Tribhuvan University

Faculty of Humanities and Social Sciences

E-library

A PROJECT REPORT

Submitted to

Department of Computer Application

Kantipur College of Management and Information Technology

In partial fulfillment of the requirements for the Bachelors in Computer Application

Submitted by

Prajwal Dahal [38802006]

November, 2023

Under the Supervision of

Mr. Sudip Khadka

Tribhuvan University

Faculty of Humanities and Social Sciences

Kantipur College of Management and Information Technology

Supervisor's Recommendation

We hereby recommend that this project prepared under my supervision by **Prajwal Dahal** entitled "**E-library.**" in partial fulfillment of the requirements for the degree of Bachelor of Computer Application is recommended for the final evaluation.

_____

Mr. Sudip Khadka
SUPERVISOR
Kantipur College of Management and Information Technology
Department of BCA

Tribhuvan University

Faculty of Humanities and Social Sciences

Kantipur College of Management and Information Technology

LETTER OF APPROVAL

This is to certify that this project prepared by **Prajwal Dahal** entitled "E-library" in partial fulfillment of the requirements for the degree of Bachelor in Computer Application has been evaluated. In our opinion, it is satisfactory in the scope and quality as a project for the required degree.

Mr.
External Examiner
Tribhuvan University

_____ Signature

_____ Date

Mr. Amit Pandey
HOD/Coordinator
KCMIT

_____ Signature

_____ Date

Mr. Sudip Khadka
Supervisor
KCMIT

_____ Signature

_____ Date

Acknowledgement

This project is prepared in the partial fulfillment of the requirement for the degree of Bachelor in Computer Application (BCA). The satisfaction and success of completion of this task would be incomplete without heartfelt thanks to people whose constant guidance, support and encouragement made this work successful. On doing this undergraduate project we have been fortunate to have help, support and encouragement from many people we would like to acknowledge them for their cooperation.

Our first thanks goes to Tribhuvan University for designing such a worthy syllabus and making us do this project. Our next batch of thanks goes to the faculty of Management of KCMIT without whose help our project would have been impossible. This list includes Principal of KCMIT, **Mrs. Lalita Chand** Our very sincere and heartfelt thanks go to **Mr. Sudip Khadka** our project supervisors who constantly guided us through the project time period. Without his guidance, our project would have been impossible. Last but not the least we want to thank every direct and indirect hands that were involved in completion of this project.

This project has been a wonderful experience where we have learnt and experienced many beneficial things.

Prajwal Dahal [38802006]

Abstract

This project focuses on the creation of the E-Library App, a user-friendly mobile application designed to provide seamless access to a wide array of books for a specified duration upon payment. The main goal is to provide a user-friendly, adaptable platform that allows book lovers to simply rent books for a small charge, meeting the changing reading tastes of contemporary readers.

Users will have an easy time navigating the E-Library app because to its simple and intuitive UI. Through a safe and user-friendly design, the main features enable customers to peruse a wide range of books, choose the titles they want, and rent them for a predetermined amount of time.

The focus of this project is to use mobile technology's convenience to improve the accessibility of literature. With agile development approaches, the app seeks to offer a dependable and sturdy solution for book lovers looking for a novel approach to appreciate books. Every user will have a secure and customized experience thanks to the integration of safe payment methods and individual user profiles.

In conclusion, the E-Library app emerges as a valuable tool for readers who seek a convenient and cost-effective way to access a vast collection of books. By merging technology with literature, this app endeavors to redefine the reading experience, making it more adaptable and accessible in the digital age.

Table of Contents

List of Figures

List of Tables

List of Abbreviation

API	Application Programming Interface
SQL	Structural Query Language

Chapter 1: Introduction

1.1: Introduction

Amidst the continuous technological transformation, people's interactions with information and entertainment are changing dramatically. Acknowledging this revolutionary shift, our project presents the E-Library App, a cutting-edge mobile application that has the potential to completely alter the literary interaction environment. This app is the perfect example of how innovation and tradition can co-exist when it comes to combining cutting-edge technology with the time-tested method of book rentals.

The main goal is to design a user-friendly environment where book lovers can quickly browse, choose, and check out books for a certain amount of time—all from the comfort of their smartphones. This project embraces transaction security and customization of the reading experience, going beyond simple accessibility. Through the implementation of a minimal cost structure, the E-Library app prioritizes democratizing access to a vast literary reservoir while also catering to the financial needs of its users.

1.2: Problem Statement

- Physical libraries are bound by space, opening hours, and geographic constraints, limiting access to resources for those with time restrictions or mobility challenges.
- Manual systems for tracking book availability, cataloging, and circulation can lead to errors, delays, and difficulty in identifying and retrieving desired books.
- Manual book tracking, late fees, and inconvenient return methods can create a hassle for users.
- Traditional services often lack features for users to create reading lists or track progress.

1.3: Objective

The main objectives of E-library app are given below:

- Make books easily accessible to a wider audience, regardless of location, time constraints, or physical limitations.

- Offer competitive pricing models and rental options, making reading affordable for a wider range of users.
- Offer a seamless and hassle-free experience for renting and returning books through the app.
- Design a platform that is easy to navigate and use, even for users with limited tech experience.

1.4: Scope

- Inclusion of a diverse range of educational materials, including e-books, research papers and articles.
- Implement google authentication mechanisms for secure access.
- Implement a secure and user-friendly book renting mechanism with secure payment gateway integration
- Include collaborative features such as book clubs or reading groups.

1.5: Development Methodology

The E-Library project adopts an agile methodology, emphasizing adaptability and iterative progress. This approach allows for flexibility in responding to evolving requirements throughout the development process. In this solo development context, various roles, including project management, development, and quality assurance, are consolidated into a single participant. Regular self-assessment sessions serve as a mechanism for progress tracking and issue identification, fostering continuous improvement. Continuous integration practices are employed to ensure the seamless integration of code changes, and automated testing contributes to code stability. Feedback mechanisms, including personal reviews and reflections, facilitate ongoing adjustments and improvements. The Agile methodology, customized to the individual developer's context, prioritizes iterative development, adaptability, and a focus on delivering incremental value throughout the project's lifecycle.

1.6: Overview of Report

The report contains altogether have seven chapters. The first chapter “Introduction” contains whole information of the project “E-library”. It is sub-headed into five topics. The

chapter contains the introduction, problem statements, objectives, scope and overview of report of our project. Similarly, chapter two consists of the background study and literature review of the report. In addition, chapter three is all about the methods by which our project is being completed. It is sub headed in two topics: system analysis and system modeling of the project. Under system analysis functional and non-functional requirement are made, feasibility analysis in technical, operational and economic, schedule feasibility of the project is analyzed. Under methodology, Object modeling such as object and class diagram, Process of the project is designed. The chapter four consists of the details of the implementation and test cases of system. Chapter five includes limitation, future enhancement, conclusion and recommendation of our project. There can be some changes in future in this project to make more effective is included and also consists of overall summary of the project.

Chapter 2: Background Story and Literature Review

2.1: Background Story

The E-Library project addresses challenges faced by traditional libraries in the digital era, aiming to provide modern readers with an accessible and immersive virtual library experience. Confronting issues such as limited accessibility and outdated cataloging systems, the project seeks to bridge the gap between traditional and digital libraries. With a collective vision to preserve library essence while meeting contemporary reading preferences, the E-Library app is designed to redefine reading, offering users anytime, anywhere access to a diverse collection. This transformative mobile application focuses on both technological innovation and user-friendliness, envisioning the E-Library as a digital haven for knowledge enthusiasts and fostering a community passionate about literature.

Embarking on this journey, the project aimed to navigate the realms of technology, literature, and innovation. It recognized the importance of celebrating books in a digital space where the joy of reading transcends physical boundaries. The narrative encapsulates the project's ethos, acknowledging the cultural significance of libraries while embracing the forward-looking approach of digital solutions. With a focus on creating a seamless and enriching reading environment, the E-Library project represents a transformative endeavor in the realm of literature and knowledge dissemination in the digital era.

2.2: Literature Review

Before making this project, a brief study on old E-library System were made. Various paper regarding Book Rental System were studied.

'BookLand' [1] is an android application for renting a book. It focuses on basic operations like adding new members, new books, updating new information, searching books and members (authors) and facility to borrow and return books. It is a physical book rental services.

'Scribd' [2] developed by Trip Adler, Jared Friedman and Tikhon Bernstam is an e-book subscription services. It has over 170+ document from over the world. User can read articles from scribd by paying monthly subscription.

Chapter 3: System Analysis and Design

3.1: System Analysis

This system is designed with the series of processes starting with requirement analysis, design, implementation, testing and maintenance. During requirement analysis, all the functional and non-functional requirement are analyzed and system is developed according to the requirement then designing of the system is carried out. After the design process, coding and development part is started then after integrating the system there is testing of the system. If the testing is positive then system is implemented otherwise some maintenance is done and system come in operation.

3.1.1: Requirement Analysis

i. Functional Requirement:

These are the requirements that the end user specifically demands as basic facilities that the system should offer. It specifies 'What should the software system do?' All these functionalities need to be necessarily incorporated into the system. They are basically the requirements stated by the user which one can see directly in the final product.

For Admin:

- The system should allow the admin to login to the system.
- The system should allow the admin to add, update and remove book and file to the system.
- The system should allow the admin to view user profiles.
- The system should allow admin to view details of rented book such as username, book details and payment paid by user.

For User:

- The system should allow users to login to the system.
- The system should allow users to view and search available books.
- The system should allow users to rent book for specific time by paying some amount of money.
- The system should allow users to pay rent using payment gateway integration.
- The system should allow user to read a rented materials in app.

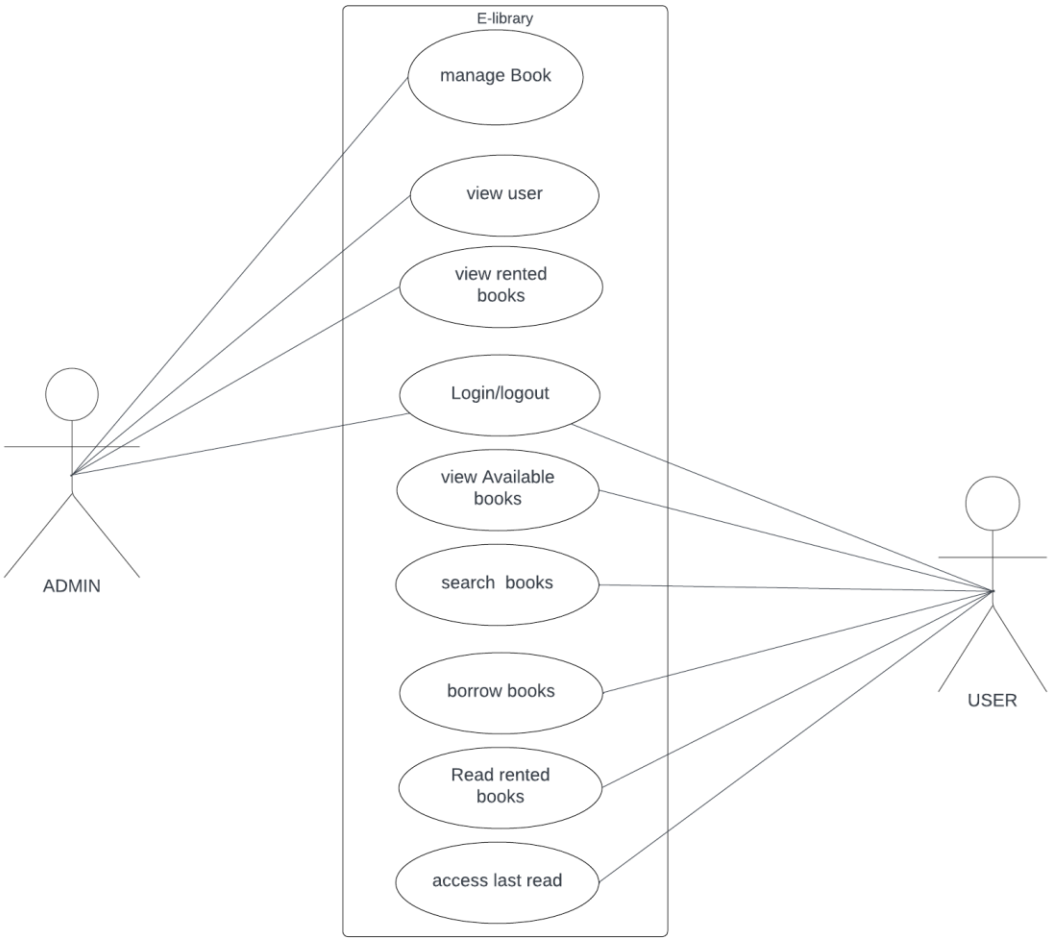


Figure 3.1.1.1: Use-case diagram

ii. Non-functional Requirement:

Non-Functional Requirement specifies the quality attribute of a software system. They judge the software system based on Responsiveness, Usability, Security, Portability and other non-functional standards that are critical to the success of the software system. This requirement describe how the functional requirement are implemented on the system.

