

A  
Project Report  
On  
ReUseBook – Reusing and Reselling of Books

Developed By  
*Ram Ghumaliya [210511211004]*  
*Jigar Dankhara [210511211002]*

as  
Partial Fulfillment of Semester III of  
Master of Computer Applications  
for A.Y. 2022 - 2023

Under the Guidance of  
Prof. Vivek Dave

Submitted To  
Department of MCA  
Faculty of IT & Computer Science  
PARUL University





## **CERTIFICATE**

This is to certify that **Mr. Ram Ghumaliya, Enrollment No. 210511211004** and **Mr. Jigar Dankhara, Enrollment No. 210511211002** students of Master of Computer Applications has satisfactorily completed the Minor Project on **“ReUseBook – Reusing and Reselling of Books”** at **Faculty of IT & Computer Science, Parul University** as partial fulfillment of MCA Semester III.

Seat No. \_\_\_\_\_  
\_\_\_\_\_

Date of Submission: \_\_\_\_\_

\_\_\_\_\_  
Internal Guide

\_\_\_\_\_  
Dean - FITCS

**Faculty of IT & Computer Science**  
**PARUL University, Vadodara**

## **Preface**

The major purpose of each master of computer application student is to gain as much of practical and industrial knowledge as possible. Being able to have a practical understanding by work with the project and project team is a lifetime experience. We are grateful for having a project, since practical knowledge is vital as theoretical knowledge.

We have gained a great deal of experience in the development of several strategies which may be used in the project development. The steps for our carrier are this project.

We are pleased to introduce this initiative. Proper attention was paid during the development of the project to understand. Also, various software engineering & web development concepts have been implemented.

## Acknowledgement

This project is not possible without the exceptional support of our guide, **Prof. Vivek Dave** (Head of Department) his passion, knowledge and exacting attention to detail have been a stimulation and kept our work on track and also endures to complete the final draft of this project.

We would also like to special thanks to **Dr. Priya Swaminarayan (Dean)** and all our faculty member and staff, who encouraged us during this project Then, we would also like to thank our friends who helped us to accomplished this project and give motivation to invoked on track to finalize this project.

Last but not the least, we would like to thanks our parents to believe in us.

Thanking you....

Your sincerely...

Ram Ghumaliya [210511211004]

Jigar Dankhara [210511211002]

## INDEX

No.	Description	Page No.
1.	<b>About Department of MCA</b>	1
2.	<b>Project Profile</b> <ul style="list-style-type: none"> <li>2.1 Project Definition</li> <li>2.2 Project Description</li> <li>2.3 Existing System / Work Environment</li> <li>2.4 Problem Statements</li> <li>2.5 Need for New System</li> <li>2.6 Proposed System &amp; Features</li> <li>2.7 Scope</li> <li>2.8 Outcomes</li> <li>2.9 Tools &amp; Technology used</li> <li>2.10 Project Plan</li> </ul>	2
3.	<b>Requirement Analysis</b> <ul style="list-style-type: none"> <li>3.1 Feasibility Study</li> <li>3.2 Users of the System</li> <li>3.3 Modules</li> <li>3.4 Process Model</li> <li>3.5 Hardware &amp; Software Requirements</li> <li>3.6 Use Cases</li> <li>3.7 Use Case Diagram</li> </ul>	7
4.	<b>Design</b> <ul style="list-style-type: none"> <li>4.1 Use Case Scenarios</li> <li>4.2 Diagrams                             <ul style="list-style-type: none"> <li>4.2.1 class diagram</li> <li>4.2.2 Sequence Diagram</li> <li>4.2.3 Activity Diagrams</li> <li>4.2.4 Deployment Diagram</li> <li>4.2.5 Entity Relationship Diagram</li> </ul> </li> <li>4.3 Data Dictionary</li> </ul>	13
5.	<b>Implementation</b> <ul style="list-style-type: none"> <li>5.1 Form Layouts</li> <li>5.2 Report Layouts</li> <li>5.3 Coding Convention (Business Logic)</li> </ul>	24
6.	<b>Testing</b> <ul style="list-style-type: none"> <li>6.1 Test Strategy</li> <li>6.2 Test Cases</li> </ul>	55
7.	<b>Future Enhancement</b>	63
8.	<b>Bibliography</b>	64

## 1. About Department of MCA

PARUL University Parul University is a legitimate university established under Gujarat Private University Act 2009, after legislation passed by the Government of Gujarat on 26th March 2015 giving University status to Parul Group of Institutes functioning under the aegis of Parul Arogya Seva Mandal Trust.

### Faculty of IT & Computer Science

Faculty of IT and Computer Science, Parul University has materialized as one of the prime IT education providers at global level. Various departments under Faculty of IT and Computer Science strive in preparing IT-industry ready professionals by means of various skill development courses, vocational courses, co-curricular & extra-curricular activities, industry visits and expert lectures.

### MCA Department

The Department of Master of Computer Application at Parul University emphasizes on building professionals in the domain of computer applications by providing necessary environment by means of facilitating suitable blend of technical and non-technical learning experience. The department cultivates students in various curricular, cocurricular and extra-curricular activities in order to produce future system analysts, system designers, system programmers, application programmers, testing professionals, system managers, project managers, researchers and other leading positions in systems/IT department.

The department offers various subjects from diversified technical/non-technical areas such as – core IT domain, management, communication skills, mathematics & logic building and rich pool of elective subjects.

The department of MCA focuses on project-based learning, and hence students are motivated to work on tiny hands-on projects in practical oriented subjects to get better exposure. Moreover, throughout their MCA studies, students are required to work on around 2 mini/major projects in individual/team to get enough confidence on software-development and thereby become industry-ready.

## 2. Project Profile

### 2.1 Project Definition

#### ReUseBook – Reusing and Reselling of Books

“ReUseBook” shall be developed as an android application which will allow learners and book lovers to resell and reuse books and other resources. Users can buy used books from others. This application is being development for a university campus. This application can be used by students who are having financial issues or students who want to buy or sell used books.

Using this application, students can communicate with other students. This application can be helpful for those students who don't know much other students to get books and other resources online. so “ReUseBook” can be very helpful for students to get or to sell/share books and other resources by spending minimum cost and time.

### 2.2 Project Description

Buying new books is expensive and getting used ones are harder. We can sell old books for affordable prices using this application, perfect for students who don't want to spend more than they need to. **ReUseBook**” is android application where many students can communicate with each other to sell and buy used books with in one university.

Student can register their self with enrolment number. They can create a post for books or purchase a book from other students. this android application shall use enrolment number so other venders can not registered them self into this system. Because of it students can get books at minimum cost and less effort. this application can be having a facility that students can communicate with each other using WhatsApp chat application and if any students want to make a call, then it can also possible with calling feature of this application.

**ReUseBook**” can be very helpful for those students who having a new admission in university. So probably they don't have any contact with other students.so using this application they can contact with existing students and have a look for required books using searching feature.

#### 2.2.1 Users of The System

- Student
- Administrator

#### 2.2.2 Modules

- Registration and login
- Add books for reselling a books

- Search for books
- WhatsApp chat and calling
- Feedback
- View profile

## 2.3 Existing System / Work Environment

We have 3 existing system based on old book reselling concept. Which is describe as below:

### 2.3.1 Ziffit

Ziffit can both identify and price your books quickly and easily by using the barcode scanning feature. Once it's identified the book, the app will assign a price to it and add it to your trading basket. However, if you don't think the app is offering a fair price, you can always edit your basket to remove books.

Keep in mind that the app doesn't allow you to make trades under 10 items, or trades under fifteen dollars, so make sure you have enough books in order to make a trade.

### 2.3.2 Depop

Along with fashion and beauty products, Depop also has a wide range of books on sale from hundreds of individual sellers. These pre-owned books can be bought for a fraction of their original price, with most of them still being in great condition.

Depop allows you to communicate directly with sellers, where you can ask about the product, or get in touch if you have issues with your item. You can even negotiate on the price a little.

### 2.3.3 ThriftBooks

ThriftBooks is a great app for finding some great second-hand books easily. You can simply use the app's search bar to find what you're looking for, and you surely won't be short for choice, with over 13 million titles available to buy. These titles also include vinyl records, video games, and DVDs, all of which come at a discounted price.

### 2.3.4 Limitation on existing system

Here in all existing application, we having a problem that all people can login to this system so if vendor can register them self into this system, then it will become problem for students. Because vendors are going to include their profit in book price and if it happens then students can not buy books at minimum cost.

## 2.4 Problem Statements

- To develop a dynamic mobile app for students who wants to use or purchase used books by connecting with campus people.



- Using these mobile applications only students can register with their enrolment number. so external vendors can not register themselves in this application. Because of it, students can get used books at minimum cost and minimum time.
- users can check all activity in their profile so in that user can add books for reselling and check books for purchase also.
- Users can communicate with each other using WhatsApp chat application and calling functionality. This can give more security for communication.
- many students can not purchase new books and they don't have enough contact to purchase books from other students at minimum cost and minimum time.

## 2.5 Need for New System

In daily life many students facing many issues to purchase old books which is described below:

- there is no application who doesn't allow to registered external vendors to sell old books.
- many students don't have enough contacts with other students so using it they can buy old books.
- Students want books at minimum cost, efforts and time.
- Student having a problem with to find a particular book out of university. Because it is not necessary that all books are in market which is used within one university.

## 2.6 Proposed System & Features

Nowadays, many applications are there to sell old books. But they have many limitations like they are going to include vendors in system. They have fixed minimum rate of books so because of it students can't sell their books at low cost compare to that fix rate. In many applications with old books users can put some other things also. so it does not have proper use.

So, because of many problem **"ReUseBook"** offers many features which is mentioned below:

- Registration with enrolment id
- Post old books for resell
- Purchase books at minimum cost
- Communication between sell and buyer using WhatsApp chat application.
- Having calling facility for communication.
- Maintain own old books store in profile
- Easily get old book from other students within one university without any efforts and contacts.

## 2.7 Scope

- **"ReUseBook"** can be used within one university only. Here in this application, we need student enrolment id for registration so student can use within one university only.
- **"ReUseBook"** wants some basic knowledge about English language and person having basic knowledge about android phone. Person having a device with any one browser.
- **"ReUseBook"** require installation of application and registration with enrolment number. internet connection is also required to use this application.

## 2.8 Outcomes

After using “ReUseBook” application users can get their required books from other users with minimum price and efforts. This application can work as bridge between multiple users for communication and buying or selling old books. Users can manage his/her profile and books for reselling purpose. Admin can manage all user of this application.

## 2.9 Tools & Technology used

### 2.9.1 Tools

#### 2.9.1.1 Android Studio(development)

- Android studio is in syllabus so it is easy to learn and develop an application. Android studio is light weight and fast for development process. android is open-source technology.

#### 2.9.1.2 Firebase(database)

- Firebase is easy to use and it provides a real time data.it is fast and secure database. Firebase provides a seamless way to integrate with Google analytics using which the user behavioural data can be acquired, and insightful information can be generated through it. The data is useful in improving customer retention for businesses and enhancing the user engagement rate for our application.

#### 2.9.1.3 Tools for documentation

- Draw.io is very useful software to generate documentation.draw.io provide lots of shapes to generate documentation graphs and charts.it very easy to use and free of cost. In draw.io we can generate image of graphs and charts.
- Draw.io provides a facility that we need to just drag and drop all shapes so if any developer does not have idea for it then they can easily learn it.

### 2.9.2 Technologies

#### 2.9.2.1 Java(back end)

- Java is platform independent language. it is in syllabus so it is easy to learn and develop an application.
- Java is a simple programming language since it is easy to learn and easy to understand.

#### 2.9.2.2 Xml(front end)

- XML is platform independent and programming language independent, thus it can be used on any system and supports the technology change when that happens. XML supports Unicode. This feature allows XML to transmit any information written in any human language.
- The data stored and transported using XML can be changed at any point of time without affecting the data presentation.

## 2.10 Project Plan

Start date	End date	task	Done by
10/07/2022	15/07/2022	Project profile	Both
16/07/2022	22/07/2022	Requirement analysis	Both
23/07/2022	13/08/2022	Testing	Both
14/08/2022	01/09/2022	design	Both
02/09/2022	10/09/2022	implementation	Both

Table 1: project plan

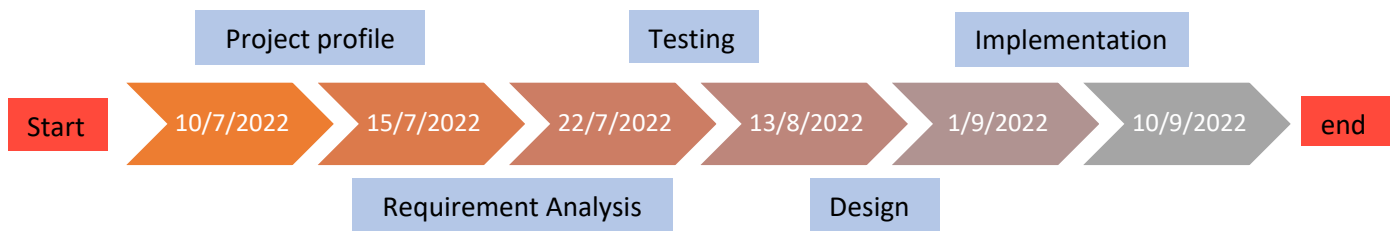


Figure 1: project plan

### 3. Requirement Analysis

#### 3.1 Feasibility Study

##### 3.1.1 Operational feasibility

- “ReUseBook” can be used for those students who wants to purchase and resell old book easily and with in minimum time. Student can register with enrolment so no other venders can register them self in to application.
- So, this application can be used with in one campus and student can easily purchase and resell old books.

##### 3.1.2 Technical Feasibility

- this feasibility deals with technicality of the system instead of storing data of books and user in manual way this system is fully automated. No efficient manpower is required to handle the system.
- Here to develop this application we have used android development tools because using it we can easily develop an application. both developers having knowledge about technology.

##### 3.1.3 Economical Feasibility

- In this application we need to purchase API for WhatsApp chatting and calling facility. And for storing a data some time when we have more transactions that time, we need to pay some amount. And android development tools are open source so for that we don't have to need pay any amount for android studio.
- in economical feasibility we can say this application is feasible because there is no big cost for development.

##### 3.1.4 Social Feasibility

- currently students having many problems to purchase old books. Because there is no such a platform using it student can communicate with each other to purchase old books and resell it. And this system is socially feasible because student can easily communicate with each other freely and easily.

#### 3.2 Users of the System

This system can have two users of the system: -

##### 3.2.1 Student

“ReUseBook” is mainly developed to solve student's problem. Students are very important user of this application. In this system student can register using enrolment number and have a look for books. Also, student can upload their used book over here with affordable price and sell books in this application. Student can purchase according to their requirement and buy old book in minimum cost.so it will be very useful those students who does not want to spend money in new book and having single use only.so for those students “ReUseBook” is very useful application.

### 3.2.2 Administrator

“ReUseBook” is student base application. But whenever if student having a problem that time, we need one person who can solve their problem.so, administrator will be that person who is going to solve user problems. If, any illegal content will be uploaded from user side then administrator can take strict action on it and he/she can block that particular user. administrator can manage all user and watch all activity in this application. Administrator can see all feedback of buyers for book seller.

## 3.3 Modules

### 3.3.1 Registration and login

Registration module can include all information about user like enrolment number, name, contact etc. This module is mandatory for login module. Because if user wants to login this application, then user must have to registered them self in this application first.

Login module include authentication of application user.so for login user must have a user name and password.in case of if user forgot his/her password then we can have a facility that user can reset password using forgot password.

### 3.3.2 books Management and reselling

“ReUseBook” mainly develop for to resell used book. This module going to be added books for reselling. here user can add an old book for reselling with it's expected price and photo of books.

### 3.3.3 book Searching

To purchase an old book user can search a particular book from search module. This module is very important because at a time we can have many books in “ReUseBook” .so, for finding a particular book can be easy and time saving for user using search book module.

### 3.3.4 WhatsApp chatting and calling

When buyer wants to purchase a book that time buyer need to do conversation with seller. we can have a WhatsApp chat and calling module for communication purpose. And buyer and seller can do communication with each other easily.

### 3.3.5 Feedback management

For response of seller, we can have feedback module.in that buyer can add some comments in application and administrator can see feedback for all users.

