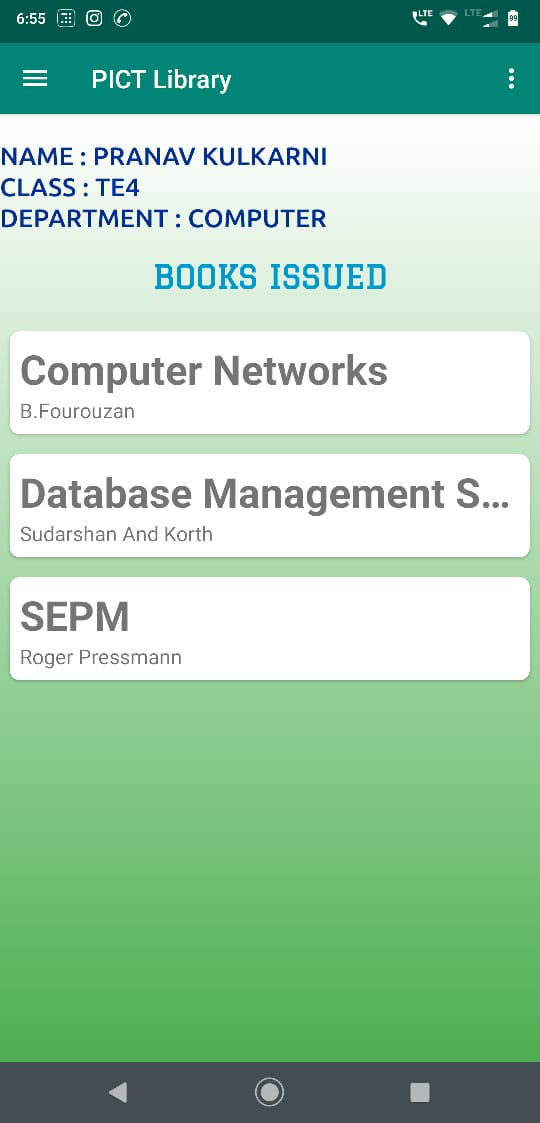
LOGIN PAGE



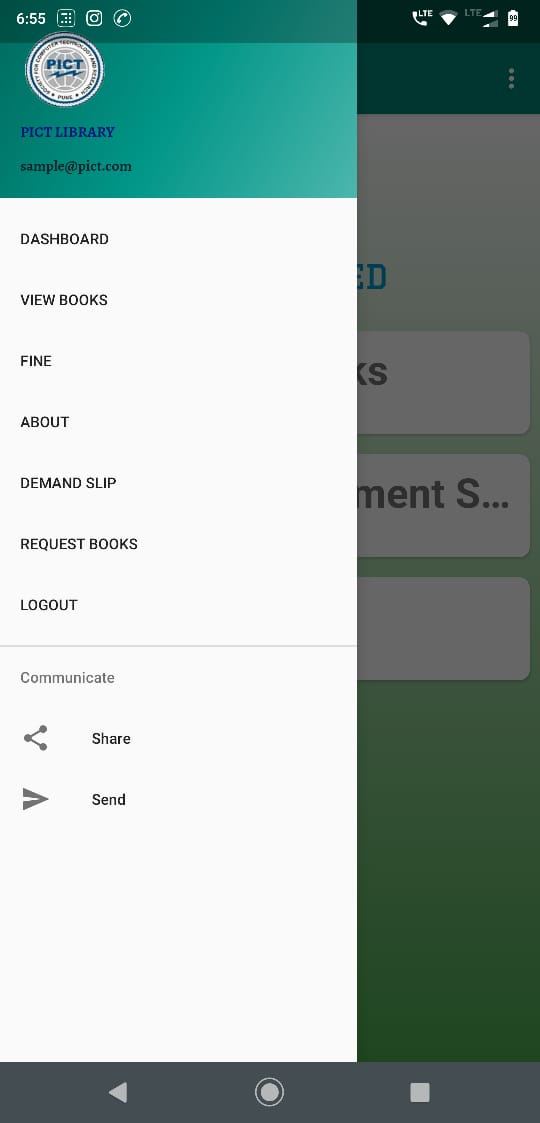
SIGN UP



DASHBOARD



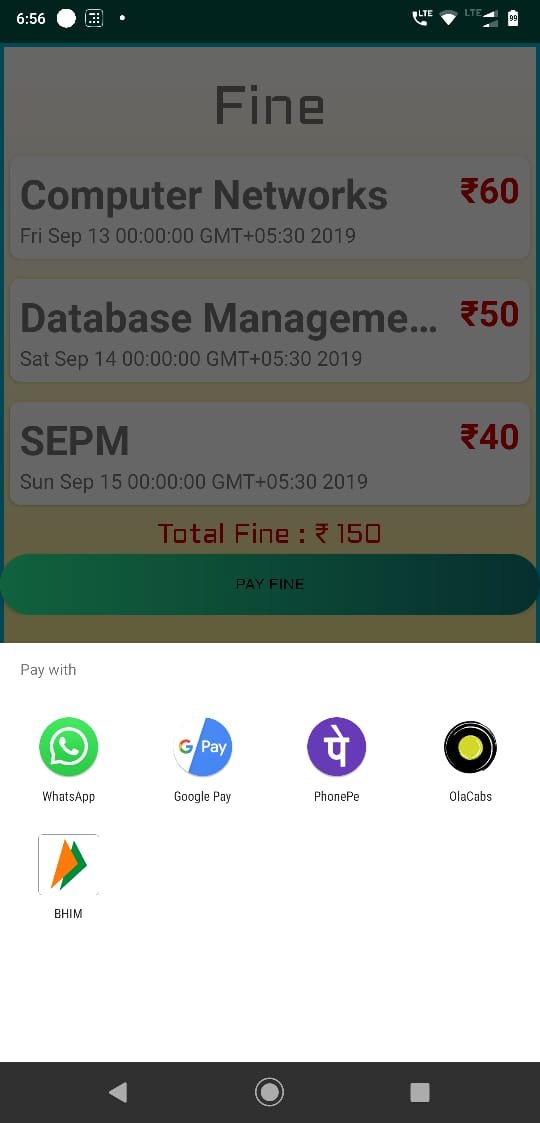
NAVIGATION DRAWER



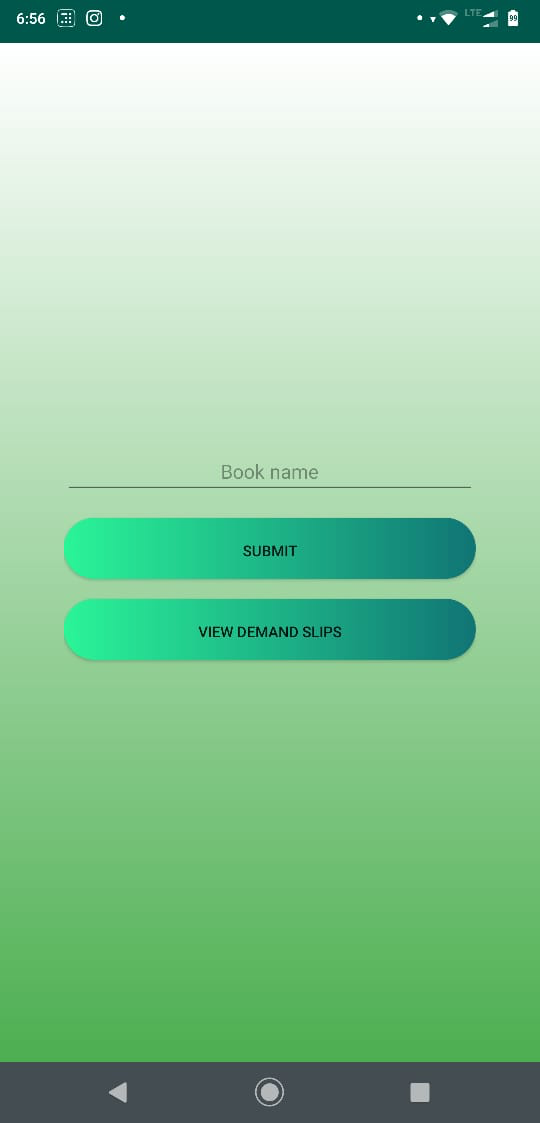
SEARCH BOOKS



FINE SYSTEM & UPI PAYMENT



DEMAND SLIPS



**CHAPTER 4**

CONCLUSION AND FUTURE ENHANCEMENTS

4.1 CONCLUSION

DIGILIB app was intended to simplify various library tasks for the users. With this application, user can perform numerous library tasks from his phone itself. This application will make the process of viewing, demanding, requesting and payment related to the books digital and very simplified. The GUI provided is so simple that any novice can also learn to use it.

This application will also be useful to library workers as the number of students visiting the library will only be limited to students issuing or returning books. The fine can be paid online and demand slips need not be physical , hence, this app will encourage eco-friendly paperless transactions.

Developing DIGILIB was a very enjoyable and learnable experience for our team. We had the privilege of going through the entire SDLC right from the requirement-gathering phase. We plan to further work on it and overcome its limitations and introduce additional features in the future.

4.2 FUTURE ENHANCEMENTS

* Admin level interface in the app.
* Optimized search algorithm for the app using recommender system.
* Barcode scanner system: User sends request to the librarian, the librarian accepts the request of the book and scan the barcode thereby, reserving the book for the user.
* Introduction of an e-book section for users.

5. REFERENCES

[1] Youtube. “ *FirebaseUI - Firestore + RecyclerView (FirestoreRecyclerAdapter) - Android Studio Tutoria*l”.

[Online] Available: <https://www.youtube.com/playlist?list=PLrnPJCHvNZuAXdWxOzsN5rgG2M4uJ8bH1>

[2] Youtube. “ *Custom Buttons Design - Android Studio Tutoria*l”.

[Online] Available: <https://www.youtube.com/watch?v=nlPtfncjOWA>

[3] Cloud Firestore Documentation [Online] Available: <https://cloud.google.com/firestore/docs/>

[4] Android Documentation [Online] Available: <https://developer.android.com/docs>

[5] John Horton, *Android Programming for Beginners* ,Packt Publishing Ltd.

[6] Lucas Moyer, *“Importing JSON into Firestore”* [Online] Available: <https://medium.com/lucas-moyer/how-to-import-json-data-into-firestore-2b370486b622>