Requirement Analysis produced the following non-functional requirement:

- System should be designed to have a clean and user-friendly interface, allowing for easy navigation and intuitive use.
- The system should be designed to be scalable, allowing for potential growth in the number of users and content on the platform.
- The system should have a reliable and fast API hosting infrastructure to ensure quick and seamless access to the platform.
- The system should have a secured payment gateway integration.

3.1.2: Feasibility Study

i. **Technical Feasibility:**

The system is technically feasible as the necessary hardware and software required for the development and implementation of the system is available. The project is built using native android app development (java), API and admin portal is built using PHP and hosted on a free webhosting services. The required libraries and dependencies for both technologies are readily available and easily accessible.

ii. Economic feasibility:

The project uses android studio, visual studio code and free webhosting provider for API hosting for developing this system which is free of cost. Also the project is the part of academic qualification for Bachelor in Computer Application - Tribhuvan University and there is no monetary factor involved. So, the project is economically feasible.

iii. Operational feasibility:

This study is carried out to check the acceptance of the system. The system is easy to operate with basic knowledge of computer and internet. Users can easily access the system as it is user-friendly with a well-designed user interface. So, the system is operationally feasible.

iv. **Schedule Feasibility:**

Schedule Feasibility is defined as the probability of a project to be completed within its scheduled time limits, by a planned due date. If a project has a high probability to be completed on-time, then its schedule feasibility is high. If we want to see the project completed before they can lose their utility, we need to give proper attention to controlling their schedule feasibility.

The final schedule of the project is given below:

Task	August				September				October			
	week 1	week 2	week 3	week 4	week 1	week 2	week 3	week 4	week 1	week 2	week 3	week 4
Planning												
Requirement analysis												
UI design												
Database design												
Report writing												
Implementation												
Testing and debugging												

Figure 3.1.2.1: Gantt chart of E-library

3.1.3: Object Modeling: Class Diagram

The object modeling phase is crucial in defining the structure and relationships within the E-Library system. The class diagram provides a visual representation of the key entities, their attributes, and associations. The following diagram represent a class diagram of E-library app:

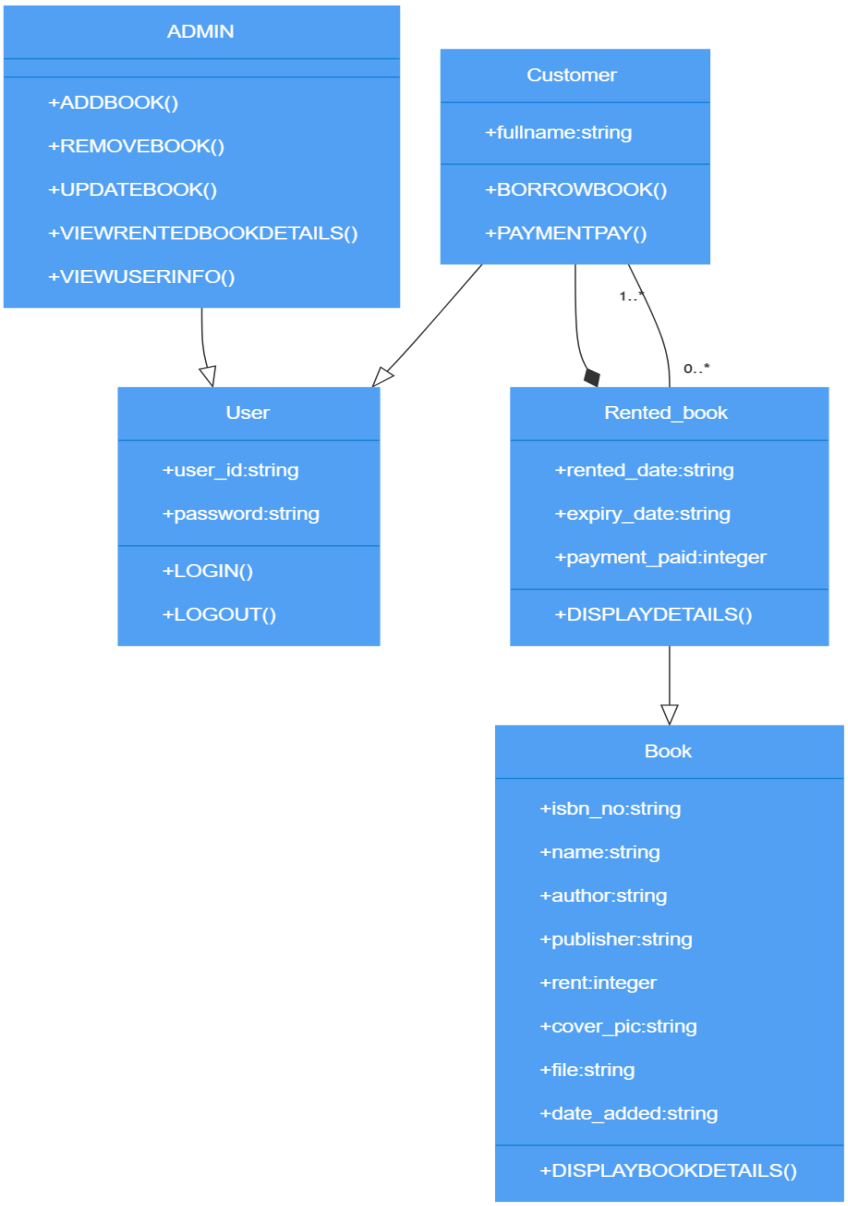


Figure 3.1.3.1: Class diagram of E-library

3.1.4: Dynamic Modeling: State and Sequence Diagram

In this section, dynamic aspects of the E-Library system are illustrated using state and sequence diagrams. The state diagram captures the lifecycle states of key entities, revealing their transitions over time. On the other hand, the sequence diagram visually depicts the interactions and message flows among system components during specific user scenarios. These dynamic models offer insights into the evolving states of entities and the sequence of events, enhancing the comprehension of system behavior.

For User:

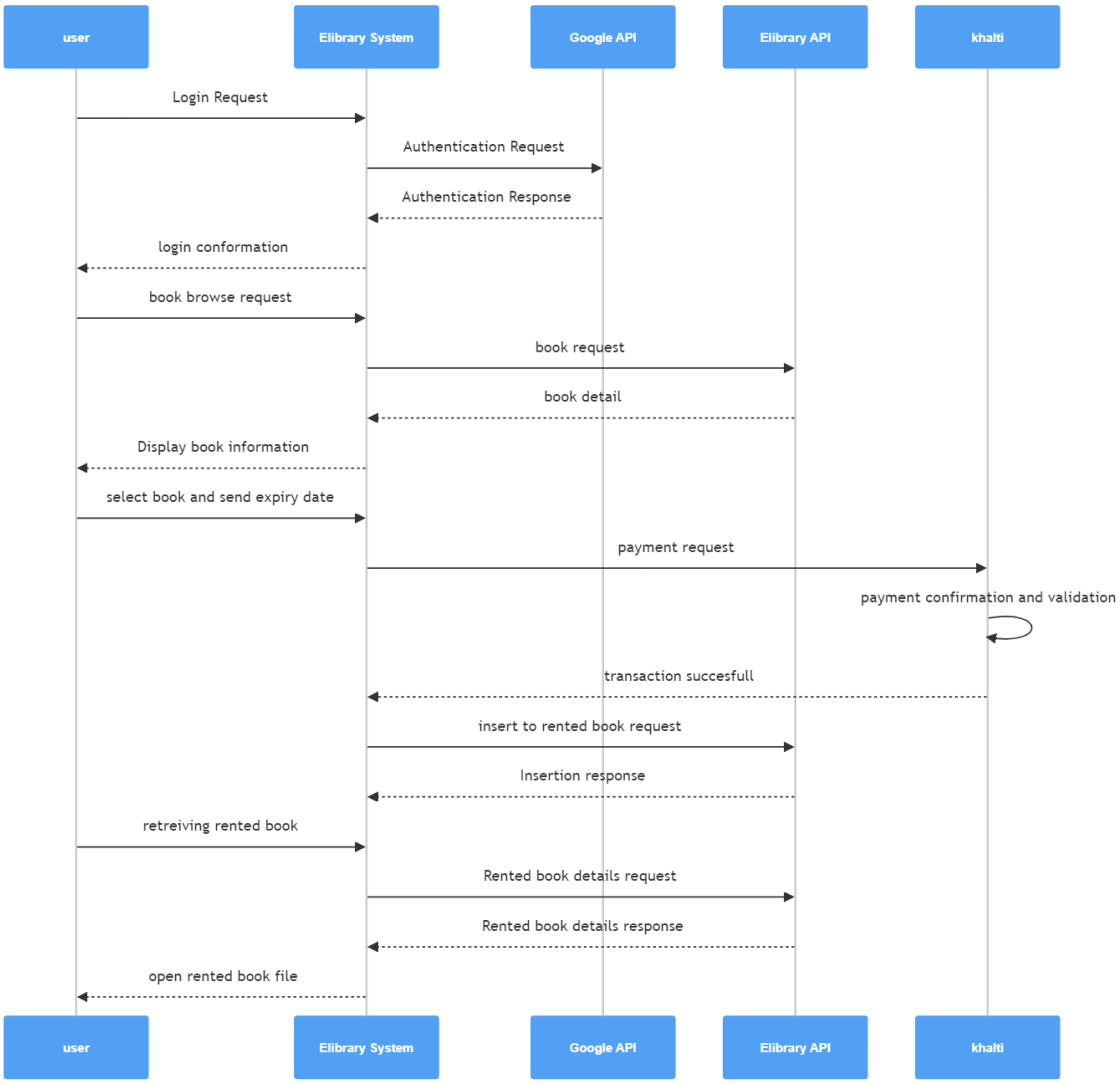


Figure 3.1.4.1: Sequence Diagram for User

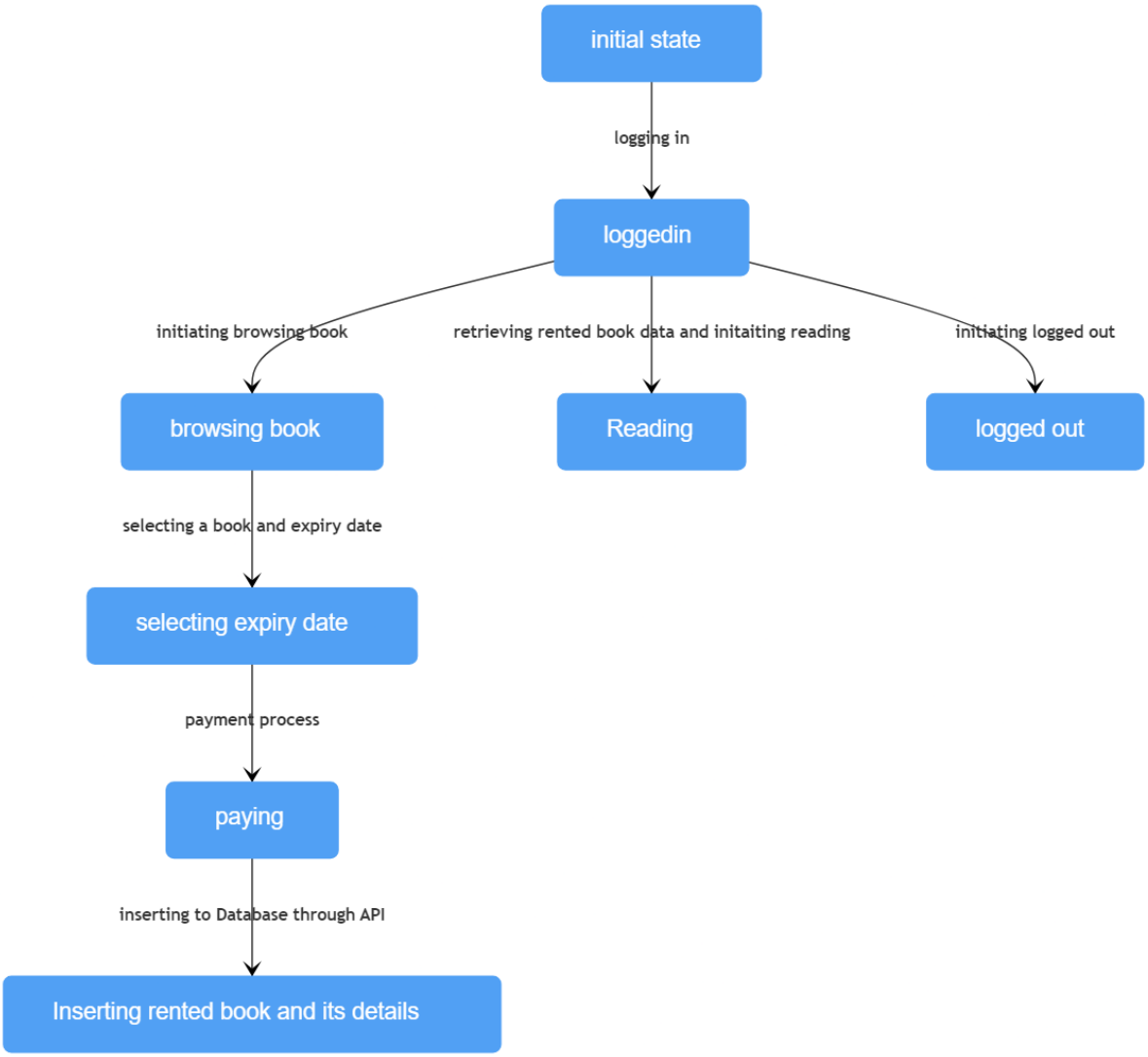


Figure 3.1.4.2: State Diagram for user

For Admin

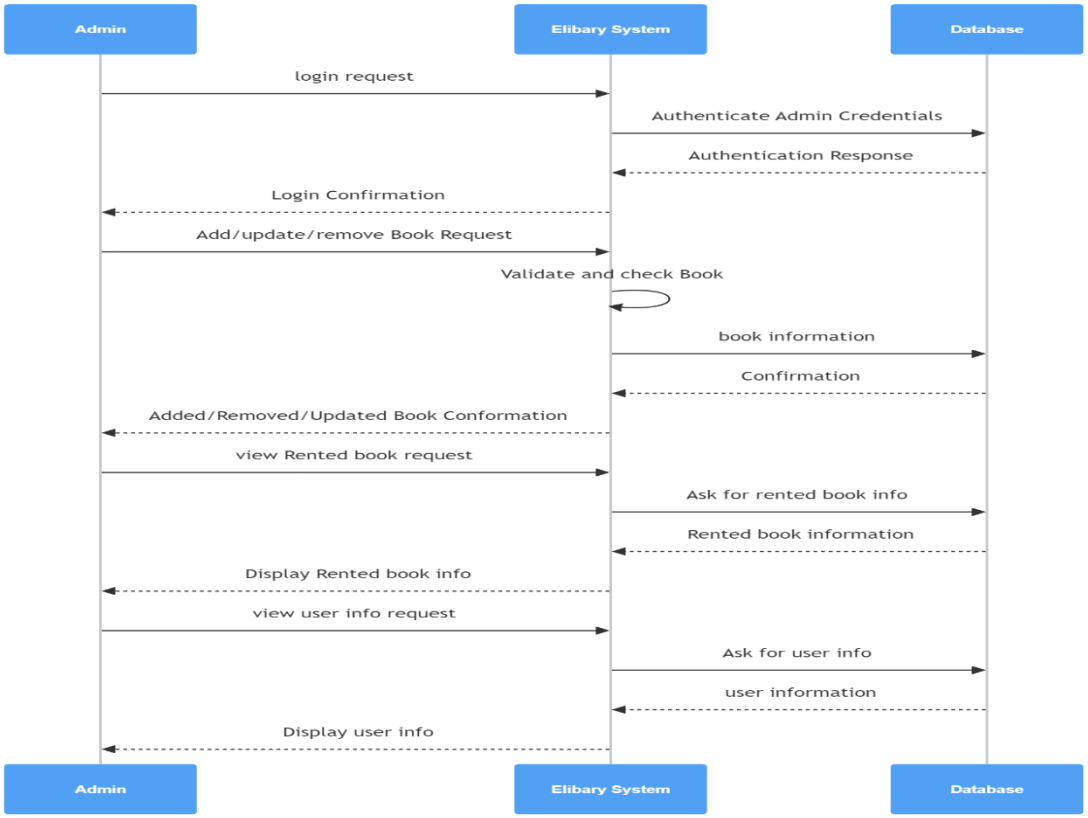


Figure 3.1.4.3: Sequence Diagram for Admin

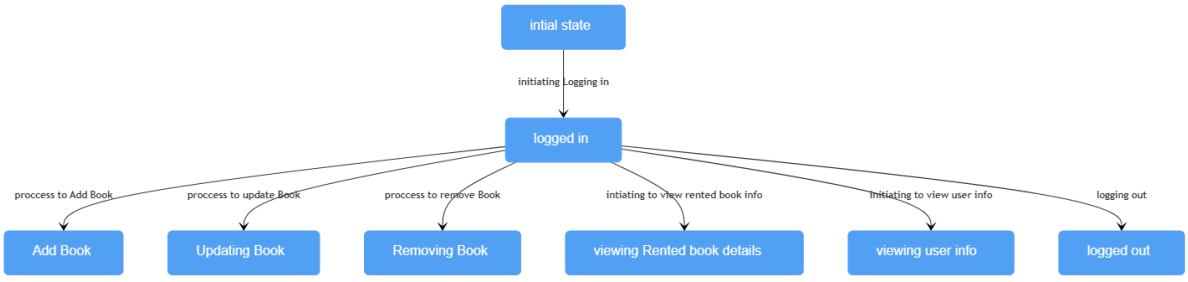


Figure 3.1.4.4: State Diagram for Admin

3.1.5: Process Modeling: Activity Diagram

Activity diagrams provide a visual representation of the workflow within a system, showcasing the activities performed and their sequence. In the context of the E-Library system, the following activity diagram illustrates the key processes and interactions involved in the system.

For Admin

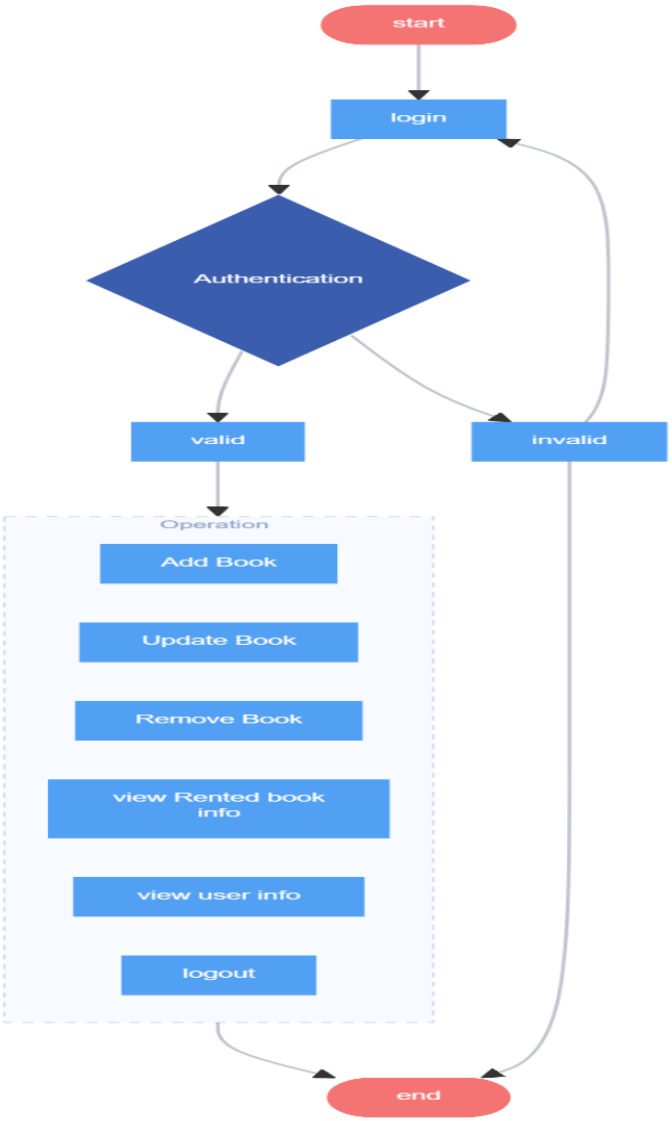


Figure 3.1.5.1: Activity Diagram for Admin

For User

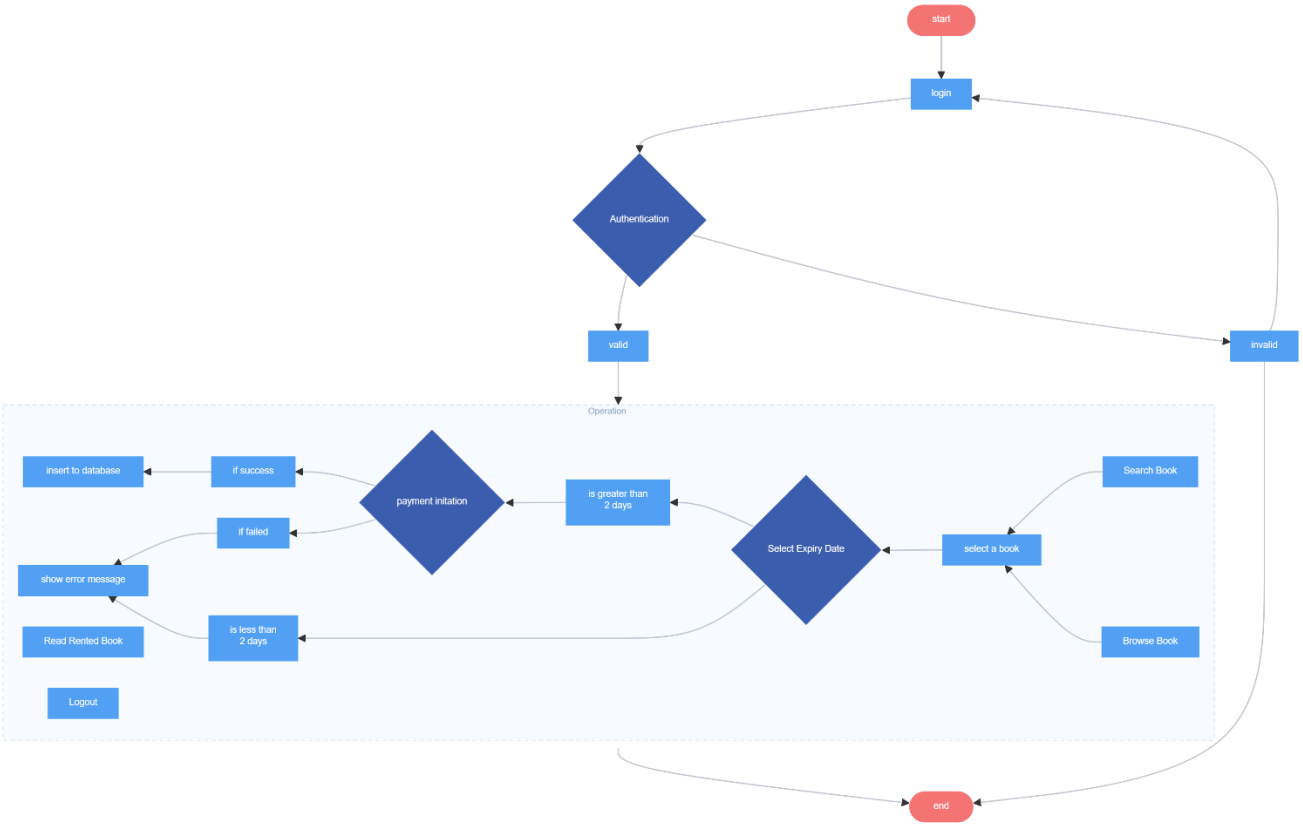


Figure 3.1.5.2: Activity Diagram for Admin

3.2: System Design

This section describes how actually system is implemented and work. To understand actual workflow of the system various model are shown in this section.