### 3.3.6 User Profile Management

Every user can have profile in this application. so in this module user can see all activity of him/her. User can check how many books are for sell over here.

### 3.4 Process Model

The Software process model chosen for the project is the “Agile Model”.

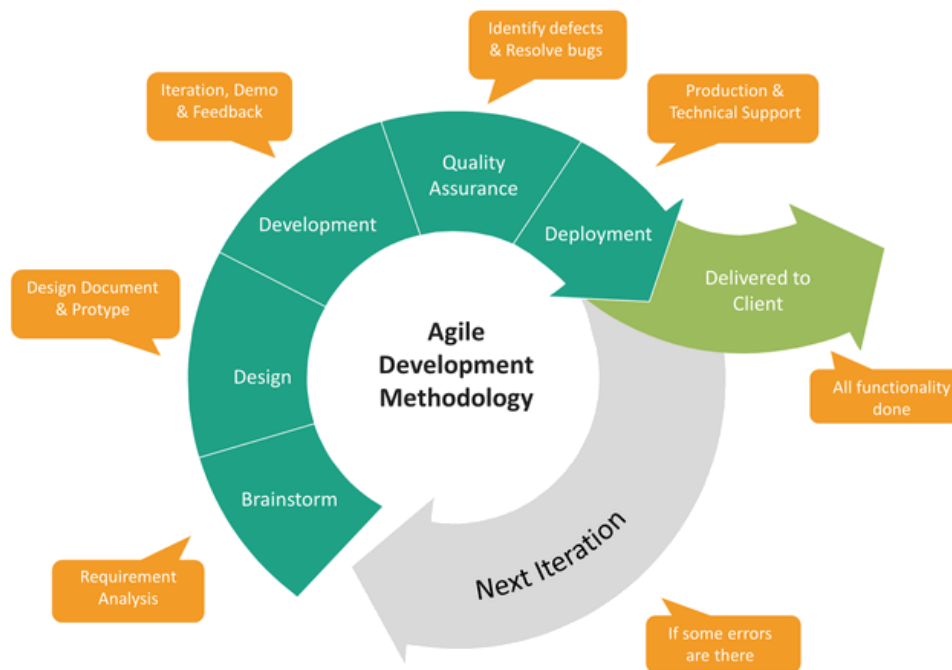


figure 2: agile model

**3.4.1 Requirements gathering:** In this phase, Business requirements and system requirements are being gathered from the client.

**3.4.2 Design the requirements:** UI will be prepared first and then implement and learn angular for design for compatibility with OS.

**3.4.3 Construction/ iteration:** According to Client requirements it will be prepare Testing: UI Testing done for design checking in the Smart devices.

**3.4.4 Deployment:** UI with xml.

**3.4.5 Feedback:** Feedback from the tester, sellers and buyers from the client-side.

#### 3.4.6 Advantages of Agile Model

- Frequent Delivery
- Face-to-Face Communication with clients.
- Efficient design and fulfils the business requirement.
- Anytime changes are acceptable.
- It reduces total development time.

### 3.4.7 Disadvantages of Agile Model

- The shortage of formal documents, creates confusion and crucial decisions taken throughout various phases can be misinterpreted at any time by different team members.
- Due to the lack of proper documentation, once the project completes and the developers allotted to another project, maintenance of the finished project can become difficult.

### 3.4.8 First week scrum master

Story	To do	In progress	To verify	done
Require analysis	-Meeting with client -Requirement verification -Map the financial requirements	-Manage time for meeting -Match all requirement with client -Estimate costing of development	-The expected response time. -The quote for the next story	-Meeting arranged with client -Receive the internal client's approval for the requirement -Set up all hardware and software requirement for cost estimate

Table 2: First week scrum master

## 3.5 Hardware & Software Requirements

### 3.5.1 Developer-Side Configuration

#### 3.5.1.1 Hardware

RAM	4GB
SSD	Needed

Table 3: Hardware Requirements

#### 3.5.1.2 Software

Operating System	Windows 10
Database	Firebase
Backend	Java
Front-End	Xml
Tool	Android studio

Table 4: Software Requirements

### 3.5.2 Client -Side Requirements

#### 3.5.2.1 Hardware

RAM	1GB or higher
Device	Android smart phone

Table 5: Hardware Requirements

#### 3.5.2.2 Software

Operating System	Android
Application runner	APK
Connection	Internet (WIFI, ethernet)

Table 6: Software Requirements

### 3.6 Use Cases

- We have mainly 2 actors in this android application: -

#### 3.6.1 Admin

- User management
- User authentication & authorisation
- Book management
- Book selling
- Book purchasing
- Feedback management
- Admin profile management
- WhatsApp chatting & calling

#### 3.6.2 User

- User profile management
- Book management
- Book selling
- Feedback
- WhatsApp chatting & calling
- Payment status
- Confirmation message



### 3.7 Use Case Diagram

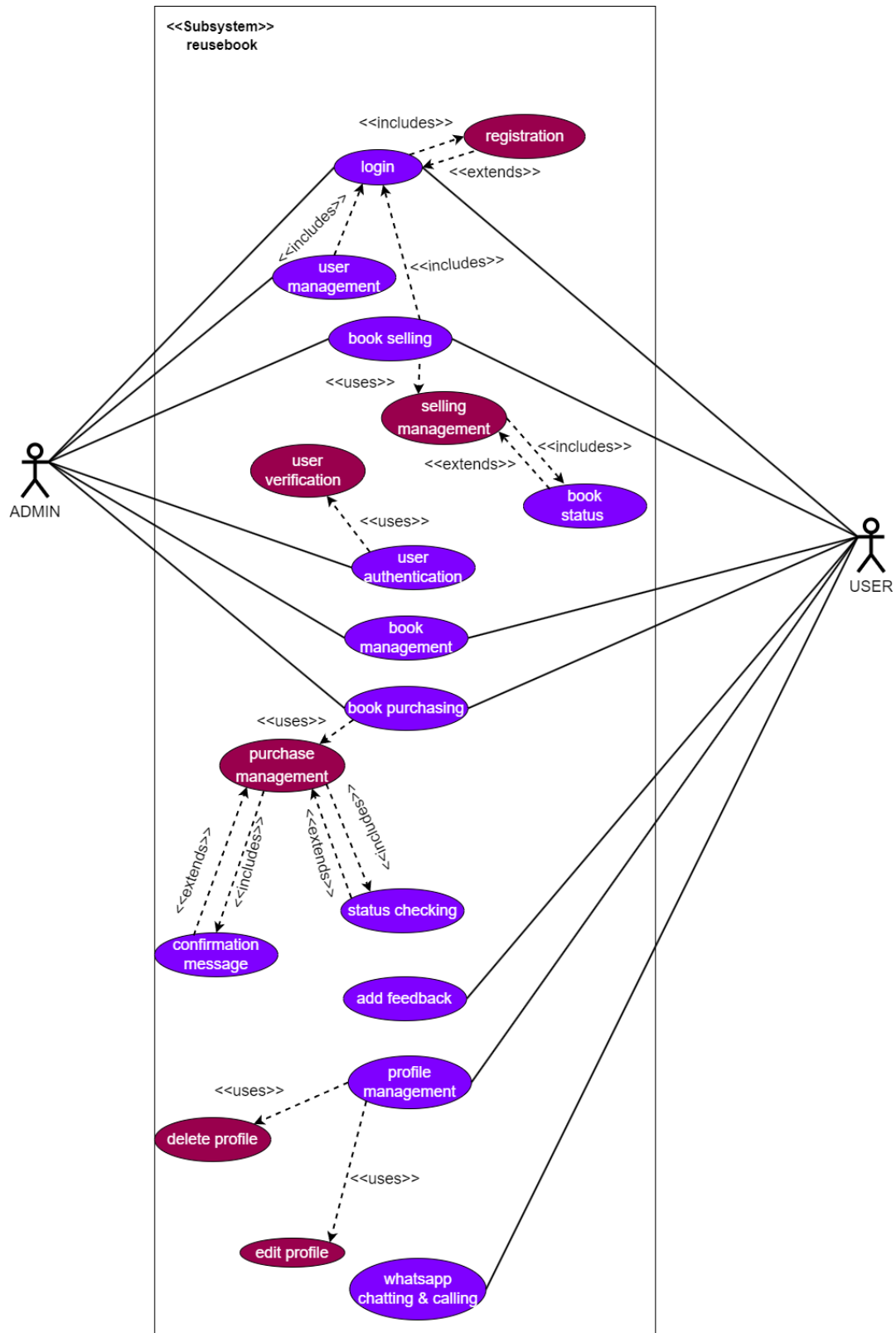


Figure 3: Class Diagram

## 4. Design

### 4.1 Use Case Scenarios

Use case Id	UC001
Use case Title	Users Login
Actor	Admin, User
Description	user can login into system by their user id and password
Pre-condition	user must have registered with system.
Post-condition	After successful login user must be redirect to the it's respective dashboard.
Basic Flow	user will open its login page and enter user Id and password field and click on login button. If both are correct then user will redirect to dashboard otherwise system will give error

Table 7: Use Case Scenarios for Login

Use case Id	UC002
Use case Title	Used Book Reseller
Actor	Admin
Description	Admin can login into system and can Manage whole system.
Pre-condition	Admin must have login the systems
Post-condition	After successful login Admin can manage user, book activity, feedback activity
Basic Flow	Admin will login into system then click on users and perform add, remove and edit operations.

Table 8: Use Case Scenarios for user management

## 4.2 Diagrams

### 4.2.1 class diagram

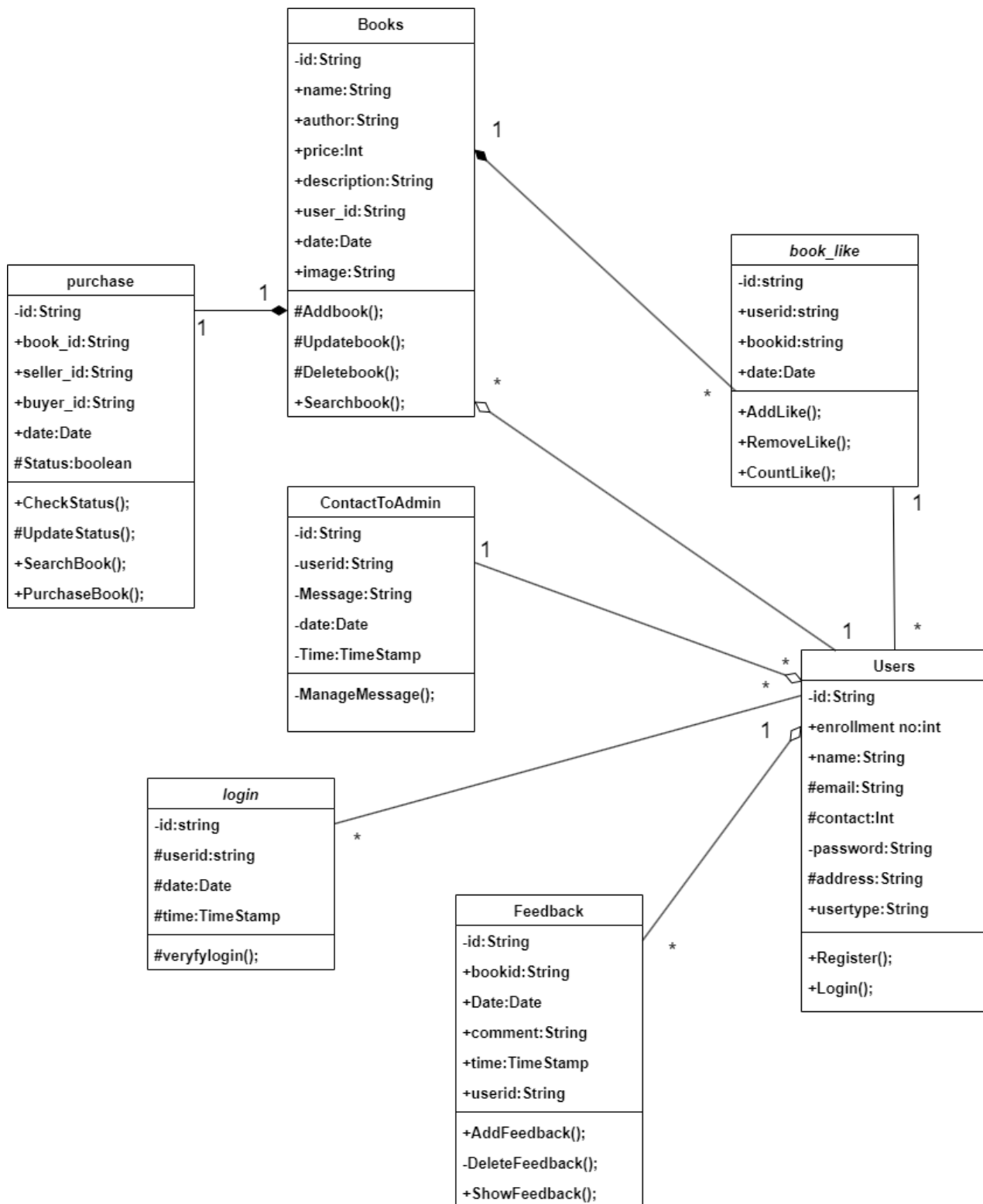


Figure 4: Class Diagram

## 4.2.2 Sequence Diagram

### 4.2.2.1 Sequence for Admin

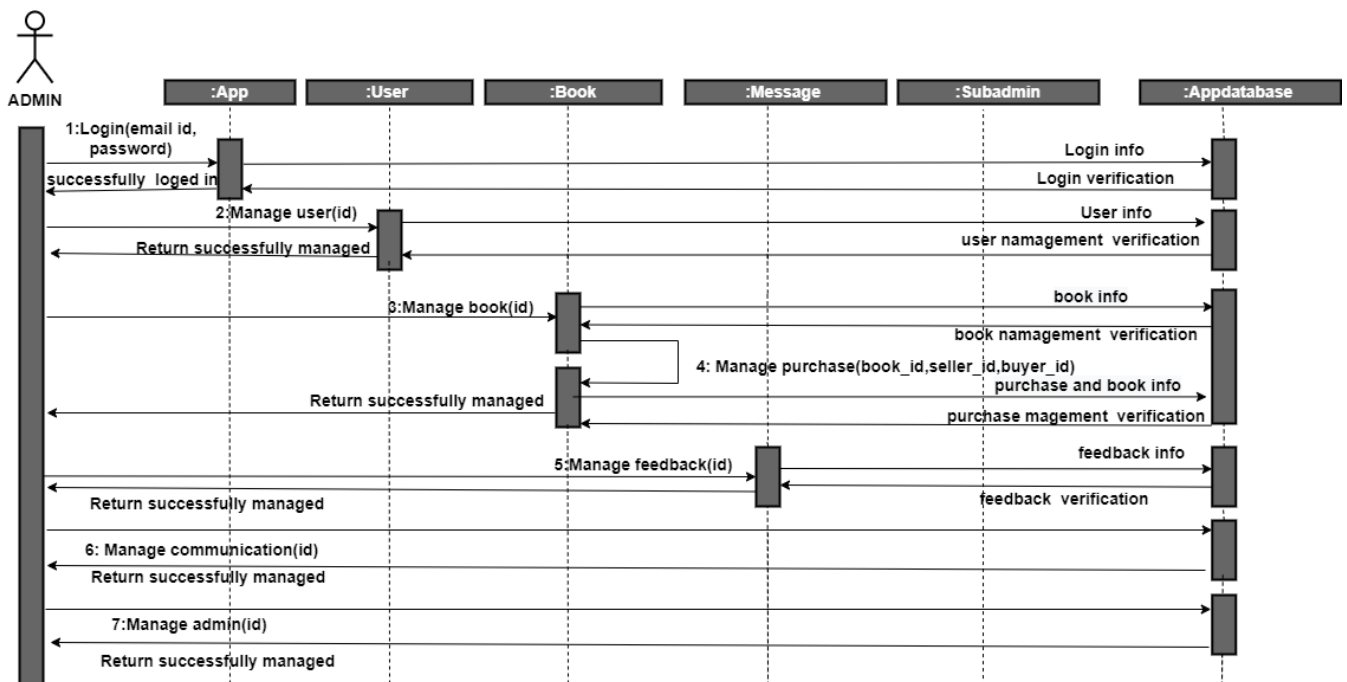


Figure 5: Sequence Diagram of Admin

### 4.2.2.2 Sequence for User

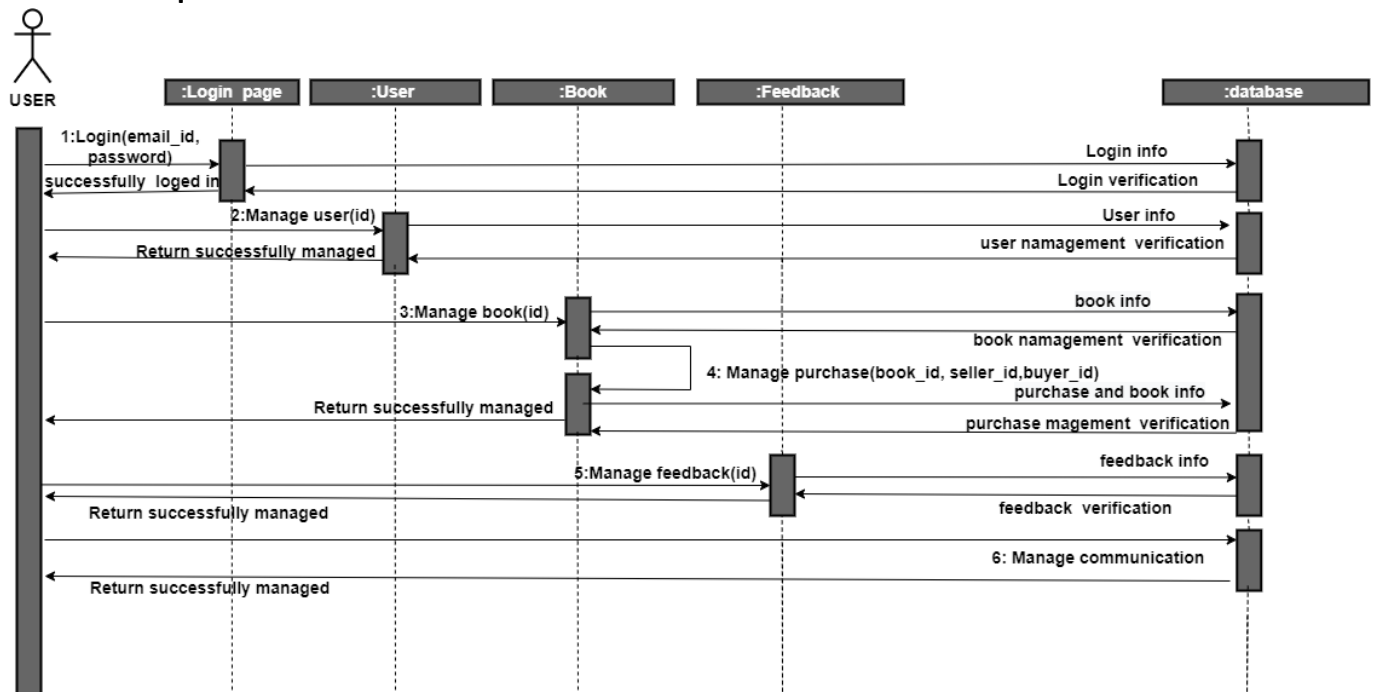


Figure 6: Sequence Diagram of User

### 4.2.3 Activity Diagrams

#### 4.2.3.1 System activity diagram

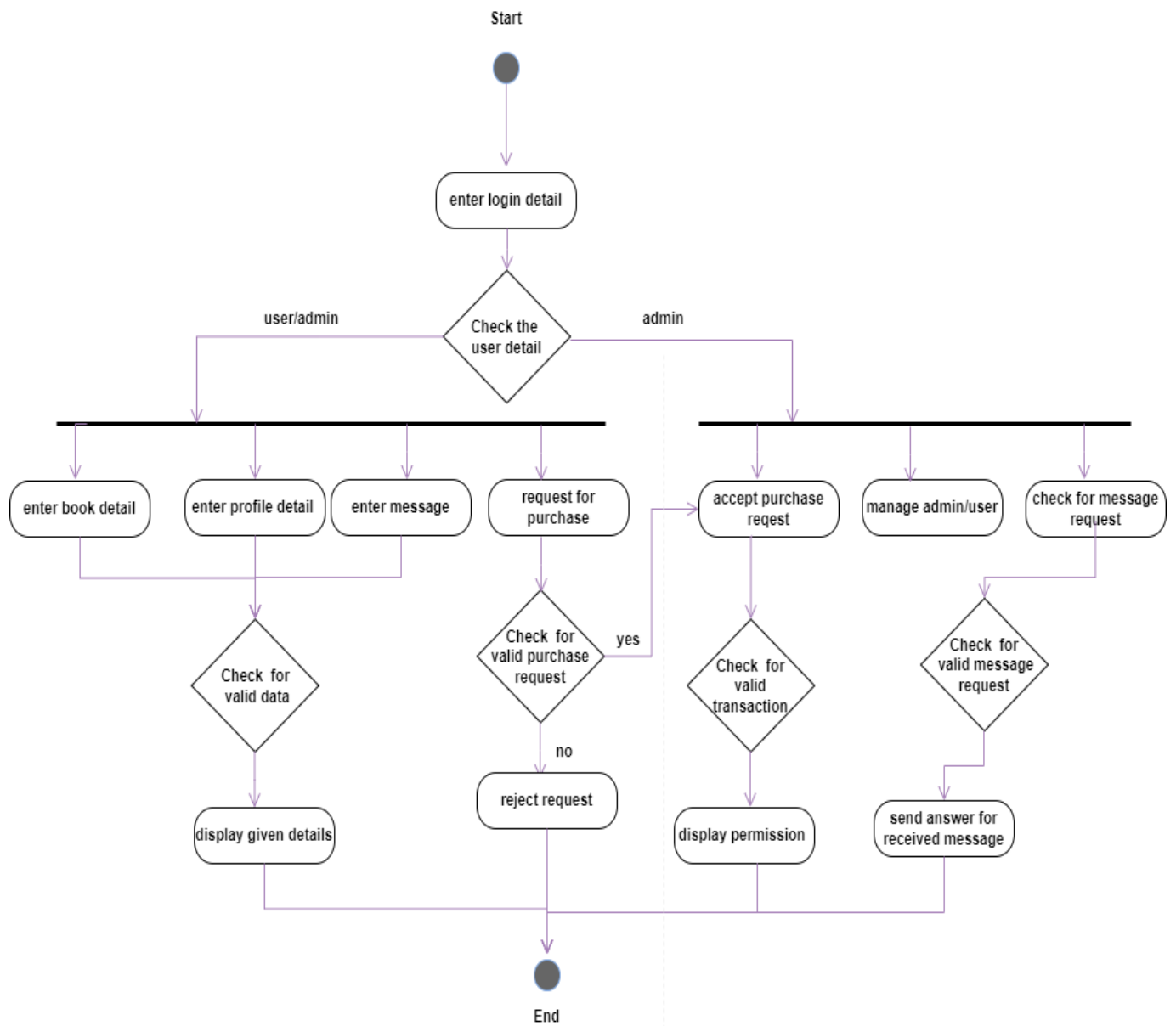


Figure 7: Activity Diagram of ReUseBook

## 4.2.3.2 login Activity Diagram

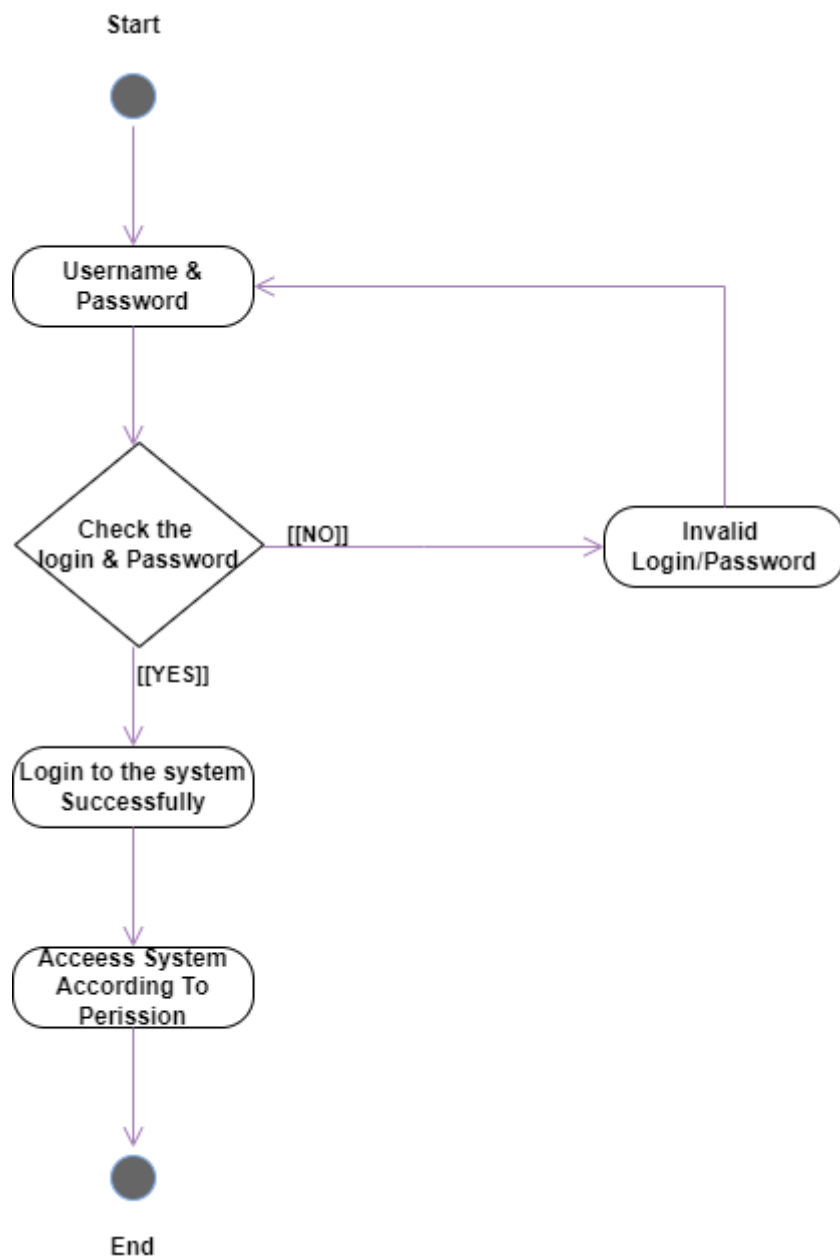


Figure 8: Activity Diagram of Login

#### 4.2.3.3 Book Management Activity Diagram

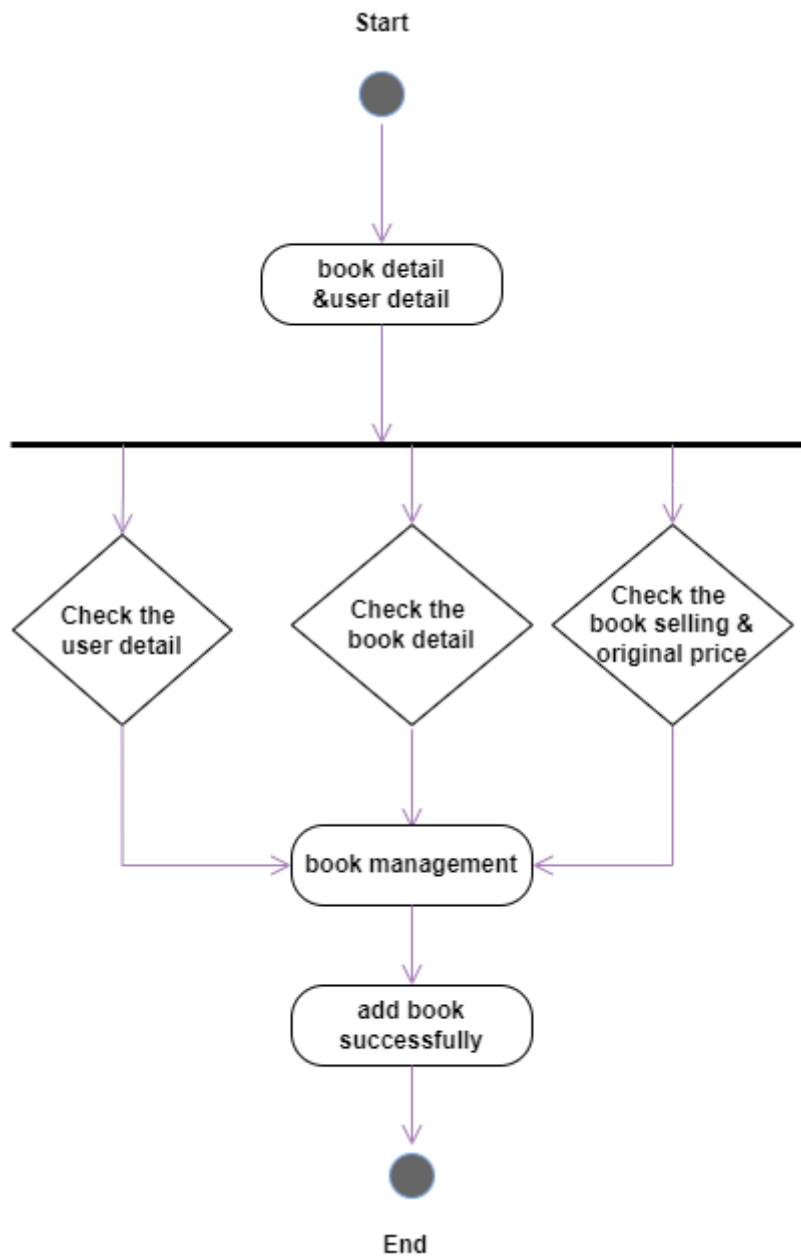


Figure 9: Activity Diagram of Book Management

#### 4.2.3.4 Feedback Management Activity Diagram

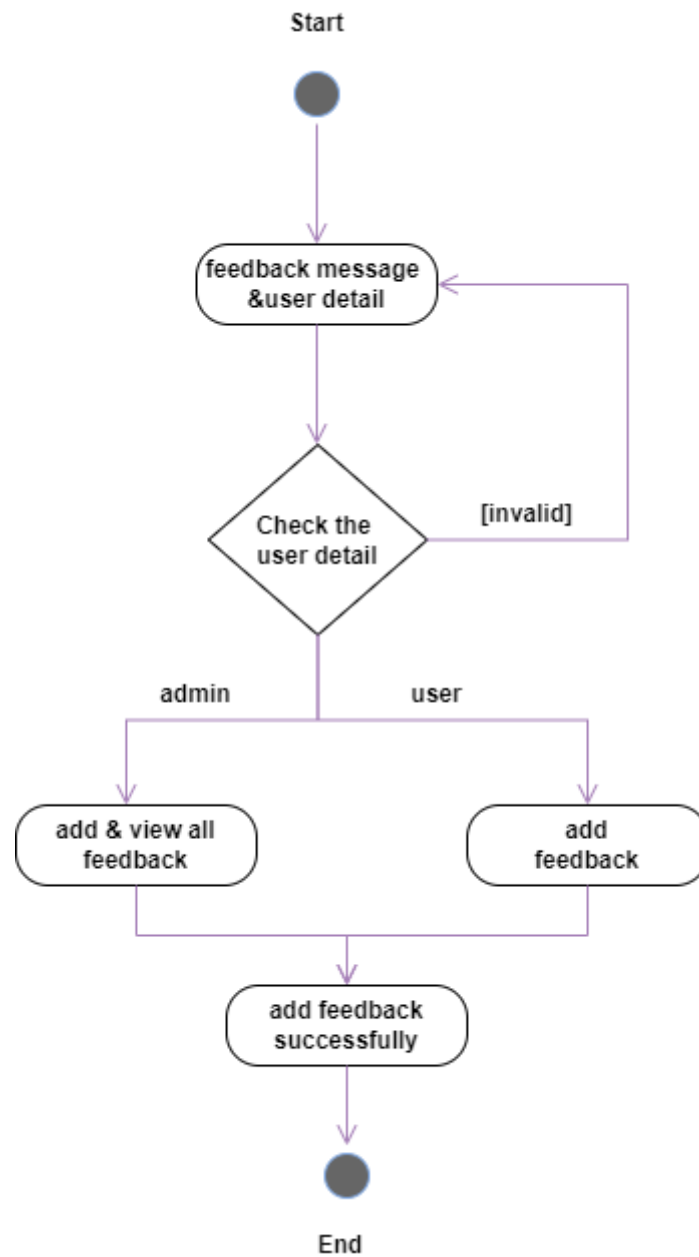


Figure 10: Activity Diagram of Feedback Management



#### 4.2.4 Deployment Diagram

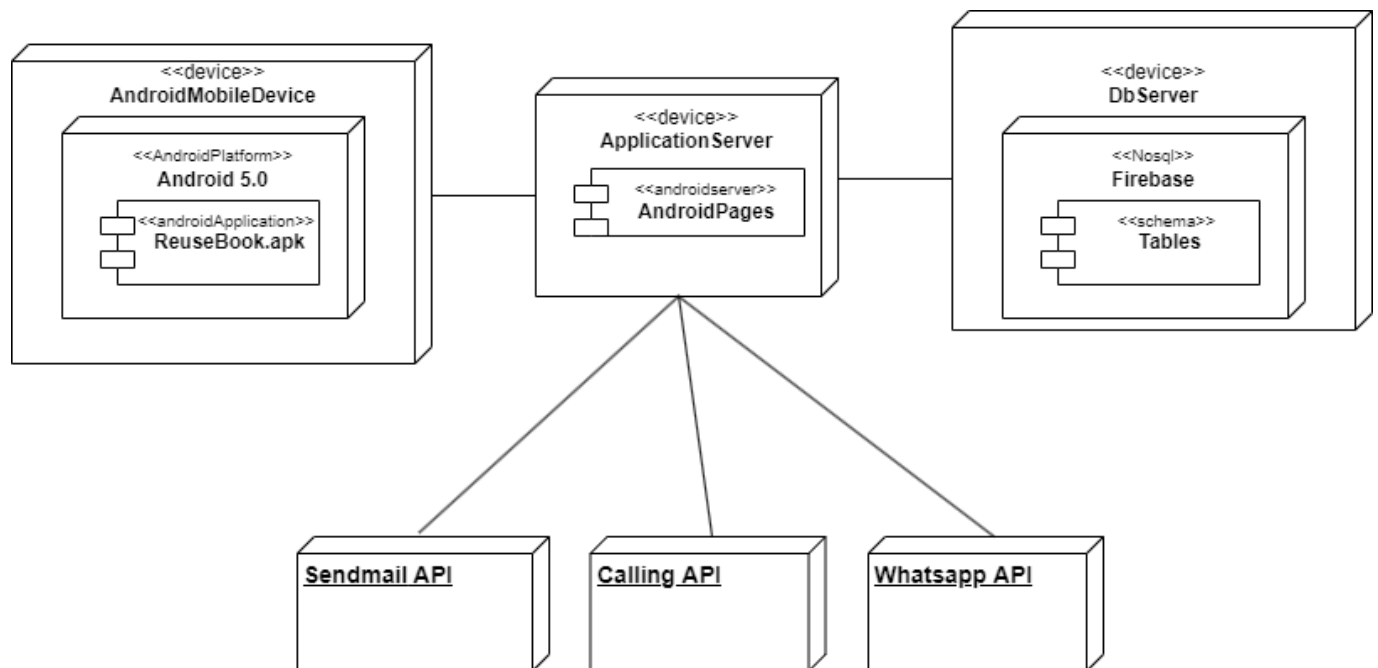


Figure 11: Deployment Diagram

#### 4.2.5 Entity Relationship Diagram

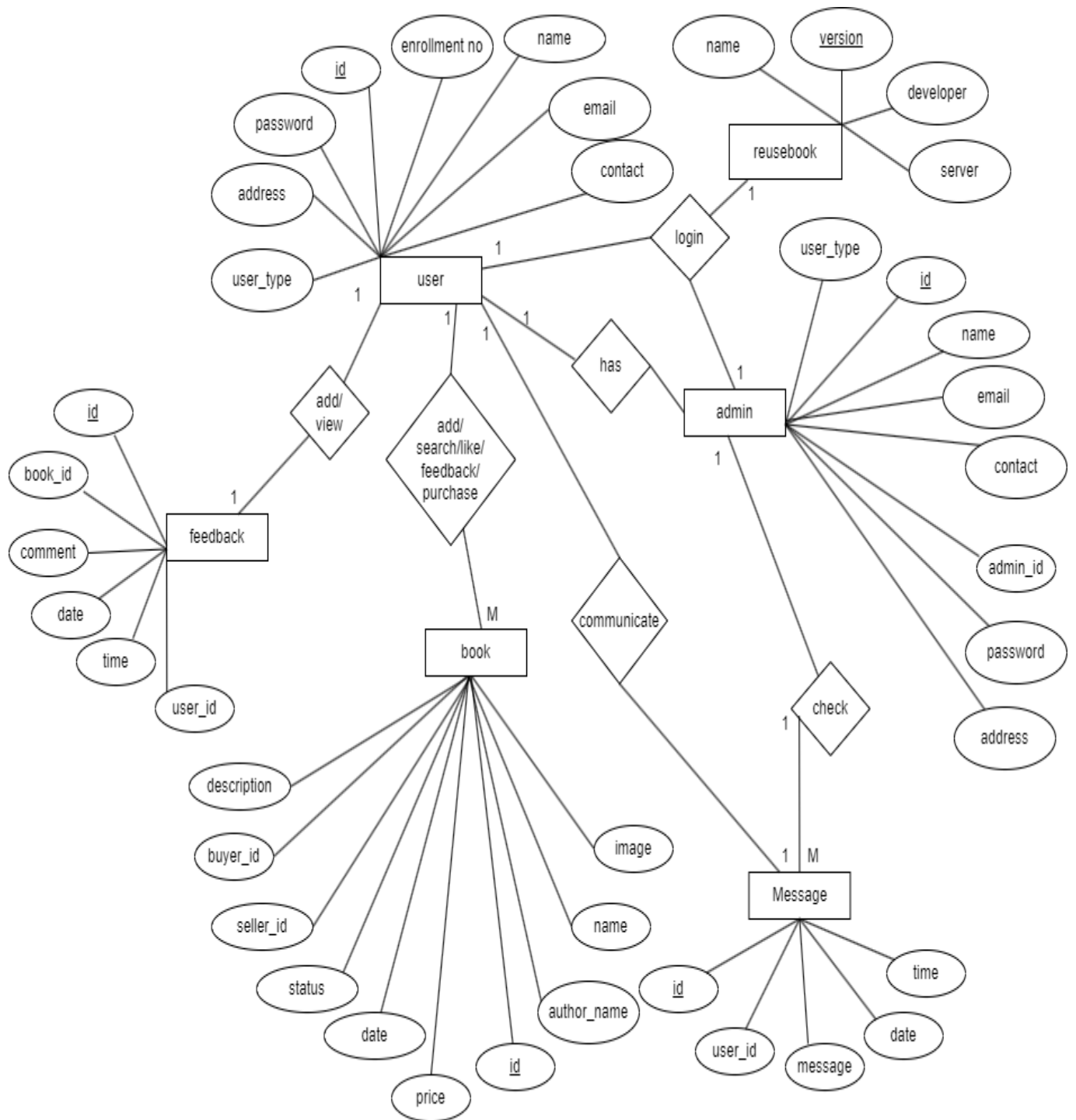


Figure 12: Entity Relationship Diagram

## 4.3 Data Dictionary

### 4.3.1 user\_info

Data Fields	Data Constraints	Datatype	Data Description
Id	Primary key	String	id of user
Enrolment_no	Not null	int	-
Name	Not null	String	-
Email	Not null	String	-
Contact	Not null	int	-
password	Not null	String	-
Address	Not null	String	-
User_type	Not null	String	-

Table 9: Data Dictionary of user\_info

### 4.3.2 books

Data Fields	Data Constraints	Datatype	Data Description
Id	Primary key	String	id of book
Name	Not null	String	-
Author	Not null	String	-
Price	Not null	Int	-
description	Not null	String	-
User_id	Foreign key	String	User id of user_info as foreign key
Date	Not null	date	Date of book added for selling
image	Not null	String	-

Table 10: Data Dictionary of books

### 4.3.3 feedback

Data Fields	Data Constraints	Datatype	Data Description
Id	Primary key	String	id of feedback
book_id	Foreign key	String	book id of books as foreign key
comment	Not null	String	-
date	Not null	Date	-
time	Not null	Timestamp	-
User_id	Foreign key	String	User id of user_info as foreign key

Table 11: Data Dictionary of feedback

#### 4.3.4 Login

Data Fields	Data Constraints	Datatype	Data Description
Id	Primary key	Int	Id of login
User_id	Not Null	Int	Id of user table as foreign key
Time	Not Null	Timestamp	-
date	Not Null	Date	-

Table 12: Data Dictionary of login

#### 4.3.5 Purchase

Data Fields	Data Constraints	Datatype	Data Description
Id	Primary key	Int	Id of purchase
Book_id	Foreign key	Int	Id of book table as foreign key
Seller_id	Foreign key	Int	Id of seller table as foreign key
Buyer_id	Foreign key	Int	Id of buyer table as foreign key