3.2.1: Component Diagram

To visualize the physical components of the system and their dependency relationship component diagram has been prepared.

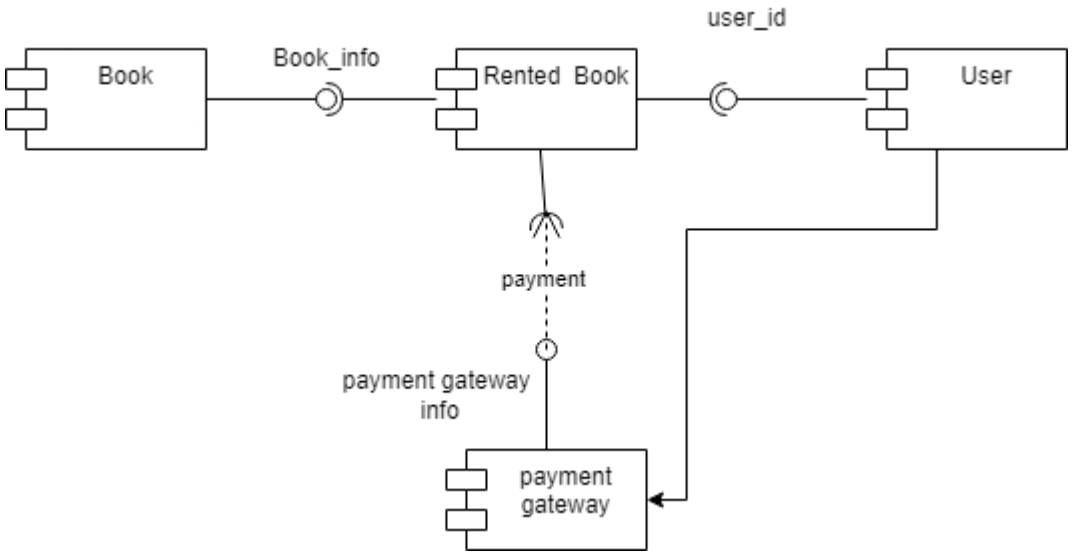


Figure 3.2.1.1: Component Diagram

3.2.2: Deployment Diagram

The deployment diagram has been made to show the execution architecture of the system which includes nodes such as hardware and software components, and the middleware for the system execution.

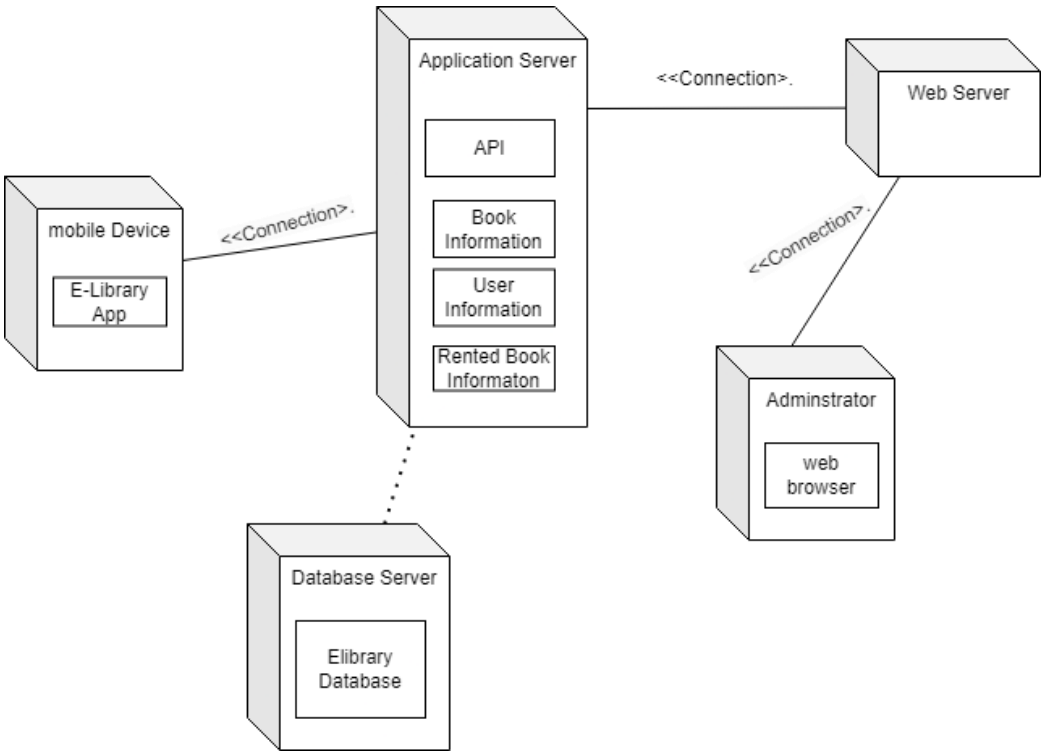


Figure 3.2.2.1: Activity Diagram for Admin

3.3: Algorithm Details

Price calculation algorithm is used to calculate the price to be paid for renting a book for specific period of time. The algorithm is given below:

Step 1: Start

Step 2: Input expiry date selected by user.

Step 3: Compute days difference between expiry date and today's date.

Step 4: if days difference is less than 2

If yes: show "invalid date" error

If no: calculate price=days difference*rent/30 and display it

Step 5: stop

Merge Sort algorithm invented by John von Neumann in 1945 is use for providing sorting feature in an app. it is also called divide and conquer algorithm. This algorithm is use for sorting a book by name, price, ISBN no, author, rented date, expiry date etc.

```
mergeSort (books):
    if books = null or books.size()<= 1:
        return books

    tempBooks = new ArrayList<T>()

    // Call recursive mergeSort function
    mergeSortRecursive (books, tempBooks, 0, size of books - 1)

    return books

mergeSortRecursive (books, tempBooks, low, high):
    if low < high:
        mid = (low + high) / 2

        // Recursively sort left and right halves
        mergeSortRecursive (books, tempBooks, low, mid)
        mergeSortRecursive (books, tempBooks, mid + 1, high)
```

```
        // Merge the sorted halves
        merge(books, tempBooks, low, mid, high)

merge(books, tempBooks, low, mid, high):
    i = low
    j = mid + 1
    k = low

    // Copy original books list to tempBooks
    tempBooks.addAll(books)

    // Merge the two halves
    while i <= mid and j <= high:
        switch (which):
            case NAME:
                if books[i].getName() <= books[j].getName():
                    tempBooks[k++] = books[i++]
                else:
                    tempBooks[k++] = books[j++]
                break

            case PRICE:
                rentLeft = parseInt(books[i].getRent())
                rentRight = parseInt(books[j].getRent())
                if rentLeft < rentRight:
                    tempBooks[k++] = books[i++]
                else:
                    tempBooks[k++] = books[j++]
                break

            case ISBNNO:
                if books[i].getIsbnno() <= books[j].getIsbnno():
                    tempBooks[k++] = books[i++]
```

```
            else:
                tempBooks[k++] = books[j++]
            break

        case EXPIRY:
            if books[0] instanceof RentedBook:
                rentedBook = (RentedBook) books[i]
                rentedBook2 = (RentedBook) books[j]
                if rentedBook.getExpiryDate() <= rentedBook2.getExpiryDate():
                    tempBooks[k++] = books[i++]
                else:
                    tempBooks[k++] = books[j++]
            break

        case RENTED:
            if books[0] instanceof RentedBook:
                rentedBook = (RentedBook) books[i]
                rentedBook2 = (RentedBook) books[j]
                if rentedBook.getRenteddate() <= rentedBook2.getRenteddate():
                    tempBooks[k++] = books[i++]
                else:
                    tempBooks[k++] = books[j++]
            break

// Copy remaining elements from left and right halves
while i <= mid:
    tempBooks[k++] = books[i++]

while j <= high:
    tempBooks[k++] = books[j++]

// Copy sorted elements from tempBooks back to books
books.addAll(tempBooks)
```

Chapter 4: Implementation and Testing

4.1: Implementation

4.1.1: Tools Used

Following are the tools and framework used for the accomplishment of this project:

- **Visual Studio code**

  Visual Studio Code (VS Code) is a free and open-source source code editor developed by Microsoft. It is a lightweight and highly customizable tool that supports a wide range of programming languages and extensions. It is use for developing admin portal using html, CSS, JavaScript and PHP.

- **Frontend Technologies**

  HTML, CSS and JavaScript is use for developing frontend of Admin portal. HTML is use for structuring admin portal, CSS is use for designing admin portal and JavaScript is use as client side scripting for form validation and improving user experience.

- **Backend Programming Language**

  PHP was chosen as the server-side scripting language for backend development of Admin Portal. It enabled the execution of dynamic server-side logic and seamless integration with databases for data retrieval and manipulation. PHP is also use for developing restful API for mobile app interaction.

- **MySQL**

  MySQL, a robust relational database management system, served as the database management system to store data. It efficiently handled structured data storage, retrieval, and management.

- **Android Studio**

  Android Studio is the official IDE for Android app development. It provides a rich set of features, including a powerful code editor, visual layout editor, emulator for testing, and a range of debugging tools. Android Studio supports both Java and Kotlin, the officially recommended languages for Android development. I have used java for developing this project.

- **Java Development Kit (JDK)**

The Java Development Kit (JDK) is a set of software development tools that allows developers to create Java applications and applets. It provides the tools and libraries

necessary for java. Android Studio requires JDK to compile Java code and build Android applications.

4.1.2: Implementation details of Module

Modules of this system are described as below:

User Module:

- **Google Authentication**

  The google authentication module's implementation for Google Login in our project involves configuring APIs and OAuth in the Google Developer Console. On the Android app client side, integration of the Google Sign-In library, initialization, and result handling ensure a user-friendly experience. Server-side validation of the ID token enables secure user management. Robust error handling, logout mechanisms, and security measures are implemented, complemented by comprehensive testing, logging, and documentation for seamless integration and future development.

- **Search module**

  The Search Module in our E-Library app enhances user exploration with an intuitive search feature directly within the client interface. This module enables users to seamlessly search and discover books using a responsive search bar. Efficient client-side algorithms manage metadata, providing accurate and relevant results. Rigorous testing ensures a smooth user experience, and comprehensive documentation supports ongoing improvements.

- **Rent Book Module**

  The Rent Book Module is use for renting a book. User select a book and its expiry date. Using price calculation algorithm amount is calculated based on expiry date. Payment Gateway Integration is use for paying the amount. After payment is successful, book metadata and user info is stored in database using API.

- **Read Book Module**

  The Read Book Module is use for implementing reading feature in an app. I have used pdfium library to integrate built in pdf reader. This module download a pdf file from the server, stored in app directory and load the pdf file in the app.

- **Last Read Module**

  The Last Read module is use for providing a progress of last read book. This module save book and its current page number accessed by the user in the database and based on this progress is calculated by formula (current page/total page *100) and display the progress.

- **Sorting Module**

  This module use for sorting of books. This module use merge sort algorithm for sorting.

Admin Module:

- **Mange Book**

  In this module, Admin can add, update, and delete book. The Admin starts the action by clicking on ‘Add book’ button and then enters info about book in form. After admin clicked on submit button, validation of data takes place. If validation is successful then only info is sent to server for saving in database. The admin can perform update and delete actions on their own comments.

- **View Rented book list**

  This module is use for displaying rented book info and its user info. This module retrieve data from database using SQL query and display data in tabular form.

- **Login Module**

  This module is use for authentication. Admin enter username and password in the login form. System check the credential against the data stored in database if credential matches, Admin is redirect to the dashboard.

4.2: Test cases

4.2.1: Unit Testing

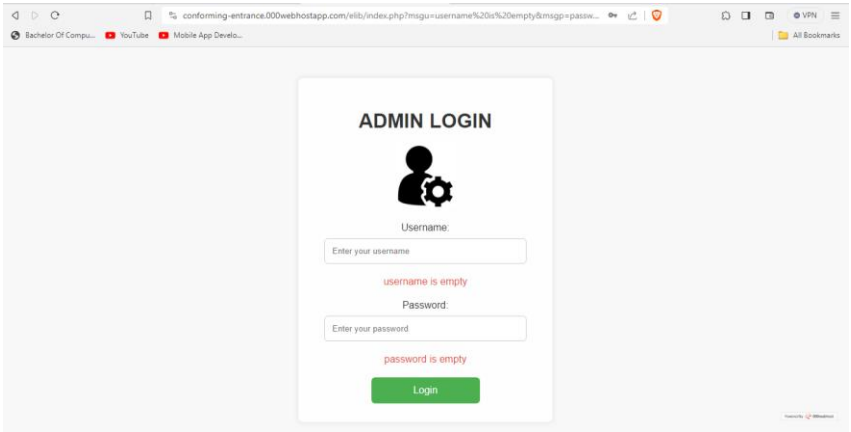
During unit testing, the system is modularized, and each module is independently tested. This includes validating input forms, ensuring accurate outputs for each module before progressing to the next, thus confirming the app's comprehensive functionality.

Table 4.2.1.1: Test Case for Admin Login

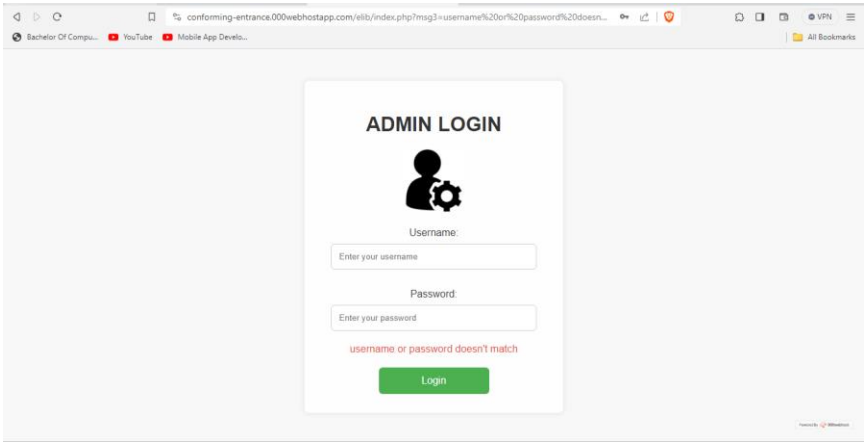
Test case id	1		
Test case description	Login		
Prerequisites	Enter the username and password. Click login		
Test scenario 1	User click login button without entering username and password		
Test data	Username: Password:		
Step	Expected output	Actual Result	Pass/fail
1.	Username is empty Password is empty	Username is empty Password is empty	Pass
Test scenario 2	Admin enter a wrong password or username		
Test data	Username: User Password: 123		
Step	Expected output	Actual Result	Pass/fail

1.	username or password doesn't match	username or password doesn't match	Pass
Test scenario 3	Admin enter valid username and password		
Test data	Username: admin Password: root@123		
Step	Expected output	Actual Result	Pass/fail
1.	Admin dashboard open	Admin dashboard open	Pass

Test scenario 1:



Test scenario 2:



Test scenerio 3:

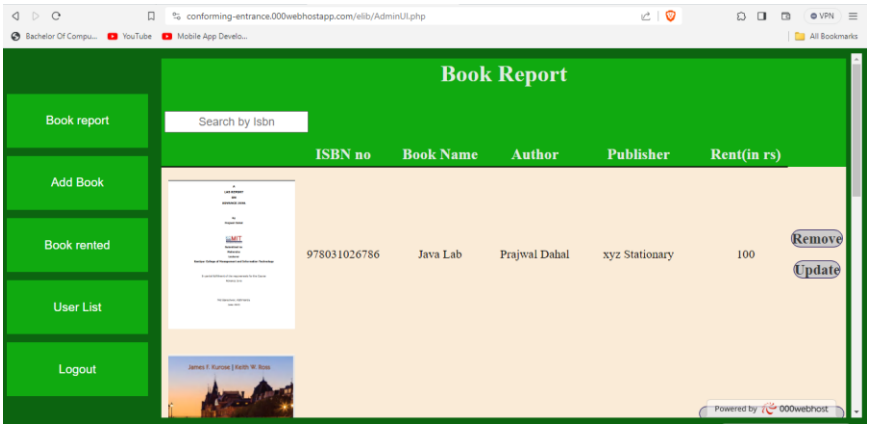
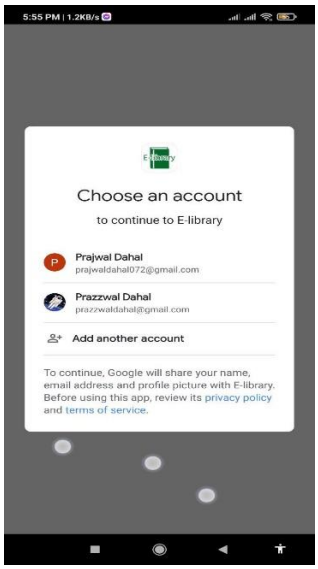


Table 4.2.1.2: Test Case for User Login

Test case id	2		
Test case description	Login		
Prerequisites	User has Gmail account		
Test scenario 1	User click on sign in button		
	Expected output	Actual Result	Pass/fail
	Show Gmail account added on devices	Show Gmail account added on devices	Pass
Test scenario 2	User select one Gmail accounts		
	Expected output	Actual Result	Pass/fail
	User enters App home Screen	User enter App home Screen	Pass
Test Scenario 3	User attempt to login without Internet Accessed		
	Expected output	Actual Result	Pass/fail
	Toast with text ” login successful” is displayed	Toast with text “login unsuccessful” is shown	Pass

Test scenario 1:



Test scenario 2:

Test case scenario 3:

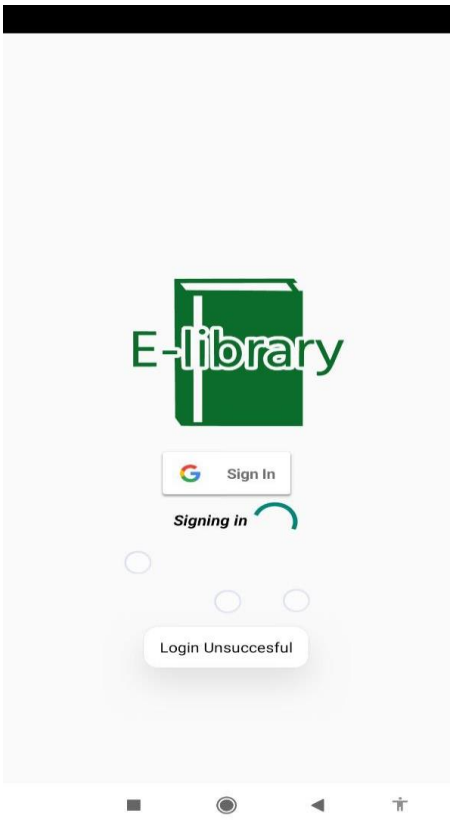


Table 4.2.1.3: Test case for payment Gateway Integration

Test Case ID	3		
Test Case Description	Test case for Payment Gateway Integration		
Prerequisites:	Public Test Id and android khalti Dependencies implemented		
Test Scenario	Click on khalti button		
	Expected Result	Actual Result	Pass/Fail
	Khalti Interface open	Khalti Interface open	Pass

Test scenario 1:

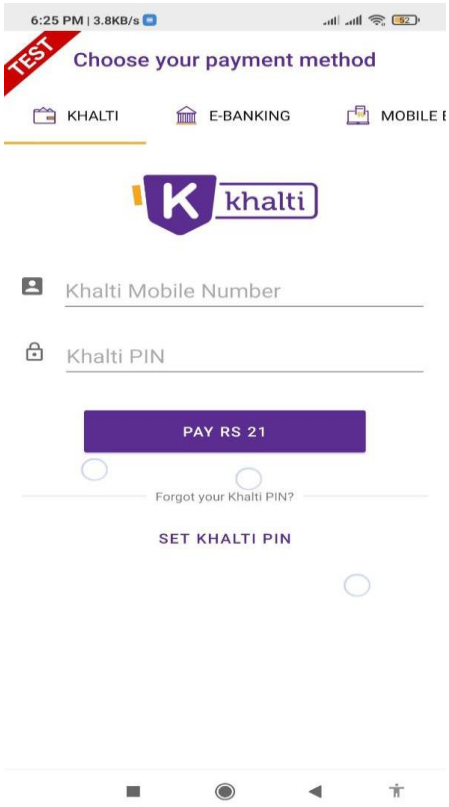
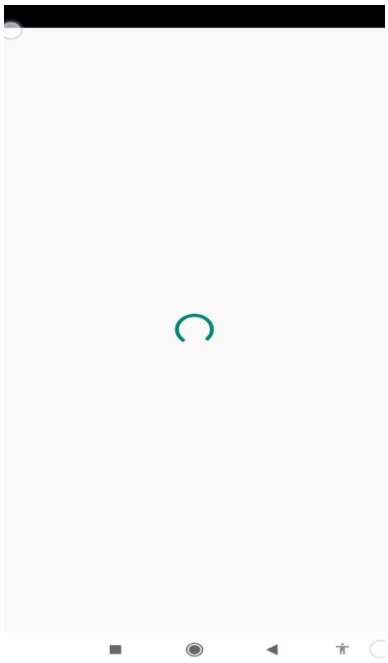


Table 4.2.1.4: Test case for opening Pdf File

Test case id	4		
Test case description	Check if rented book pdf file open		
Prerequisites	User has rented at least one book and has to click on read book button		
Test scenario 1	Book is accessing for the first time		
	Expected output	Actual Result	Pass/fail
	White blank screen with progress bar running	White blank screen with progress bar running	Pass
Test scenario 2	Book is already accessed		
	Expected output	Actual Result	Pass/fail
	Pdf file is open inside app	PDF file is open inside app	Pass

Test Scenario 1:



Test Scenario 2:

A
LAB REPORT
ON
COMPUTER NETWORKING

By
Prajwal Dahal
Exam Roll No: 388-51-22-00006

KGMIT
Knowledge Gateway

Submitted to:
Sudip Raj Khadka
Lecturer
Kantipur College of Management and Information Technology