date	Not Null	Date	Date of Login
Status	Not Null	Varchar	Purchase status

Table 13: Data Dictionary of purchase

#### 4.3.6 Book\_like

Data Fields	Data Constraints	Datatype	Data Description
Id	Primary key	Int	Id of book like
user_id	Foreign key	Int	Id of user_info table as foreign key
book_id	Foreign key	Int	Id of books table as foreign key
date	Not Null	Date	-

Table 14: Data Dictionary of book\_like

#### 4.3.7 Contacttoadmin

Data Fields	Data Constraints	Datatype	Data Description
Id	Primary key	Int	Id of contact to admin
user_id	Foreign key	Int	Id of user_info table as foreign key
Message	Not Null	Varchar	Description of contact message
Time	Not Null	Time	-
date	Not Null	Date	-

Table 15: Data Dictionary of contacttoadmin

## 5. Implementation

### 5.1 Form Layout

#### 5.1.1 Registration page

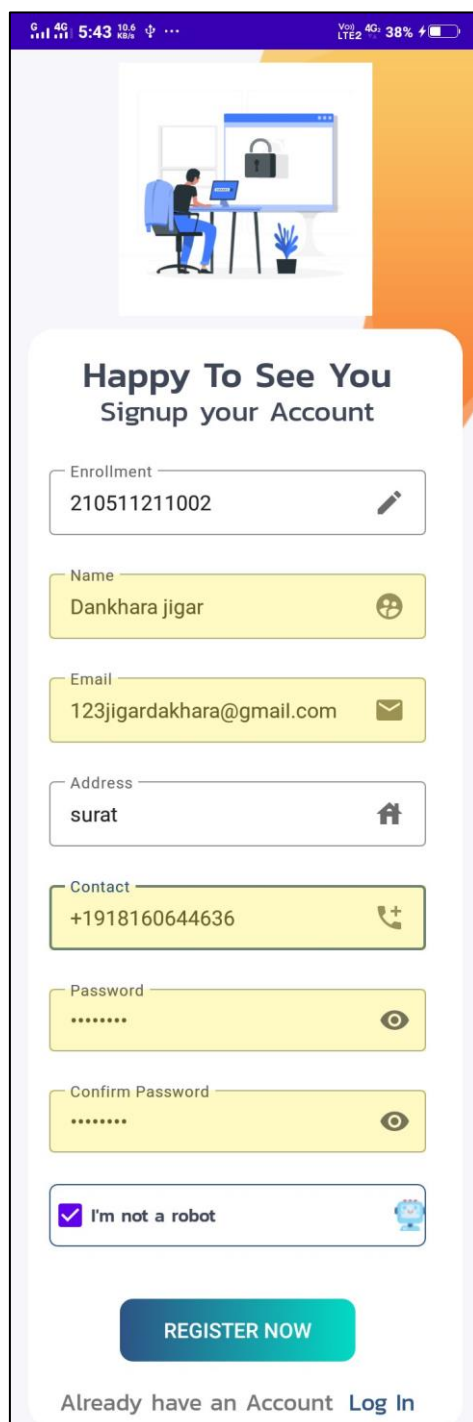
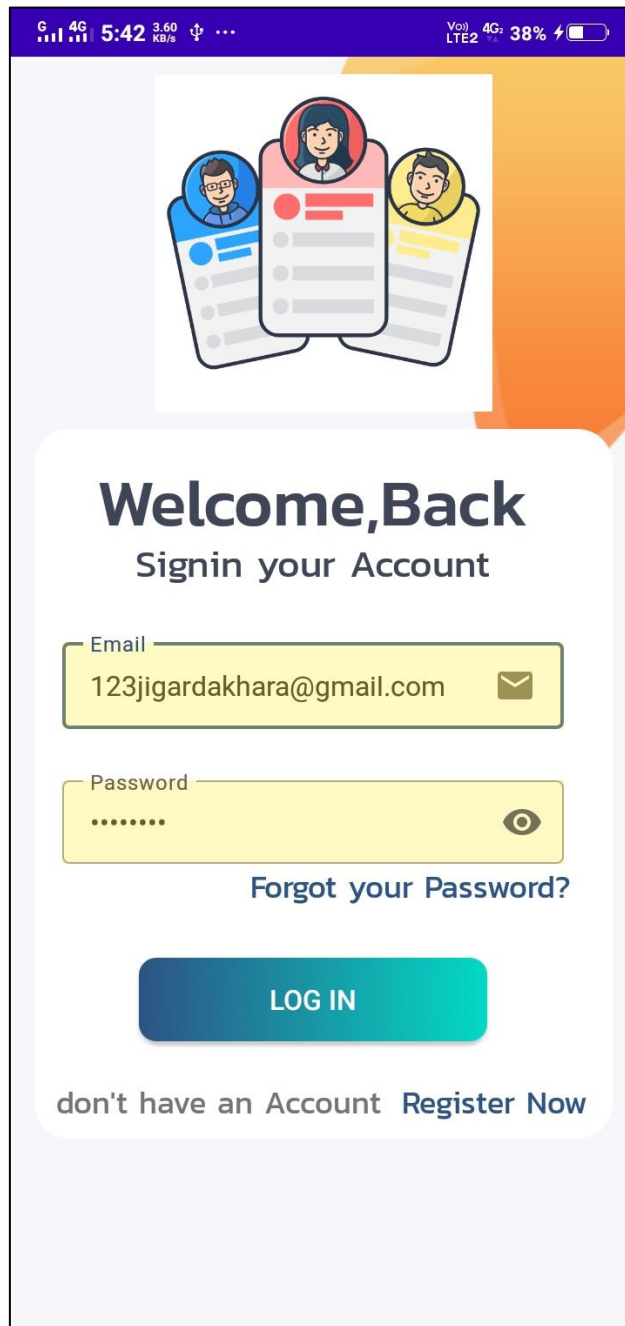
A mobile application registration form displayed on a smartphone screen. The screen has a purple status bar at the top showing '4G', '5:43', '10.6 KB/s', and 'LTE2 38%'. The app's header features an illustration of a person at a desk with a computer and a padlock icon. The main heading reads 'Happy To See You' followed by 'Signup your Account'. The form contains several input fields: 'Enrollment' with the value '210511211002', 'Name' with 'Dankhara jigar', 'Email' with '123jigardakhara@gmail.com', 'Address' with 'surat', 'Contact' with '+1918160644636', 'Password' (masked with dots), and 'Confirm Password' (masked with dots). There is a checkbox labeled 'I'm not a robot' which is checked. At the bottom, there is a blue 'REGISTER NOW' button and a link 'Already have an Account Log In'.

Figure 13: Registration page

**Description:** User and admin can register using enrolment no, name, email, address, contact and password in reusebook.

### 5.1.2 login page

A mobile application login screen. At the top, there's a status bar with network icons (G, 4G), time (5:42), speed (3.60 KB/s), and battery (38%). Below the status bar is a header with three circular profile icons. The main content area has a white background with a light blue border. It features the text "Welcome, Back" in a large, bold, dark blue font, followed by "Signin your Account" in a smaller, bold, dark blue font. Below this are two input fields: "Email" with the value "123jigardakhara@gmail.com" and a green envelope icon, and "Password" with a series of dots and a green eye icon. A link "Forgot your Password?" is positioned below the password field. A large, rounded rectangular button with a blue-to-teal gradient and the text "LOG IN" in white is centered below the link. At the bottom, the text "don't have an Account" is followed by a link "Register Now" in blue.