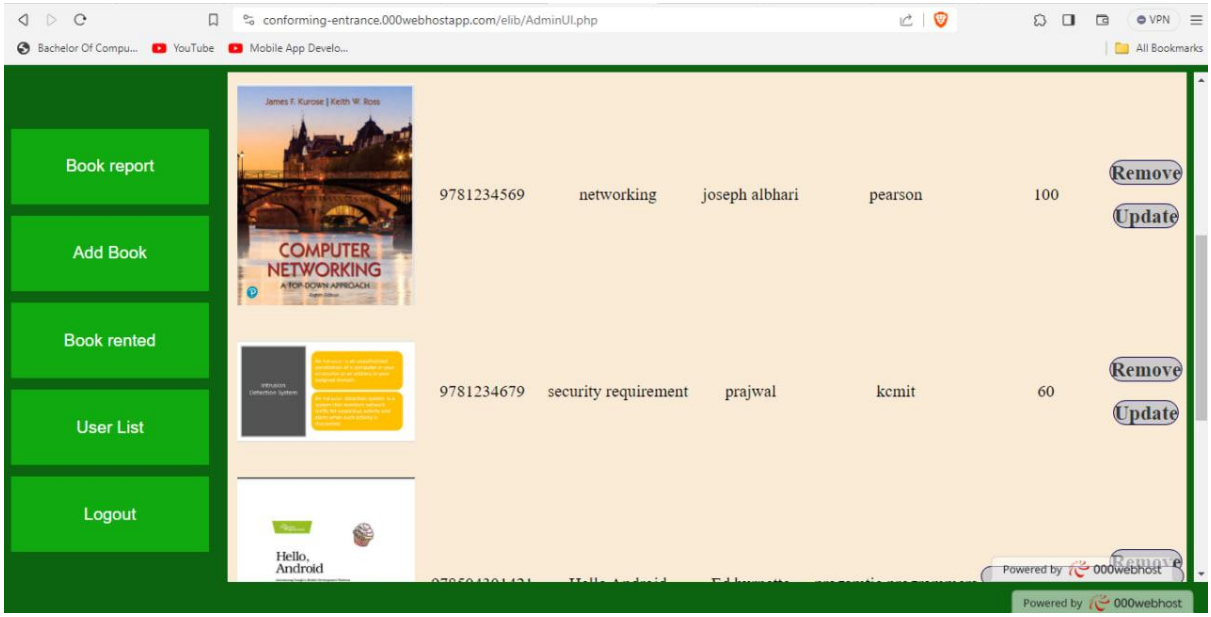
In partial fulfillment of the requirements for the Course
Computer Networking

Mid Baneshwor, Kathmandu
May 2023

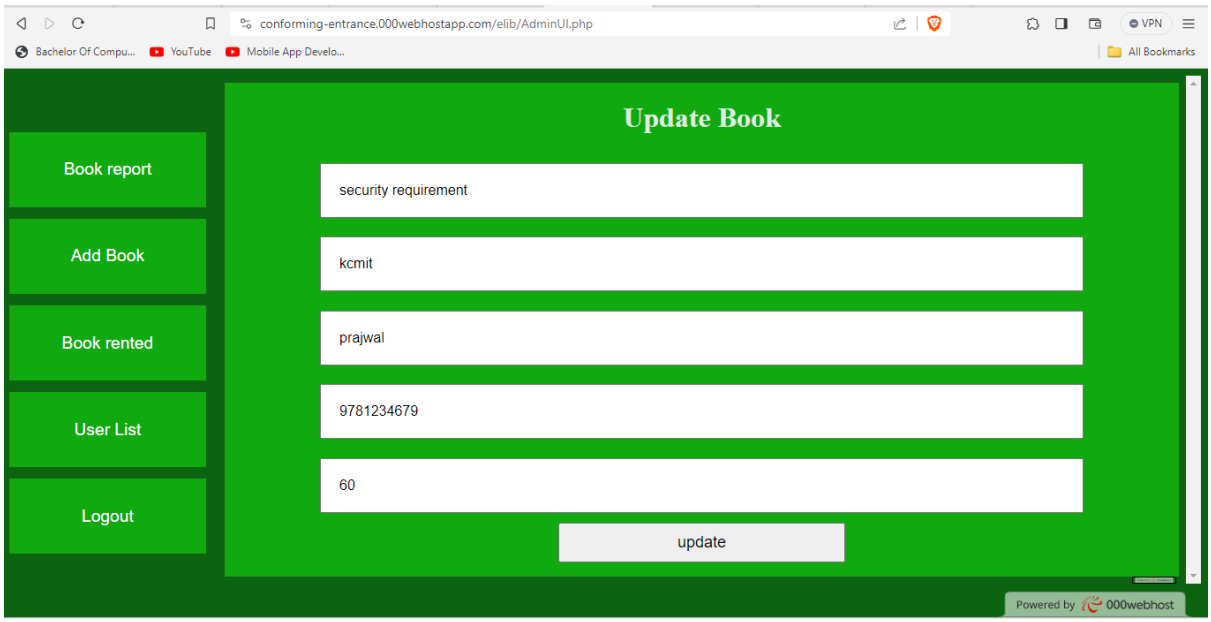
Table 4.2.1.5: Test case for add, update and delete Book

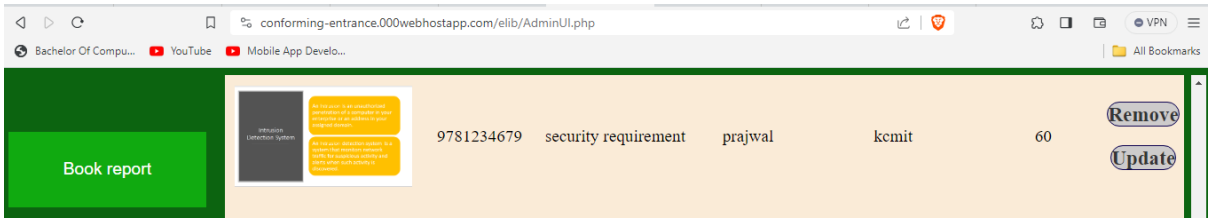
Test Case ID	4		
Test Case Description	Add, update and delete Room		
Prerequisites:	Admin is successfully logged in.		
Test Scenario	1. Enter all the fields required 2. Click on add/update button		
Test Data	**Add data** name: networking ISBN: 9781234569 Author: Joseph Albhari Publisher: pearson Upload cover pic: coverpic.png Upload file: networking.pdf **Update Book:** ISBN: 9781234679 Updated Rent :60 **Remove Book:** **ISBN:**978504301421		
	Expected Result	Actual Result	Pass/Fail
1.	Added Book displayed in book report	Added book displayed in book report	Pass
2.	Updated data show in book report	Updated data show in book report	Pass
3.	Display deleted alert dialog and remove book report	Display deleted alert dialog and remove book report	Pass

Add Book result:

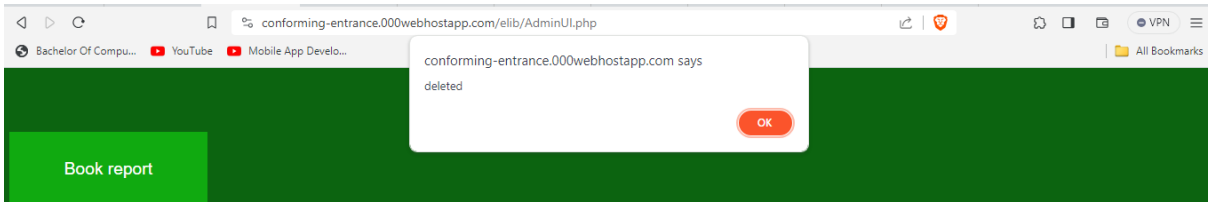


Update book result:





Remove Book Result:



4.2.2: System Testing

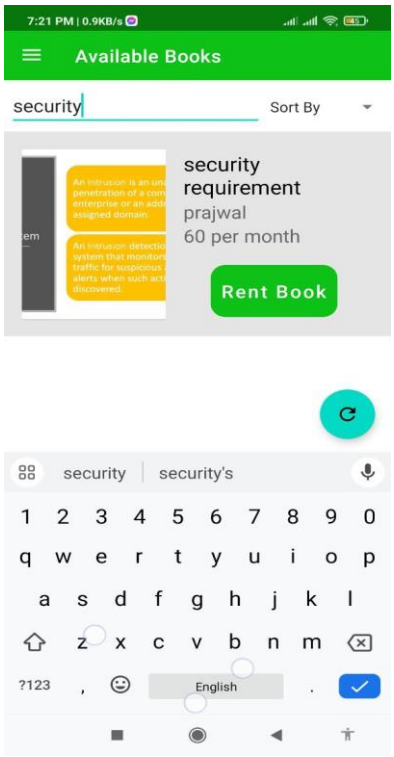
System testing is a comprehensive and end-to-end testing process that evaluates the entire integrated system to ensure it meets specified requirements. Here are key aspects and test cases to consider during system testing for an E-Library system:

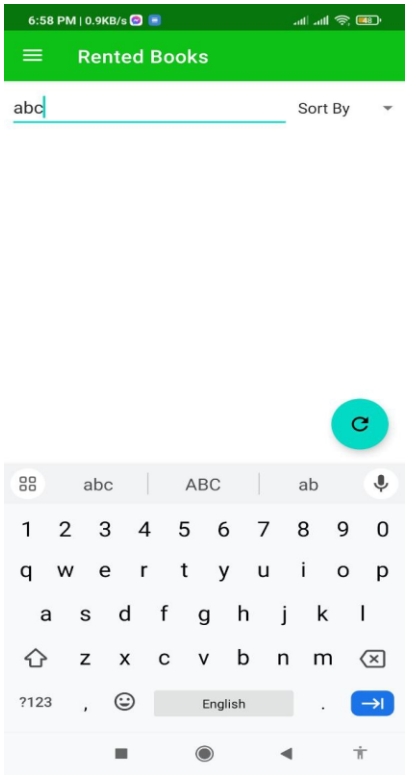
Table 4.2.2.1: Test Case for Book discovery and Rental

S.No	Test Scenario	Input	Expected Output	Actual Output	Test Result
1.	Search for a book with a valid query	Text: security requirement	Security Requirement book Displayed	Security Requirement book Displayed	Pass
2	Search for a book with invalid query	Text: abc	Nothing displayed	Nothing Displayed	Pass
3.	Rent a book with invalid Expiry date	Current date:2023-01-23 Expiry Date: 2023-1-22	Invalid date displayed	Invalid date displayed	Pass
4	Rent a book with valid Expiry date	Current date:2023-01-23 Expiry Date: 2024-01-25	Displayed pay with khalti	Displayed pay with khalti	Pass
5	Payment for renting book	User id=98******** Password=*****	Book rented successfully	Book rented successfully	Pass

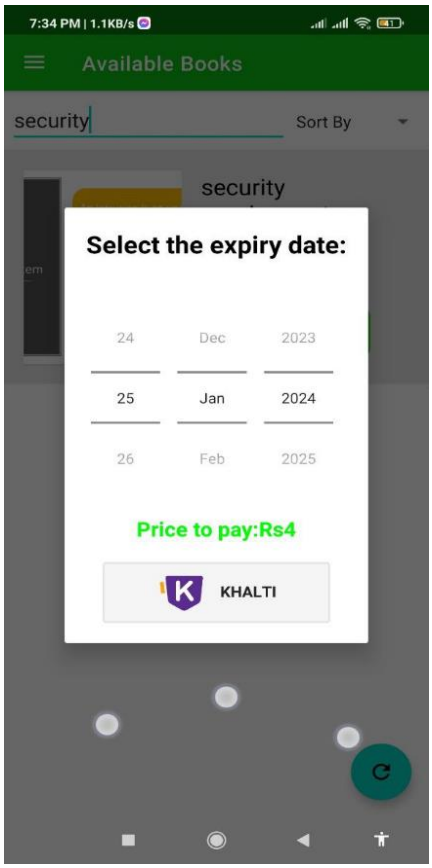
6	Viewing details of rented book	Selects hello android book	Accurate details (title, author) are displayed in dialog box	All details are accurately displayed	Pass