5:42 3.60 KB/s 38%

VoLTE 4G

Welcome, Back

Signin your Account

Email 123jigardakhara@gmail.com

Password .....

Forgot your Password?

LOG IN

don't have an Account Register Now

Figure 14: Logging page

**Description:** User and admin can login in through their registered email id and password in reusebook.

### 5.1.3 Feedback page



Figure 15: Feedback page

**Description:** user and admin can enter feedback in feedback page for any book.

#### 5.1.4 Update book Management page

**Edit Book**

Name  
php development

Price  
660₹

Author  
Jon Duckett

Description  
You'll learn all about PHP which was used to build sites.

Volume  
1

Year  
2020

Status  
unavailable

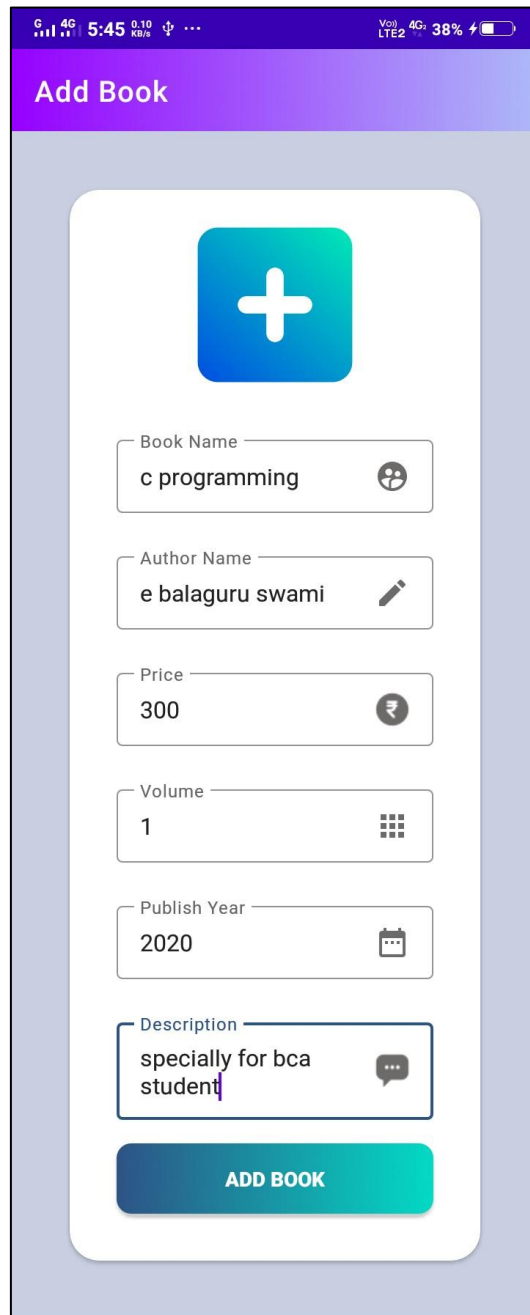
CANCEL UPDATE

Figure 16: Update book Management page

**Description:** user and admin can edit book information using book management page in reusebook.in future if any user wants to change existing information in book description, then using this page they can do it.



### 5.1.5 Add book page



**Add Book**

Book Name  
c programming

Author Name  
e balaguru swami

Price  
300

Volume  
1

Publish Year  
2020

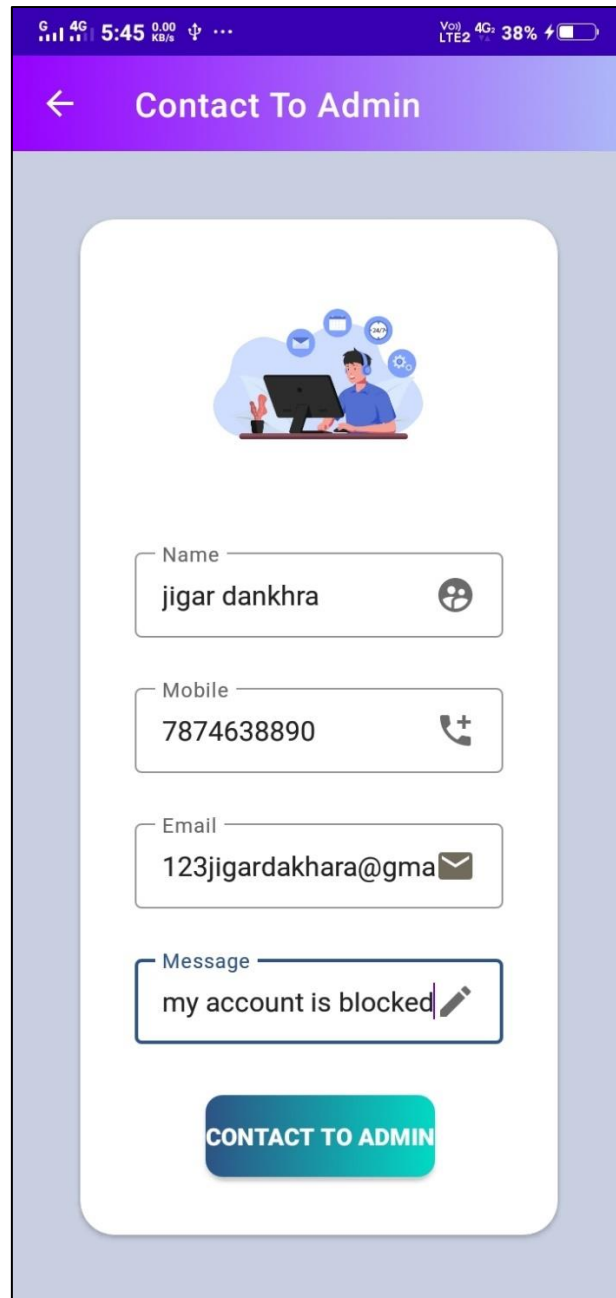
Description  
specially for bca student

**ADD BOOK**

Figure 17: Add book page


**Description:** user and admin can add book name, author name, price, volume, publish year and description of book using add book page too sell any book.

### 5.1.6 contact to admin page



4G 5:45 0.00 KB/s 4G+ 38%

← Contact To Admin



Name  
jigar dankhra

Mobile  
7874638890

Email  
123jigardakhara@gma

Message  
my account is blocked

CONTACT TO ADMIN

Figure 18: Contact to admin page

**Description:** if user have any query, then they can send a message to admin after filling name, mobile, email and message using this page.

### 5.1.7 Edit user page

The screenshot shows a mobile application interface for managing users. The top bar is purple with a back arrow, the text 'Manage Users', and a search icon. Below this is a white modal titled 'Edit User'. The modal contains six input fields, each with a label, a value, and an icon: 'Name' with 'admin' and a person icon, 'Email' with 'ramghumaliya10@gmail.com' and an envelope icon, 'Enrollment' with '210511211004' and a pencil icon, 'Contact no' with '9265713038' and a phone icon, 'Address' with 'mokar' and a house icon, and 'User Type' with 'admin' and a person icon. At the bottom of the modal are two buttons: 'CANCEL' in purple and 'UPDATE' in blue.

Figure 19: Edit user page

**Description:** admin can edit user information using edit user page. If admin wants to change any user to admin, then using user type they can do it for same.

## 5.2 Report Layout

### 5.2.1 Splash screen page



Figure 20: splash screen page

**Description:** this is a splash screen page .it will display at a starting of reusebook.

## 5.2.2 Edit user page



Figure 21: home &amp; search book page

**Description:** this is home page of reusebook, here user can search book and on click of any book user can view book description.

### 5.2.3 book detail page

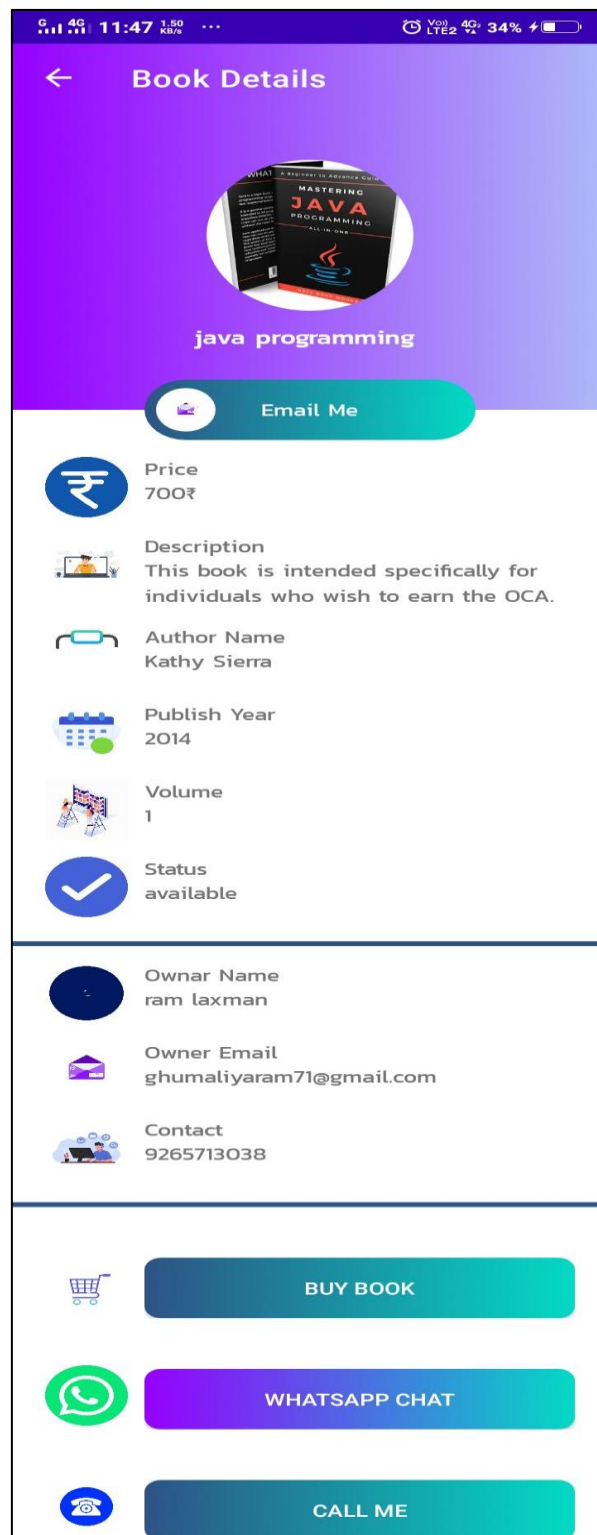


Figure 22: Book detail page

**Description:** this page provides all information related clicked book icon placed on home & search book page. If any user wants to send email, WhatsApp message or call to seller then they can do it using this page.

#### 5.2.4 Email of Book detail page



**Description:** after click of email me button of book detail page this page will be visible to user .user can write email to seller regarding purchasing of book.

### 5.2.5 WhatsApp of Book detail page

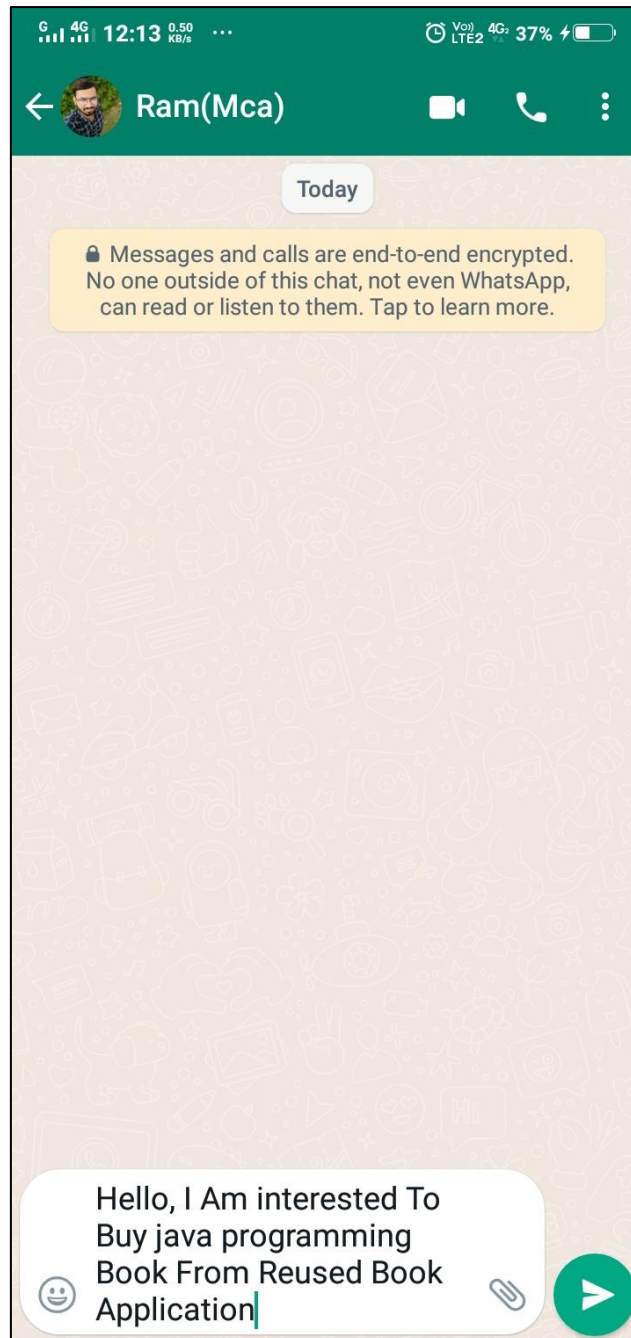


Figure 24: WhatsApp of Book detail page

**Description:** after click of WhatsApp chat button of book detail page this page will be visible to user .user can write a message to seller regarding purchasing of book.



### 5.2.6 Calling of book detail page

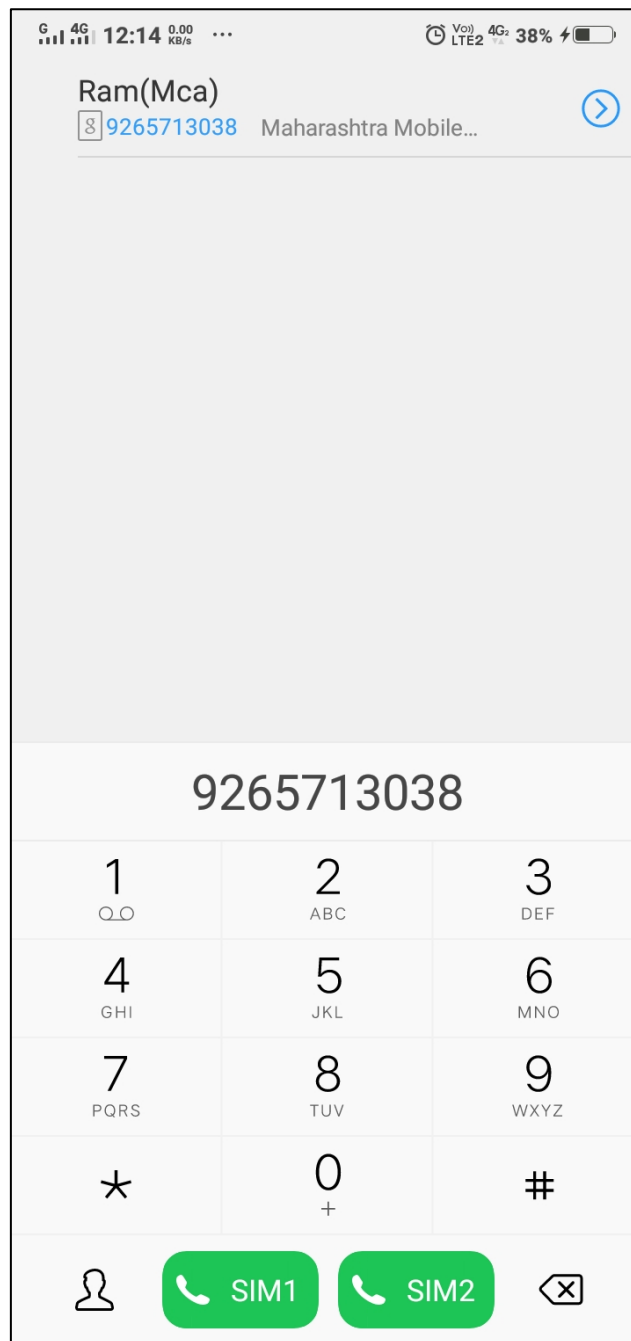


Figure 25: Calling of book detail page

**Description:** after click of call me button of book detail page this page will be visible to user .user can do call to seller regarding purchasing of book.

### 5.2.7 Menu bar page

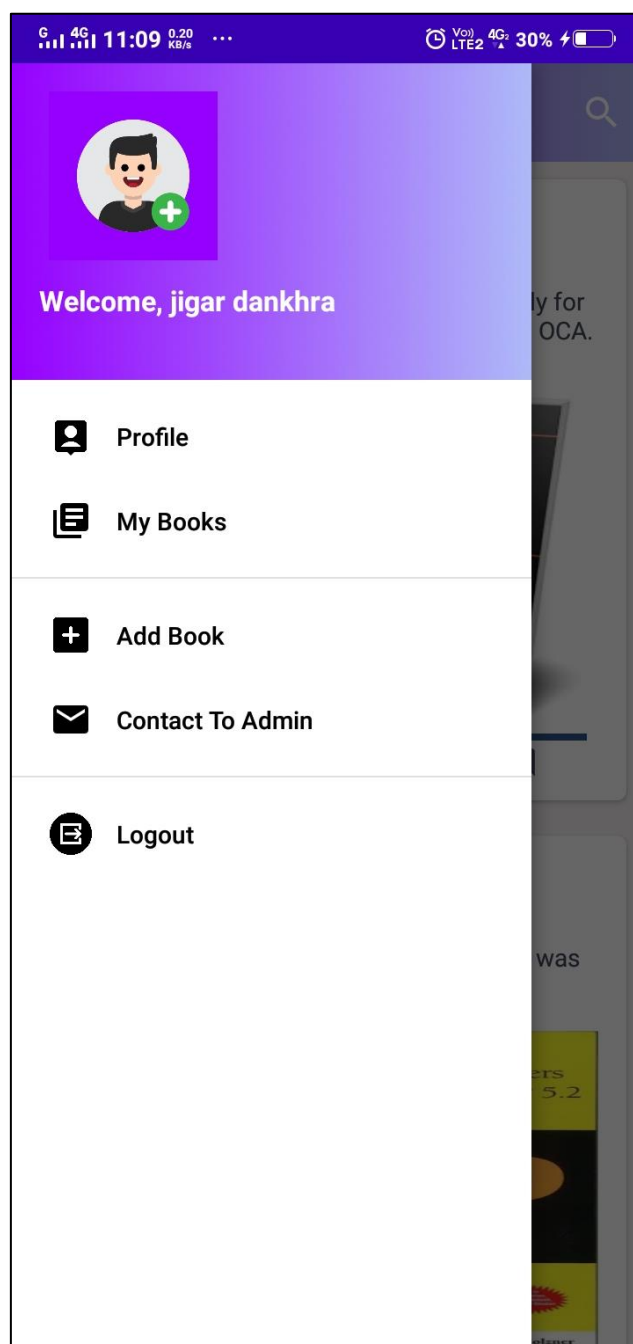


Figure 26: Menu bar page

**Description:** after click of menu button of home & search page this page will be visible to user .user can manage their profile, books, log out and contact with admin.

### 5.2.8 My profile page

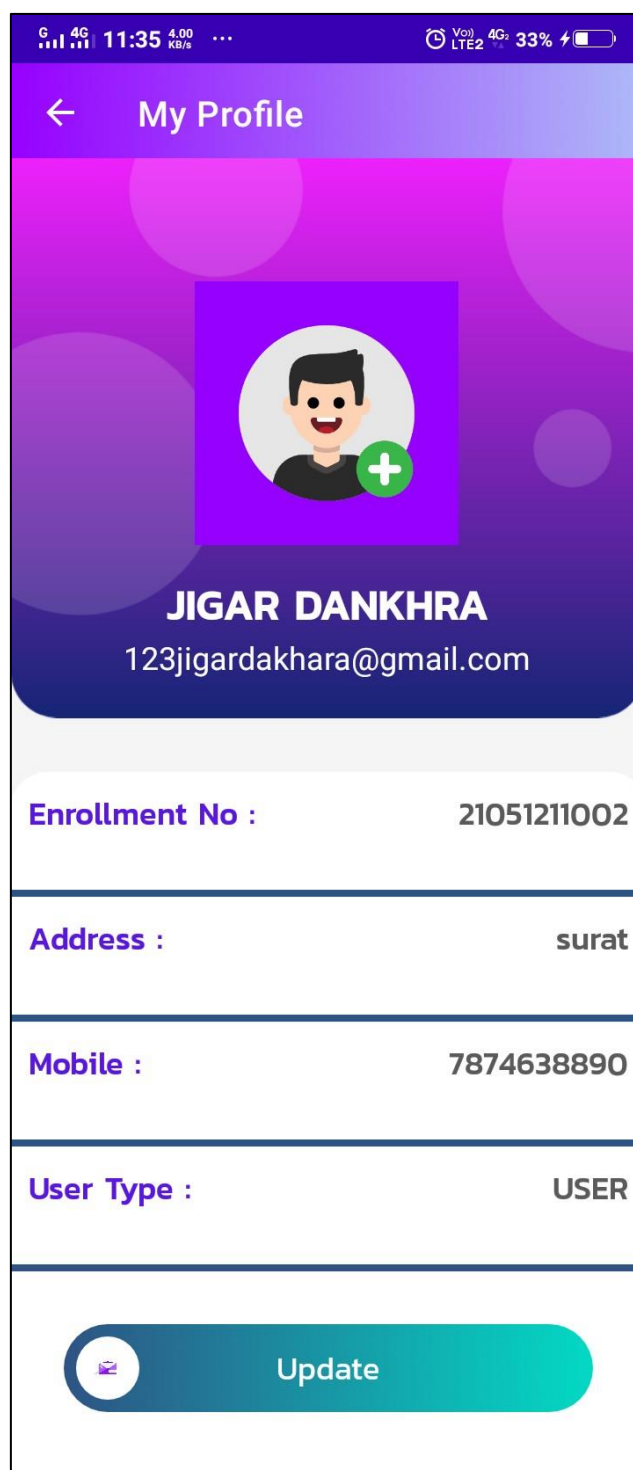


Figure 27: My profile page

**Description:** after click my profile button of menu bar page this page will be visible to user. User can see all details of their self-using this page. If any user wants to change their information, then using update button it can be possible for same.

### 5.2.9 Manage books page

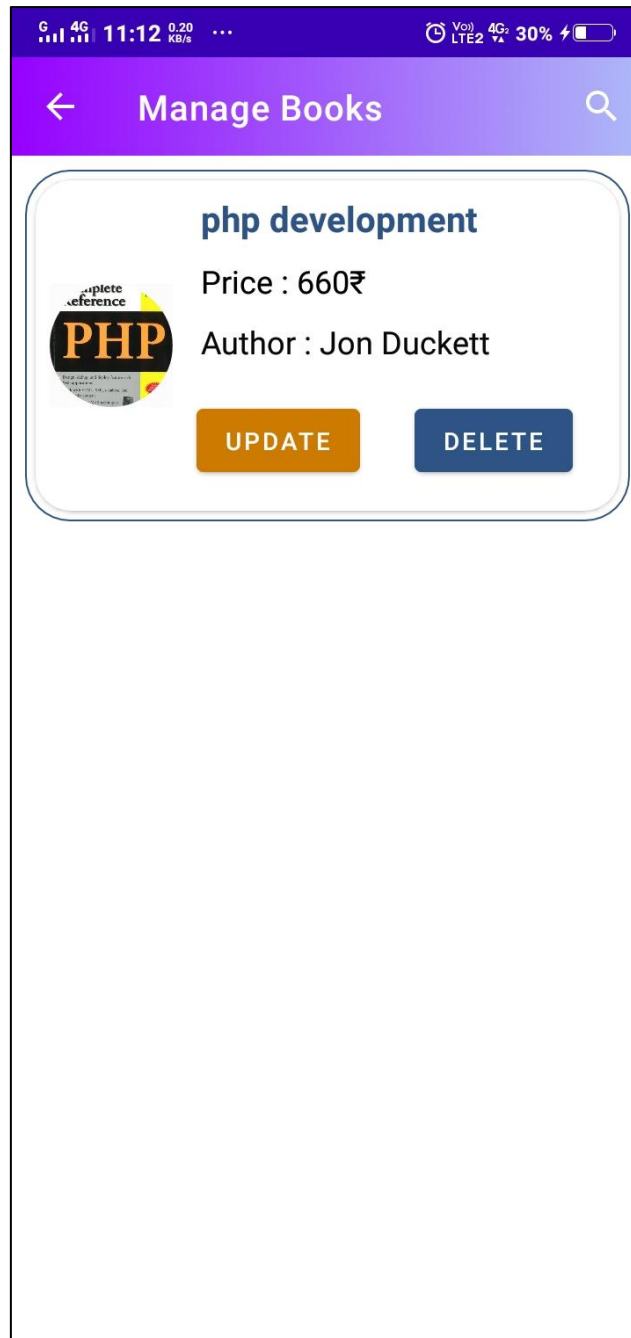


Figure 28: Manage books page

**Description:** after click my books button of menu bar page this page will be visible to user. User can see all details of their book using this page. If any user wants to change books information, then using update button it can be possible for same also user can delete a book using a delete button.

### 5.2.10 admin dashboard page

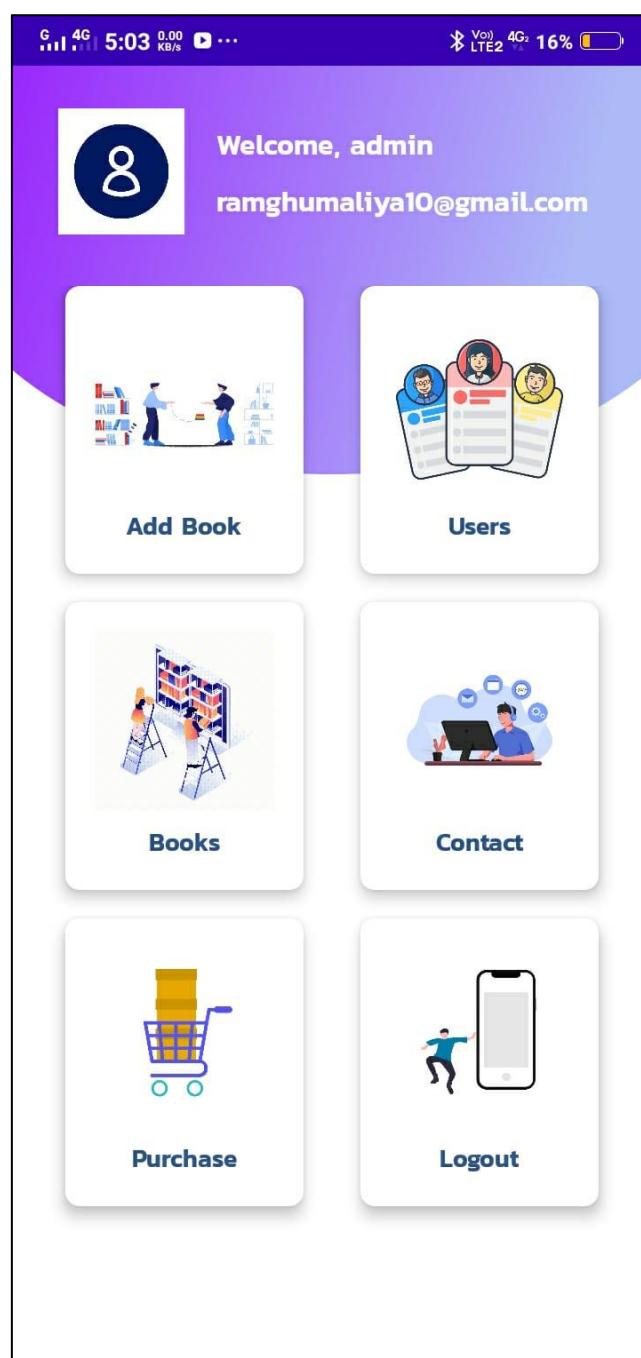


Figure 29: Admin dashboard page

**Description:** after login this page will be visible to admin. Admin can add book, maintain all users of system, books, contact with users, purchase request and logout.

### 5.2.11 Manage user page

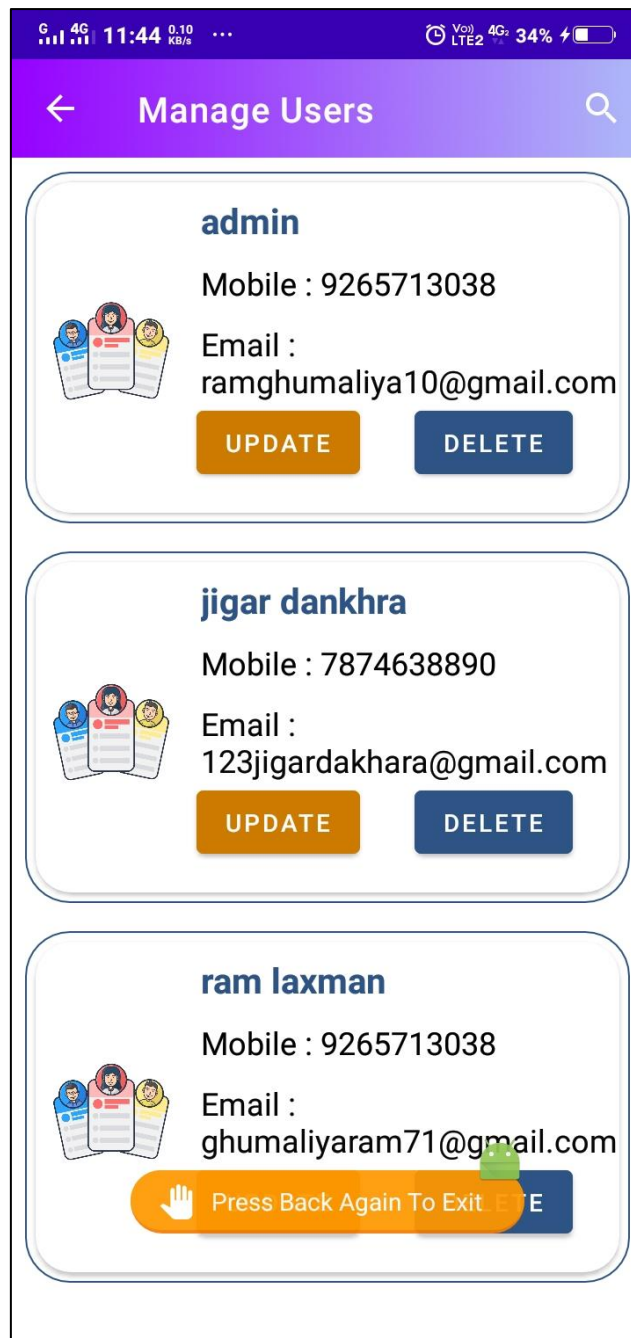


Figure 30: Manage user page

**Description:** after click on user card of admin dashboard page this page will be visible to admin. admin can see all users of system over here. if admin wants to update and delete details of users then it can be possible using update and delete button respectively.