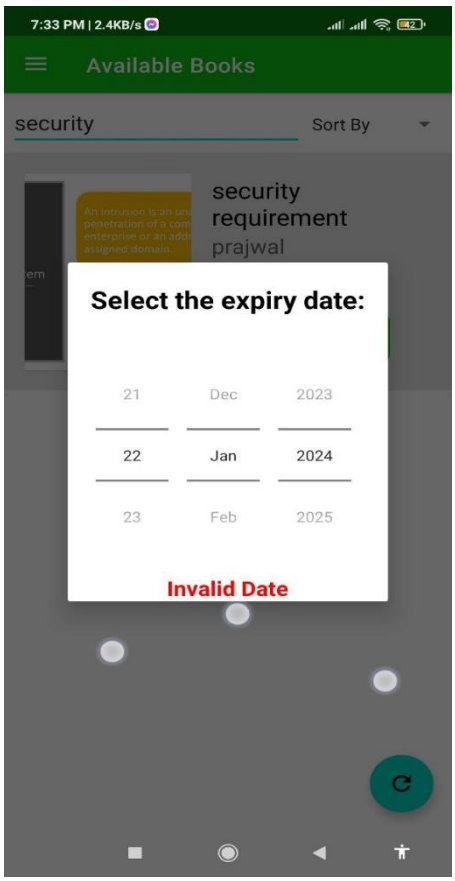
Test Result for Searching:





Test Result for renting a book:





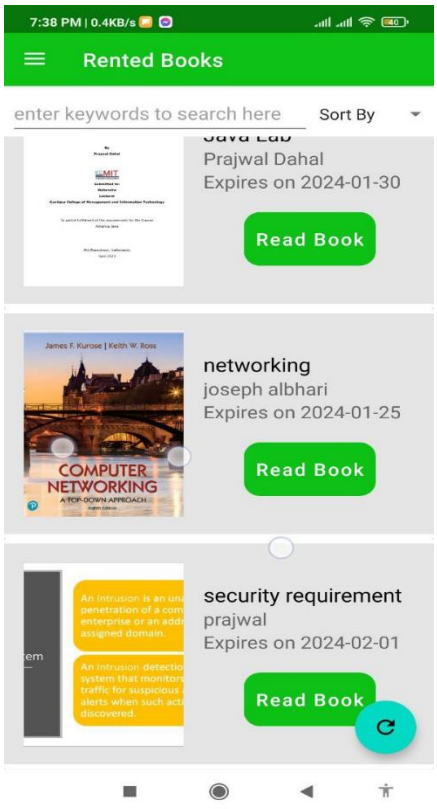
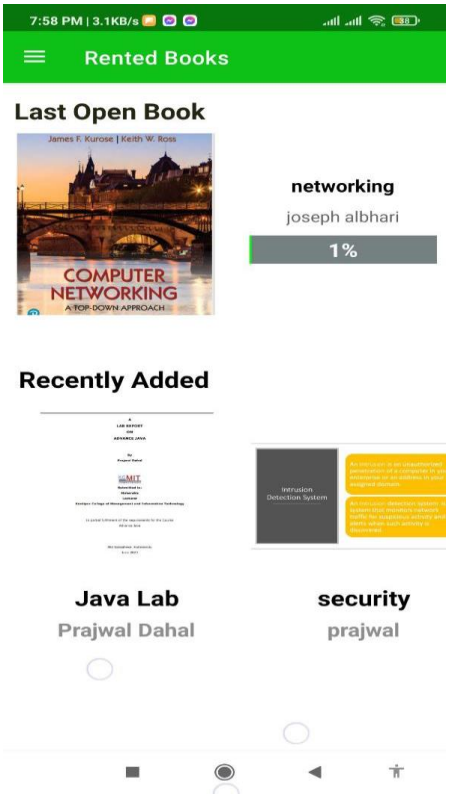
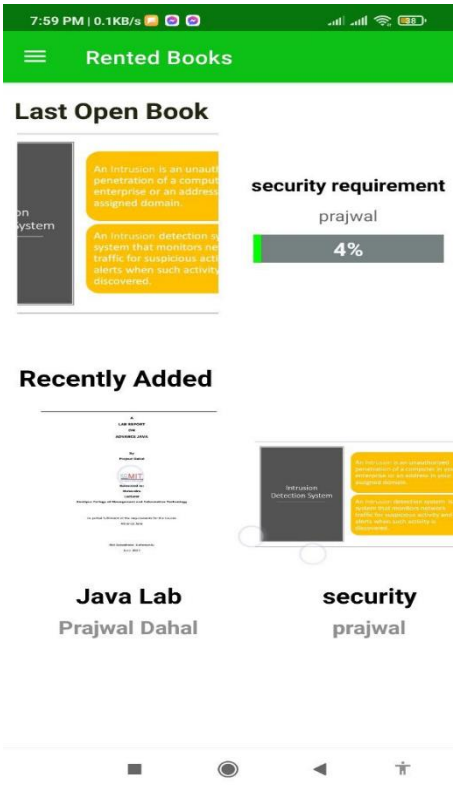

7:38 PM | 0.4KB/s
Rented Books
enter keywords to search here
Sort By
Prajwal Dahal
Expires on 2024-01-30
Read Book
James F. Kurose | Keith W. Ross
COMPUTER NETWORKING
A TOP-DOWN APPROACH
networking
joseph albhari
Expires on 2024-01-25
Read Book
security requirement
prajwal
Expires on 2024-02-01
Read Book

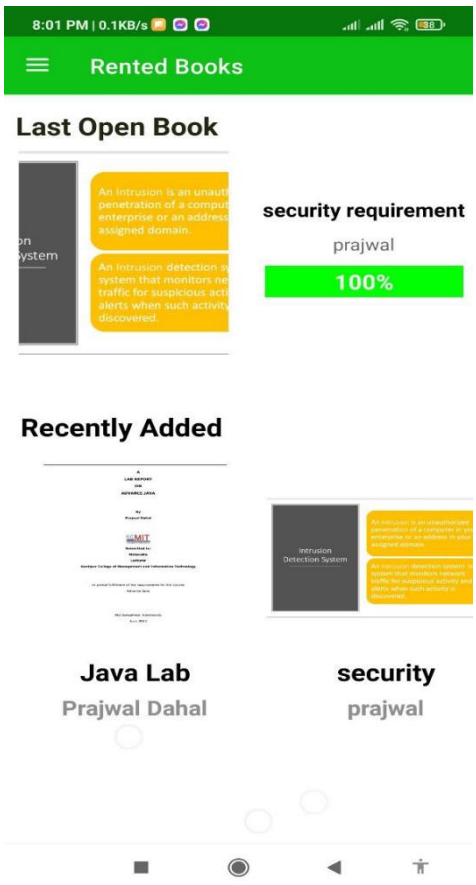
Table 4.2.2.2: Test Case for Book read and Last Read

S.No	Test Scenario	Input	Expected Output	Actual Output	Test Result
1	Currently logged in and last read is empty. Read a book you rented	Open rented screen and click read book button on networking and close it	networking book should be displayed in last read with progress	Networking book is displayed in last read	Pass
2	Currently Networking book is displayed in last read. open a security requirement book	Open rented screen and click read button on security requirement after file open close it	Security requirement should be displayed in last read with progress bar	Security requirement book is displayed	
3	Last Read progress	Total page=25 Current page=25	progress displayed=100%	progress displayed=100%	Pass

Test results:







Chapter 5: Conclusion and Recommendation

5.1: Conclusion

The E-Library App has successfully achieved its goal of providing a user-friendly platform for book rentals. The app's development involved overcoming challenges related to secure payment integration and efficient book retrieval. The user-centric design, featuring Google API authentication, enhances accessibility for a diverse audience. The project's success reflects the transformative potential of technology in modernizing traditional library services.

5.2: Lesson Learnt/Outcome

The development of the E-Library app has provided me with a wealth of knowledge and skills in various areas of software development. As the sole developer of this application, I have gained valuable experience in problem-solving, effective communication, and project management. The project also honed skills in mobile app development, web development, secure payment integration, and agile methodologies, emphasizing adaptability and user responsiveness. The experience deepened understanding in user-centric design, emphasizing the importance of creating seamless, intuitive interfaces. Overall, the project's outcomes contribute to a richer understanding of technology's role in reshaping user experiences and meeting evolving needs.

5.3: Future Recommendation

Here is what can be added in future to this E-library Project:

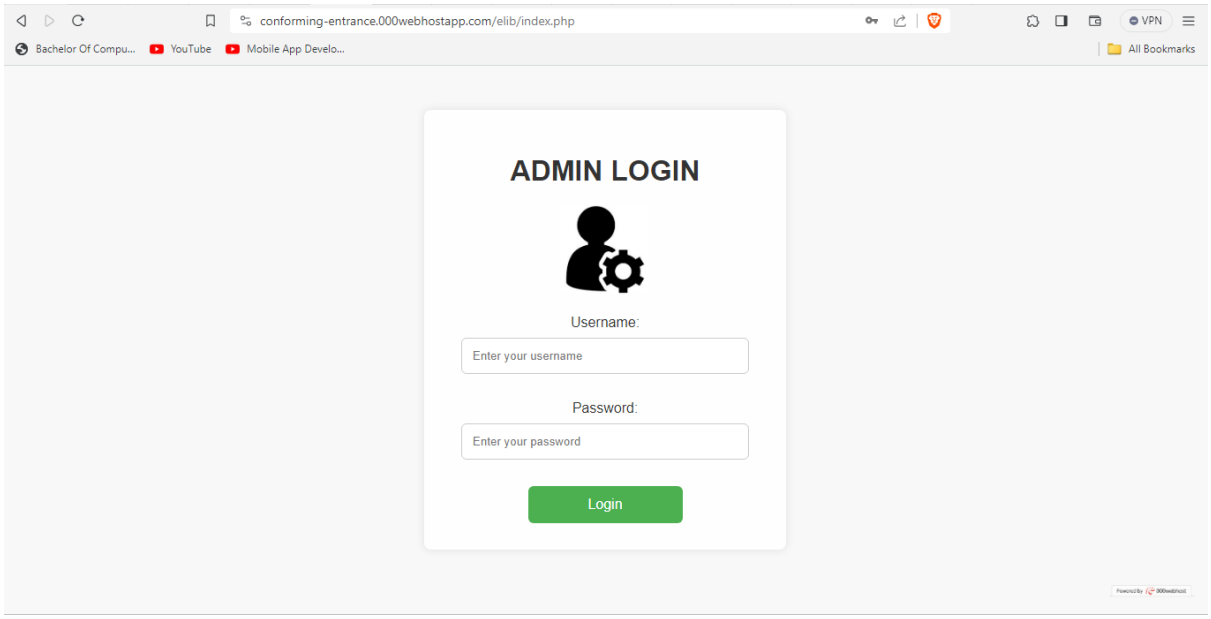
- Continuously diversify and expand the available book collection to cater to a broader range of user interests.
- Integrate recommendation algorithm to provide more personalized and accurate book suggestions based on user preferences and behaviors.
- Categorize the book according to field of study so that searching is easier and fast.
- Integrate a user-friendly review and rating system for books, allowing users to share their opinions and experiences.
- Continuously work on improving the app's accessibility features to ensure a seamless experience for users with diverse needs.