### 5.2.11 Books option page

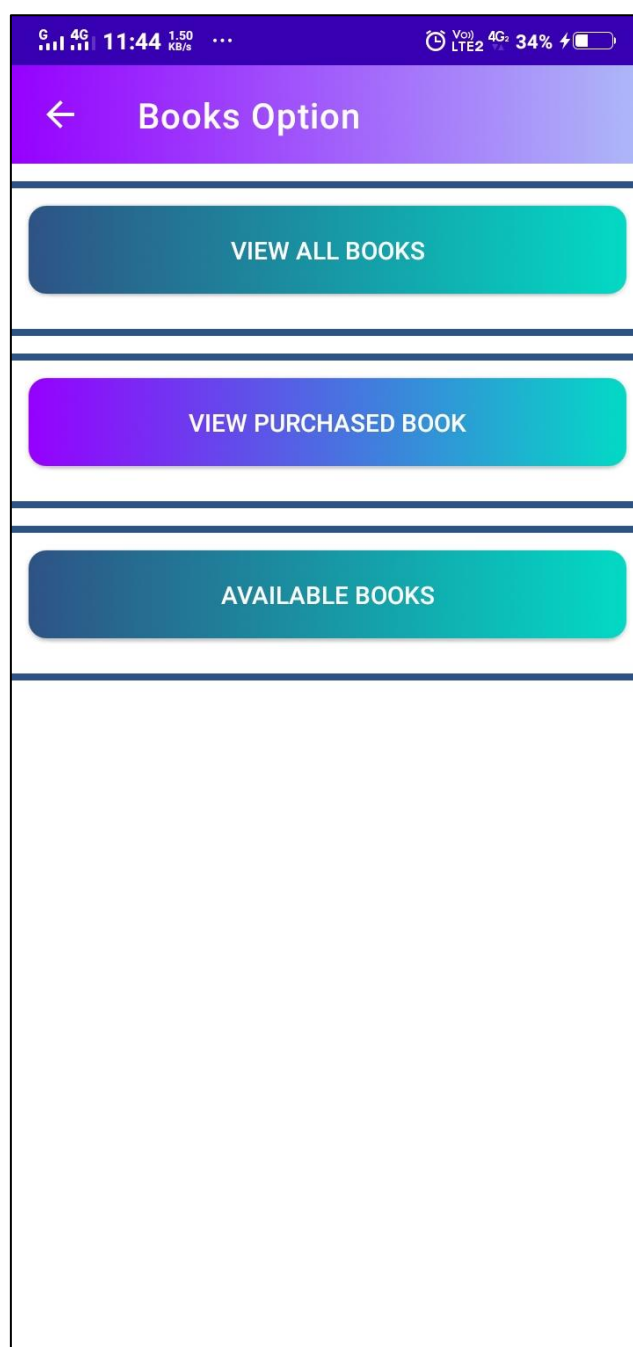


Figure 31: Book option page

**Description:** after click on books card of admin dashboard page this page will be visible to admin. here admin can check for 3 find of books: all books, purchased books and available books.

## 5.2.11 view all books page

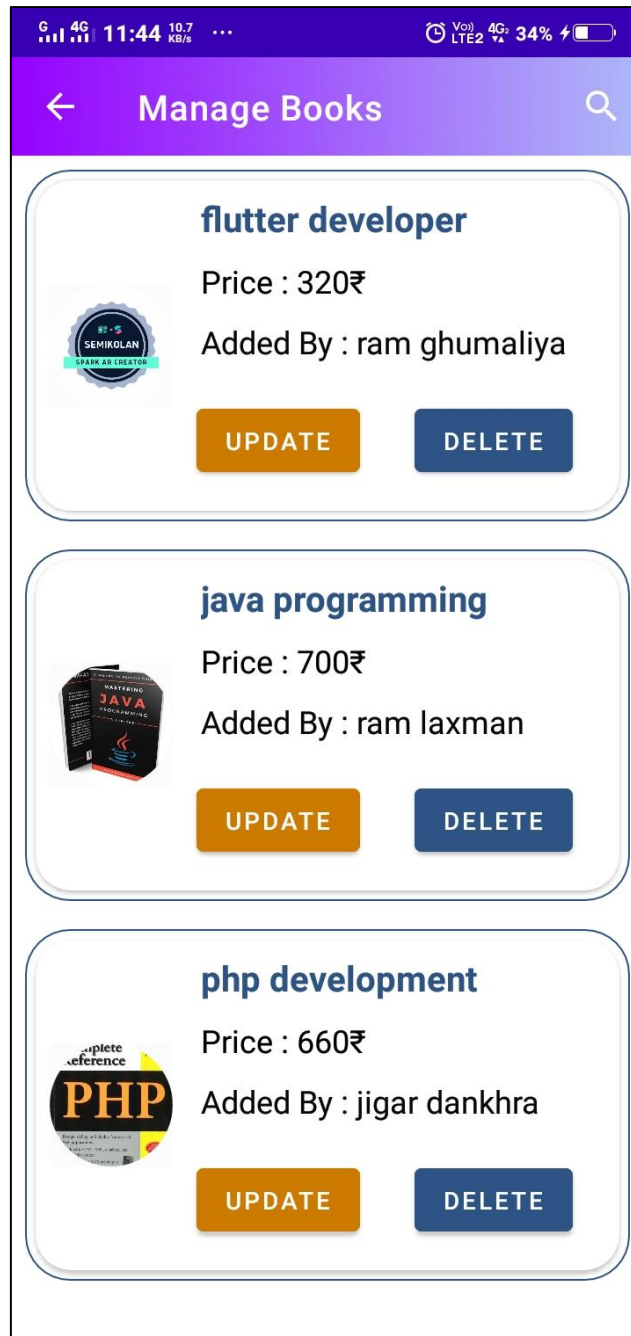


Figure 32: View all books page

**Description:** after click on view all books button of book option page this page will be visible to admin. here admin can check for all books and update and delete books information.



### 5.2.12 Purchased book page



Figure 33: Purchased book page

**Description:** after click on view purchased books button of book option page this page will be visible to admin. here admin can check for all purchased books, buyer name and seller name.

### 5.2.13 Available book page

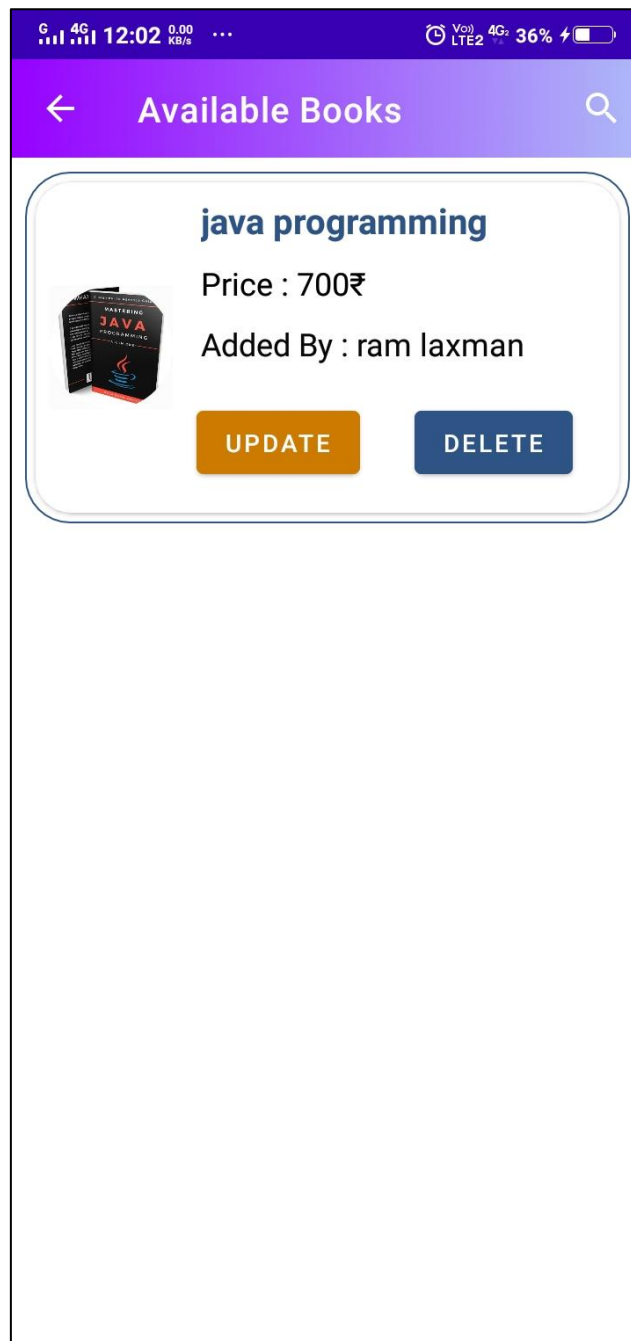


Figure 34: Available book page

**Description:** after click on available books button of book option page this page will be visible to admin. here admin can check for all available books and update and delete books information.

### 5.2.14 Manage contact page

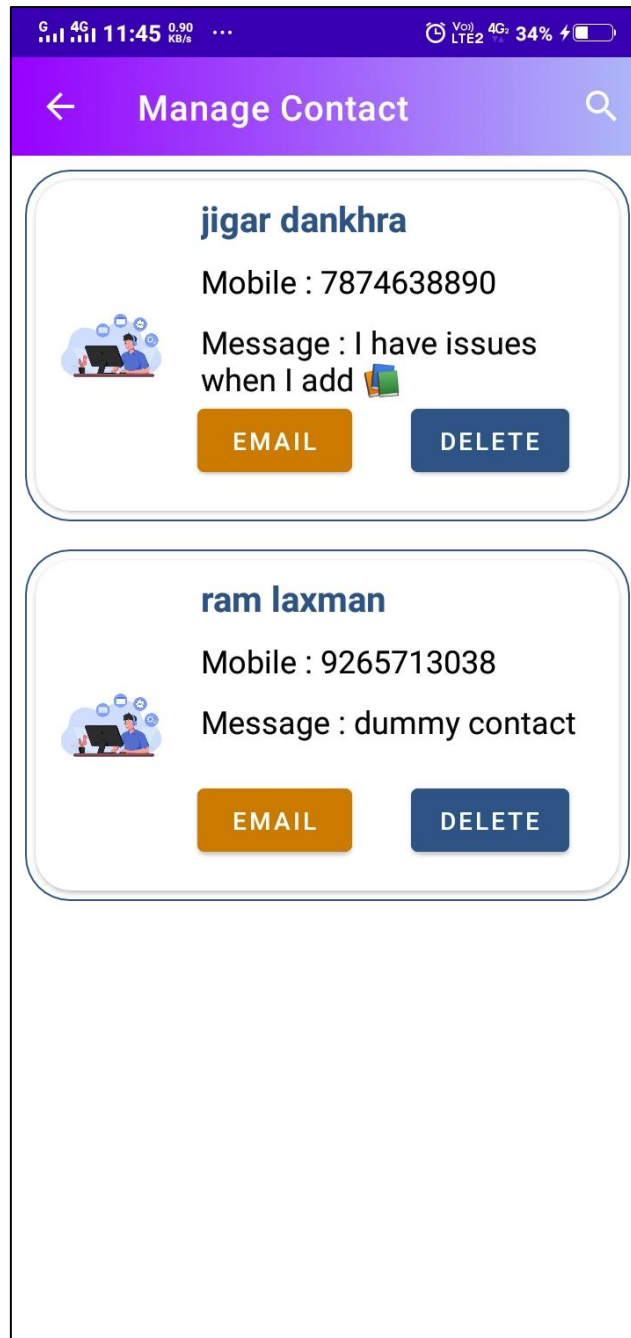


Figure 35: Manage contact page

**Description:** after click on contact card of admin dashboard page this page will be visible to admin. admin can see all query of users and they can send email to users as feedback and delete a message as a part of rejection of it.

### 5.2.15 Email in manage contact page

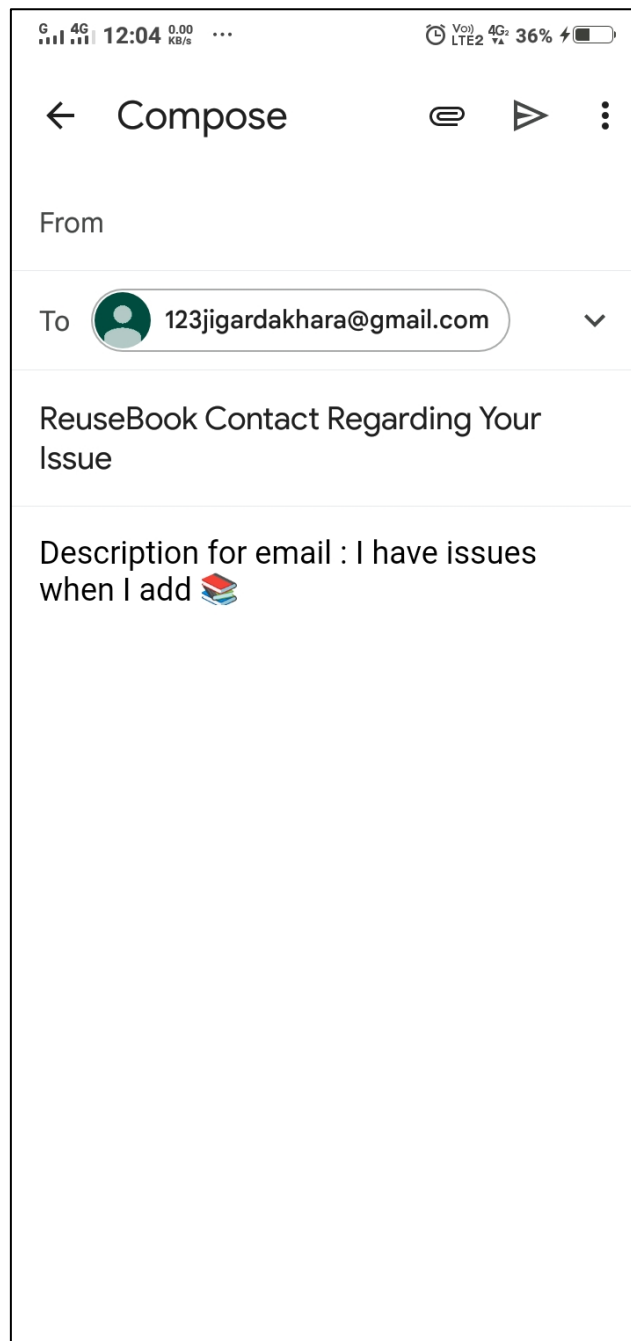


Figure 36: Email in manage contact page

**Description:** after click on email button of manage contact page this email page will be visible to admin.

### 5.2.16 Email in manage contact page

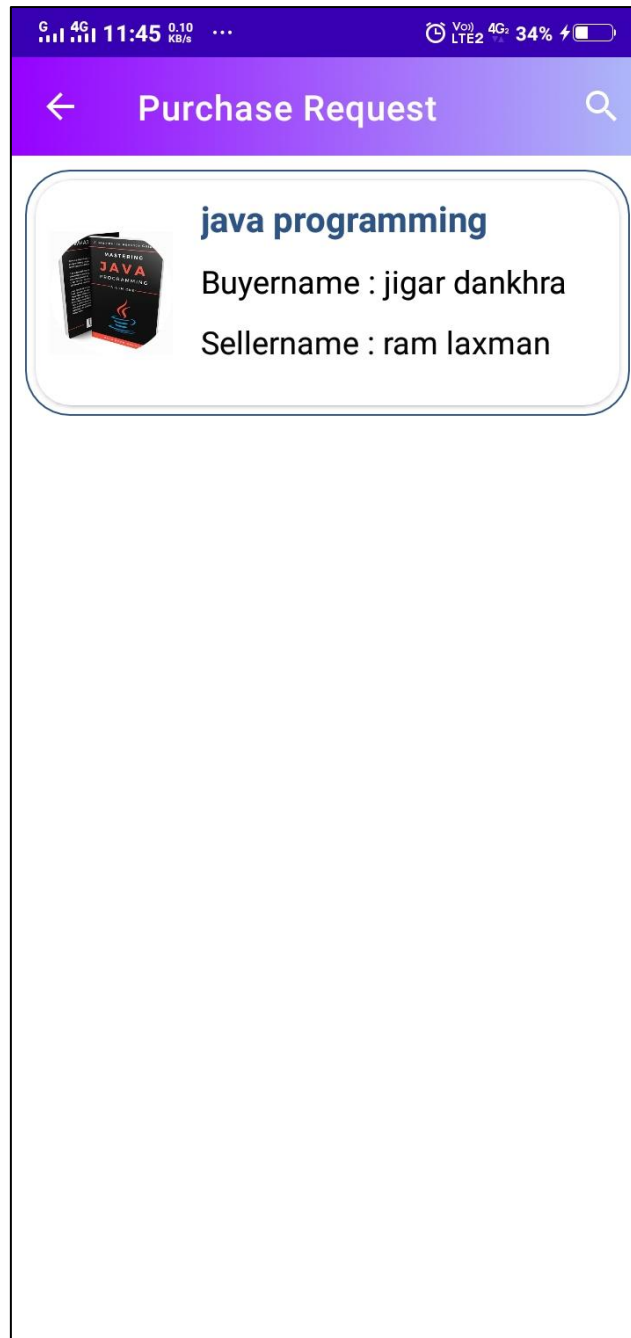


Figure 37: Purchase request page

**Description:** after click on purchase card of admin dashboard page this page will be visible to admin. admin can see all purchase request of users.