Bibliography

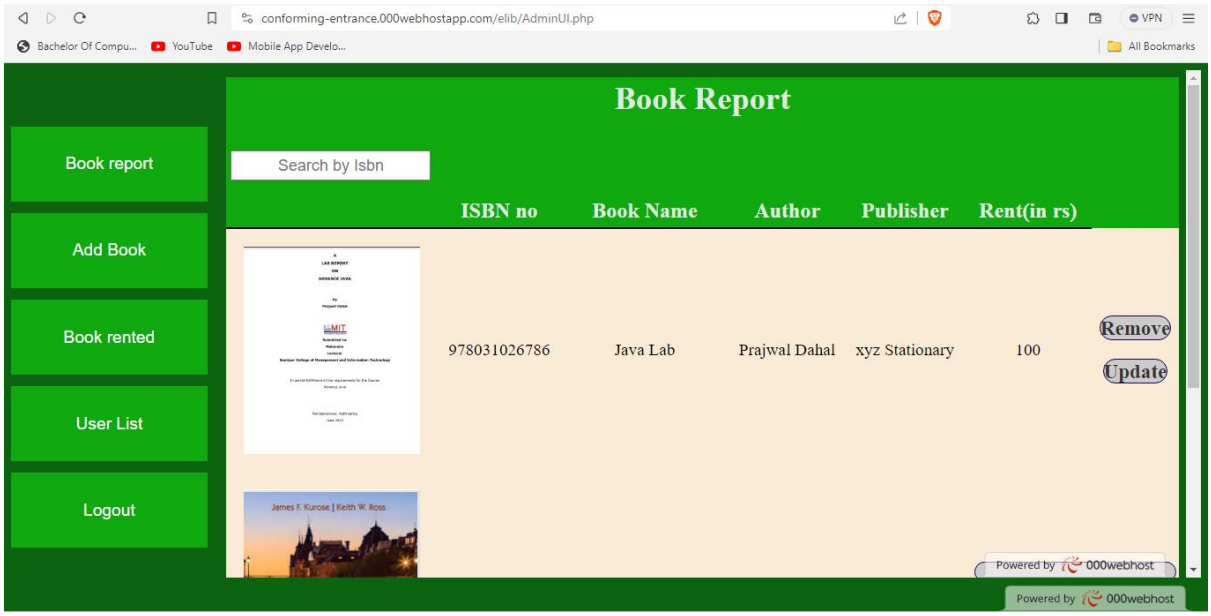
[1] A. S. M. S. R. S. M. Y. Dr. V. K. Manavalasundaram, "BOOKLAND – AN ANDROID APPLICATION FOR Book Rent," 2014. [Online]. Available: https://lemma-tijdschriften.com/gallery/goj-1643.pdf.

[2] A. Mezhova, "repositorio.ucp.pt," June 2016. [Online]. Available: https://repositorio.ucp.pt/bitstream/10400.14/20700/1/Case%20study%20Scribd.pdf.

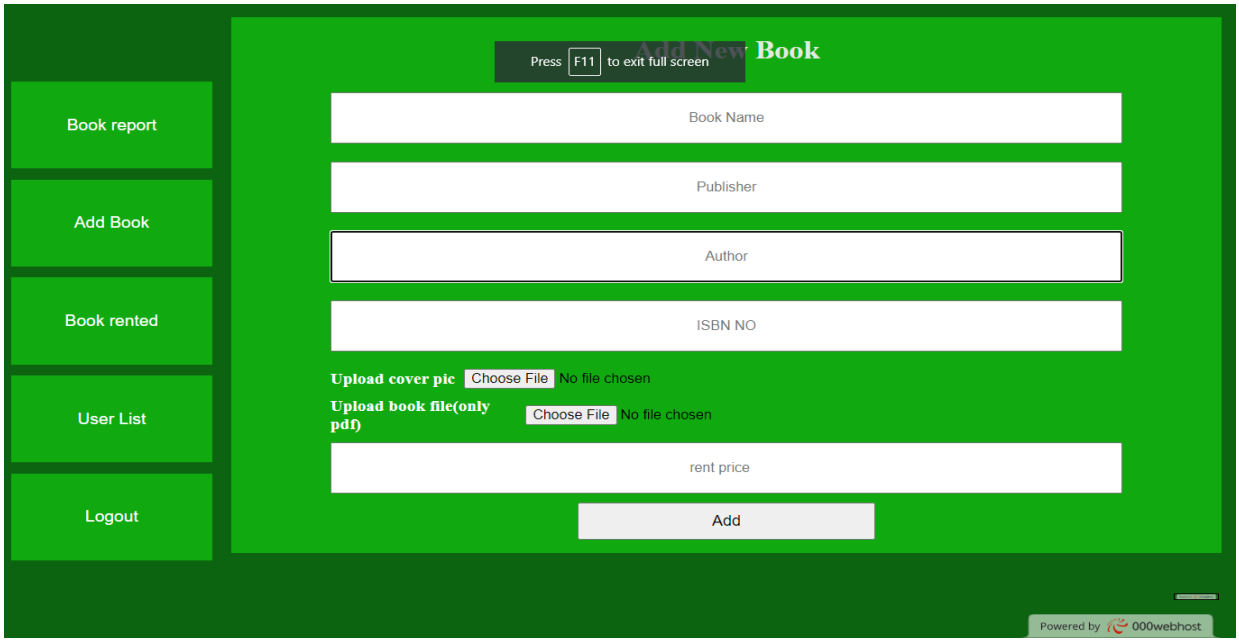
Annex



Admin Login Page

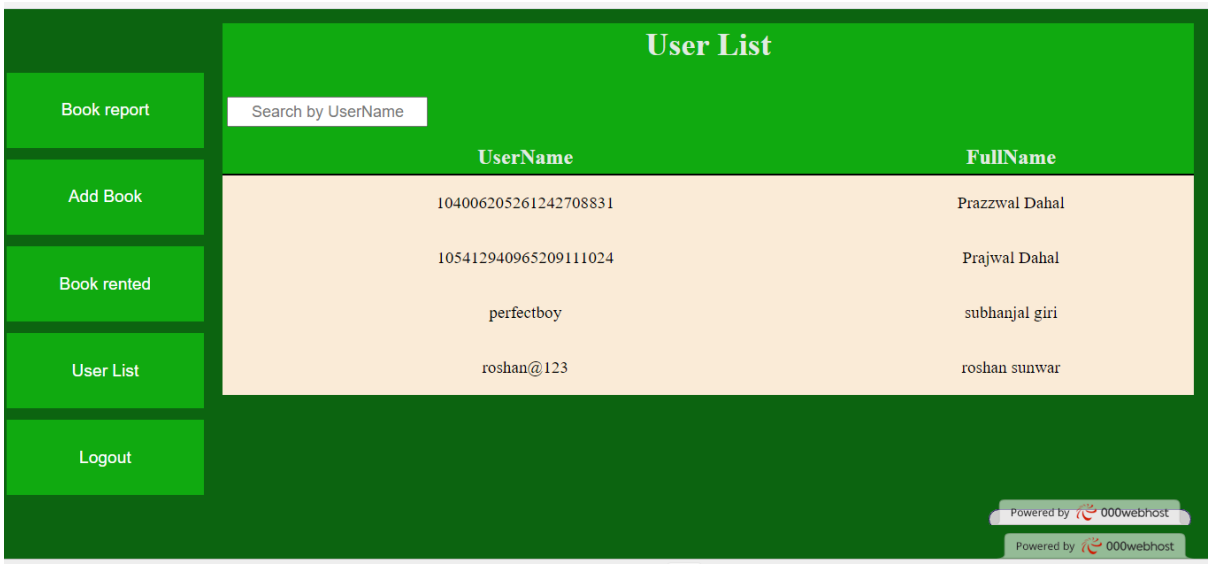


Added Book Page for admin

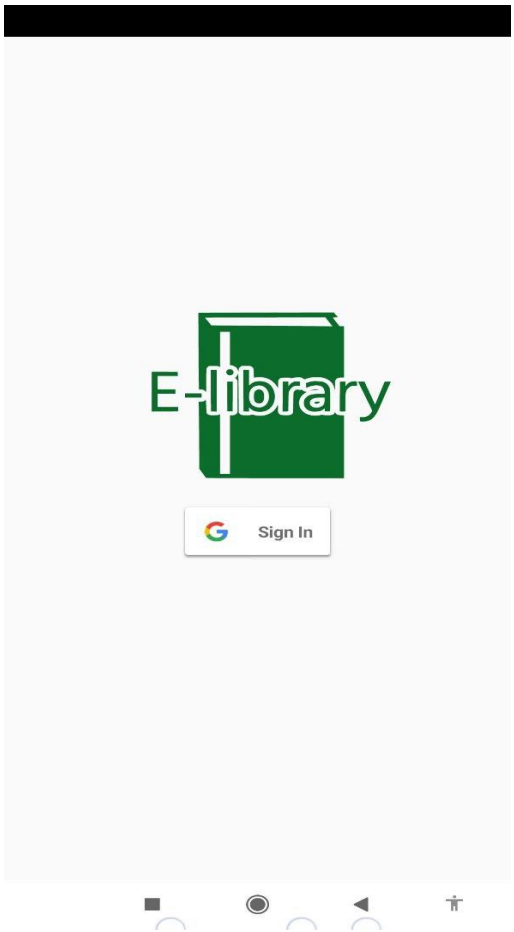


Add book Form for admin

Rented Page History

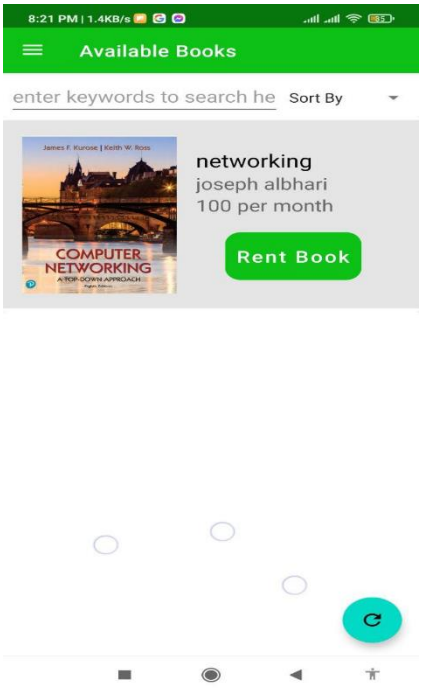


Logged In User List

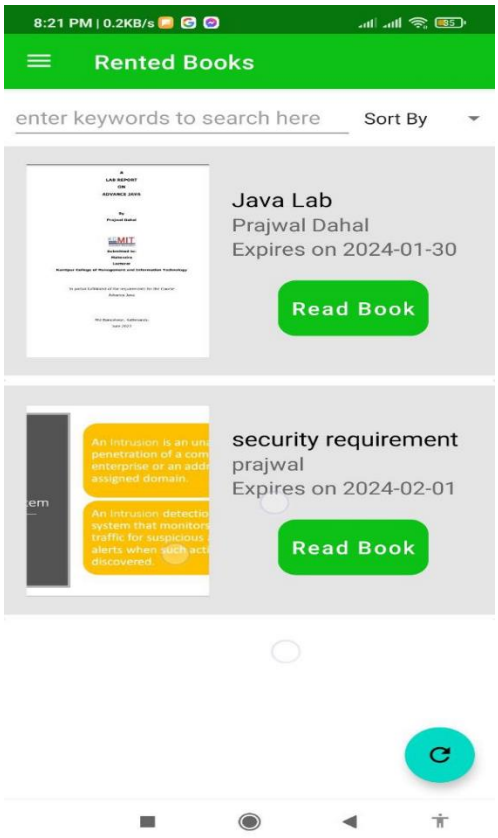


User Login Screen

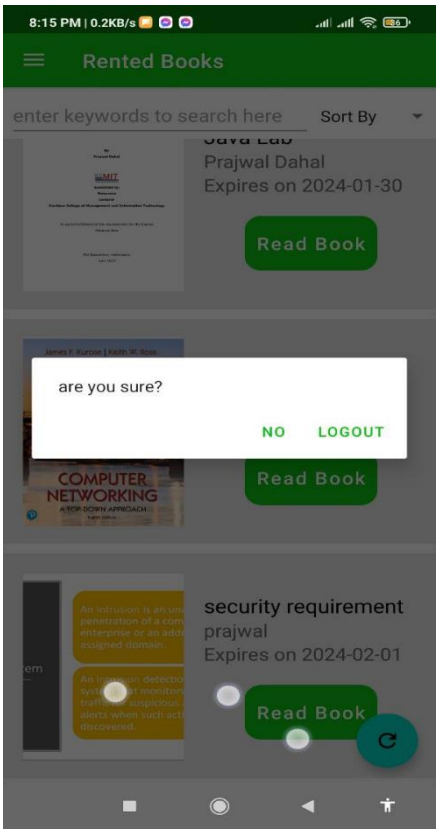
Home Screen



Book discovery Screen



Rented book Screen



Log out