### 5.2.17 Purchase request book detail page

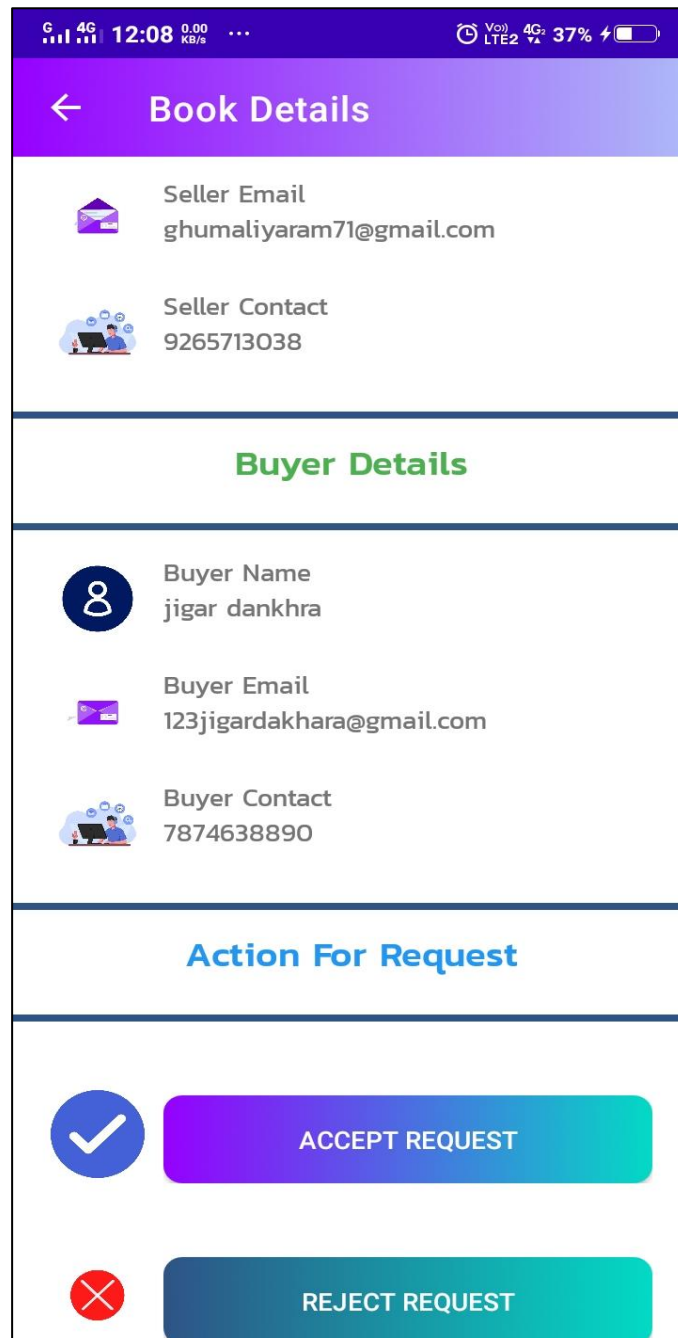


Figure 38: Purchase request book detail page

**Description:** after click on any book card of purchase request page this page will be visible to admin. here admin can check Book details, seller details, and buyer details of purchase request. Admin can accept request using accept request button and reject using reject request.

## 5.3 Coding Convention (Business Logic)

### 5.3.1 Add book

```

BookImageRef.child(mUser.getUid()+strDate).putFile(imageUri).addOnCompleteListener(new
OnCompleteListener<UploadTask.TaskSnapshot>() {

    @Override

    public void onComplete(@NonNull Task<UploadTask.TaskSnapshot> task) {

        if (task.isSuccessful())

        {

            BookImageRef.child(mUser.getUid()+strDate).getDownloadUrl().addOnSuccessListener(new
OnSuccessListener<Uri>() {

                @Override

                public void onSuccess(Uri uri) {

                    String currentTime = new SimpleDateFormat("HH:mm:ss", Locale.getDefault()).format(new Date());

                    HashMap hashMap=new HashMap();

                    hashMap.put("Date",strDate);

                    hashMap.put("PostImageUrl",uri.toString());

                    hashMap.put("bookname",txtinputbName.toLowerCase());

                    hashMap.put("bookauthor",txtinputbAuthor);

                    hashMap.put("bookprice",txtinputbPrice+"₹");

                    hashMap.put("bookvolume",txtinputbvolume);

                    hashMap.put("bookpublishyear",txtinputbpublishyear);

                    hashMap.put("status","available");

                    hashMap.put("uname",name);

                    hashMap.put("bookdescription",txtinputbDescription);

                    hashMap.put("uid",mUser.getUid());

                    hashMap.put("bookid",mUser.getUid()+strDate);

                    hashMap.put("sort",countPost+1);

                    BookRef.child(mUser.getUid()+strDate).updateChildren(hashMap).addOnCompleteListener(new
OnCompleteListener() {

                        @Override

                        public void onComplete(@NonNull Task task) {

                            if (task.isSuccessful())

```

```

        {
            mLoadingBar.dismiss();

            FancyToast.makeText(AddBookActivity.this,"Book Added Successful",FancyToast.LENGTH_LONG,
FancyToast.SUCCESS,true).show();

            book_image.setImageResource(R.drawable.addbook);//this will Set default image from
addImagePost button

            inputbName.getEditText().getText().clear();
            inputbAuthor.getEditText().getText().clear();
            inputbPrice.getEditText().getText().clear();
            inputbvolume.getEditText().getText().clear();
            inputbpublishyear.getEditText().getText().clear();
            inputbDescription.getEditText().getText().clear();
        }
        else
        {
            mLoadingBar.dismiss();

            FancyToast.makeText(AddBookActivity.this,"Error :
"+task.getException().toString(),FancyToast.LENGTH_LONG, FancyToast.SUCCESS,true).show();
        }
    }
    });
}
});
}
else
{
    mLoadingBar.dismiss();

    FancyToast.makeText(AddBookActivity.this,"Error : "+task.getException().toString(),FancyToast.LENGTH_LONG,
FancyToast.SUCCESS,true).show();
}
}
});

```

- This code provides a logic for adding books into system. When any user wants to add any old book to resell that time this code will be execute and add old book into a system.



### 5.3.2 WhatsApp communication API

```
btnwhatsapp.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        ClickUserRef.addValueEventListener(new ValueEventListener() {
            @Override
            public void onDataChange(@NonNull DataSnapshot snapshot) {
                if (snapshot.exists()) {

                    String ccontact = snapshot.child("contact").getValue().toString();

                    try {
                        String url = "https://api.whatsapp.com/send?phone="+91" + ccontact + "&text=" + URLEncoder.encode("Hello, I
Am interested To Buy " + bookname + " Book From Reused Book Application", "UTF-8");
                        Intent i = new Intent(Intent.ACTION_VIEW);
                        i.setData(Uri.parse(url));
                        startActivity(i);
                    } catch (Exception e) {
                        Toast.makeText(BookDetailsActivity.this, "Error" + e, Toast.LENGTH_SHORT).show();
                    }

                } else {
                    Toast.makeText(BookDetailsActivity.this, "Data Not Found Trt Again....", Toast.LENGTH_SHORT).show();
                }
            }

            @Override
            public void onCancelled(@NonNull DatabaseError error) {
            }
        });
    }
});
```

- This code provides a facility that one user(buyer) can do chat with another user(seller) when he/she interested to purchase old book.

### 5.3.3 Calling API

```
btncall.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        ClickUserRef.addValueEventListener(new ValueEventListener() {
            @Override
            public void onDataChange(@NonNull DataSnapshot snapshot) {
                if (snapshot.exists()) {

                    String ccontact = snapshot.child("contact").getValue().toString();

                    Intent intent = new Intent(Intent.ACTION_DIAL);
```

```

        intent.setData(Uri.parse("tel:" + ccontact));
        startActivity(intent);
    } else {
        Toast.makeText(BookDetailsActivity.this, "Data Not Found Trt Again....", Toast.LENGTH_SHORT).show();
    }
}

@Override
public void onCancelled(@NonNull DatabaseError error) {
}

});
}
});

```

- This code provides a facility that one user(buyer) can do call to another user(seller) when he/she interested to purchase old book.

### 5.3.4 Purchase request approval by admin

```

btnaccept.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view)
    {
        AlertDialog.Builder builder = new AlertDialog.Builder(BuyBookDetailsActivity.this);
        builder.setTitle("Confirm");
        builder.setMessage("Are You Sure You Really Want To Accept This Book Buy Request");
        builder.setCancelable(false);
        builder.setIcon(R.drawable.ic_warning);
        builder.setPositiveButton("Yes", new DialogInterface.OnClickListener() {
            @Override
            public void onClick(DialogInterface dialog, int which)
            {
                ClickBookRef.child("status").setValue("success");
                BookRef.child("status").setValue("unavailable");
                SendMail mail = new SendMail("ramghumaliya10@gmail.com", "yyidbhqjmqrogecx",
                    buemail,
                    "Your Book Buy Request Accepted Successfull!...",
                    " Dear,"+buname+" \n\n Your "+bookname+" Book Buy Request Accepted By Admin !...");
                mail.execute();

                SendMail mail1 = new SendMail("ramghumaliya10@gmail.com", "yyidbhqjmqrogecx",
                    seemail,
                    "You Have Accepted "+bookname+" Book Buy Request ",
                    " Dear,"+sename+" \n\n You Have Accepted "+bookname+" Book Buy Request ");
                mail1.execute();
                SendUserToMainActivity();
            }
        });
    }
});

```

```
FancyToast.makeText(BuyBookDetailsActivity.this,"Book Buy Request Accepted Successfull!...Admin Will contact You",FancyToast.LENGTH_SHORT, FancyToast.SUCCESS,true).show();
    SendUserToMainActivity();
}

}).setNegativeButton("No", new DialogInterface.OnClickListener() {
    @Override
    public void onClick(DialogInterface dialog, int which) {
        dialog.cancel();
    }
});
AlertDialog alert = builder.create();
alert.show();
}
});
```

- when any user sends a request to admin for purchasing any book. then that time admin has to accept that request if book is available.so this code provides a facility that admin can accept a request of any user.

## 6. Testing

### 6.1 Test Strategy

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub-assemblies, assemblies and/or a finished product. It is the process of exercising software with the intent of ensuring that the software is working properly or not.

Here this application has many fields so for that field testing is going to apply. for every field at back end some patterns are used to validate input which is given by user. so basically field validator validates a user input and using it application can take valid data from user.

Here for this application manual testing can be used. it has many benefits as mentioned below: -

- manual testing is very easy to understand.
- Human brain is irreplaceable. Only human can create valid test design, foresee defects, and execute designs to get the expected result. Human testers will quickly detect when something looks “Off” and can anticipate other failures that sometimes go unnoticed during automation testing. There is no replacement for the human eye.
- It is not possible to do the UX/UI testing through automation. A manual tester can spot contextual, functional, visual, and usability bugs that automation scripts may not identify.
- There are still some domains that prefer Manual testing over Automation testing for most of the system functionality.
- Several times, situations arise when there are runtime variations in the functionality of some components either as an enhancement or behavioural change. In this case, the time and knowledge of functionality play a major role.
- To optimize the price and time needed to test any software, tester cannot entirely rely upon the automation testing approach. Tester needs the right combination of manual testing and automatic testing to get error-free software.

**6.1.1 field validation:** -application has field validation to check whether input is valid or not. So, at every field validation is given by developer using regular expression.

**6.1.2 Rang validation:** -rang validation can give more support to restrict invalid input so it is also done in this application.

**6.1.3 UI/UX testing:** -UI/UX is a very important part of testing so this testing is also performed by developer and some candidates for testing purpose.

**6.1.4 required field validation:** -without filling of all fields it is not possible submission of form so required field validation provides a feature that application must have to take all required data.

**6.1.5 code coverage validation:** -in code coverage validation we have tested all statement, branch and loop part of code. because it is very important that if we have written anything then it must have some work. Useless code can give load to our system only.

**6.1.6 unit testing:** -at the completion of every module development developer had checked all possibilities of errors.so unit testing is also done by developer.

**6.1.7 integration testing:** -after unit testing one by one tester have taken all modules and done integration testing.

**6.1.8 system testing:** -at the end of development fully system is tested by developer and some selected testers.in system testing all the validation and other features has been checked by testers and working perfectly as per requirement.

**6.1.9 beta testing:** -for beta testing with the help of some selected people application has been checked.

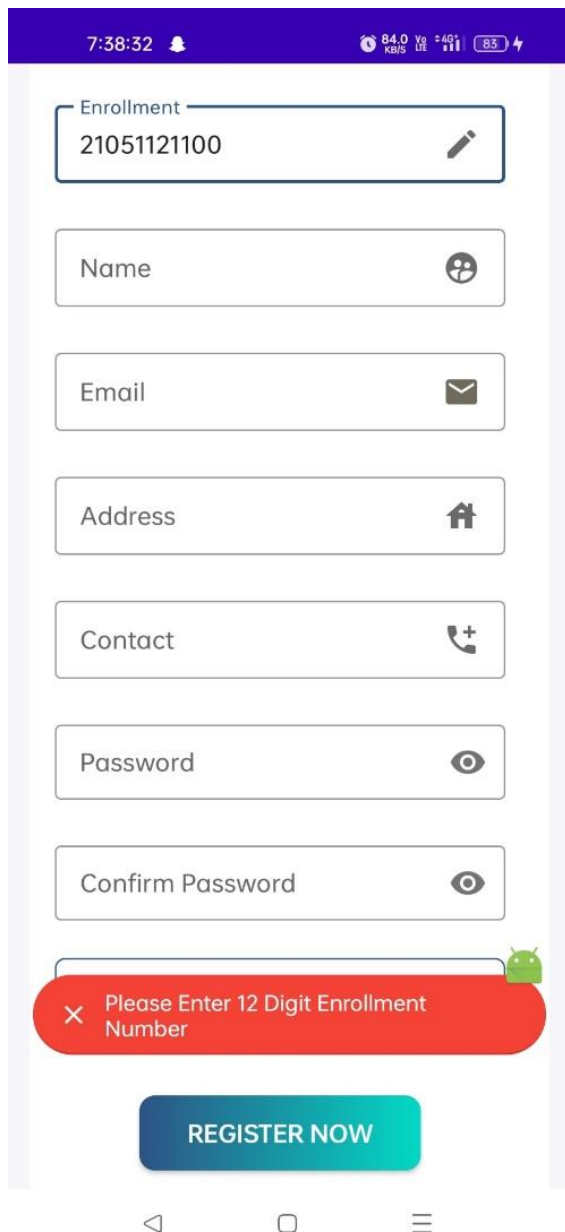
**6.1.10 alpha testing:** -developer has checked all features of the system for alpha testing and it is working very well .

## 6.2 Test Cases

### 6.2.1 Module 1: login & signup

Test Case Scenario ID:		1	Test case ID:		1	
Test Case Description:		To test Login and Signup module	Test Priority:		HIGH	
Pre-requisite:		NA	Post-requisite:		User should get logged in into the system	
Test Execution Step:		User needs to give email id/password when it is provided when ask for registration				
Sr. No	Test Action	Test Input	Browser	Expected Result	Actual Result	Pass/ Fail
1.1	1. Enter email id 2. Enter Password 3. Click on Login button	Email id entered Password entered Login clicked	Any Browser	User gets logged in	As Expected,	Pass
1.2	1. Enter Username 2. Enter Password 3. Click on Login button	Wrong email id entered Wrong Password entered Login clicked	Any Browser	User should not get logged in	As Expected,	Pass
1.3	1. Enter enrol. no 2. Enter Name 3. Enter Email 4. Enter address 5. Enter Password Confirm Password	Enrol. no entered Last Name entered Email entered address entered Password entered Password confirmed	Any Browser	User should get signed up	As Expected,	Pass
1.4	1. Enter enrol. no 2. Enter Name 3. Enter Email 4. Enter address 5. Enter Password 6. Confirm Password	Enrol. no entered Last Name entered Email entered address entered Password entered Password confirmed	Any Browser	Error message for wrong email, password and contact format	As Expected,	Pass

Figure 15: Test case for login & sign up



7:38:32 84.0 KB/s 54% 85%

Enrollment  
21051121100

Name

Email

Address

Contact

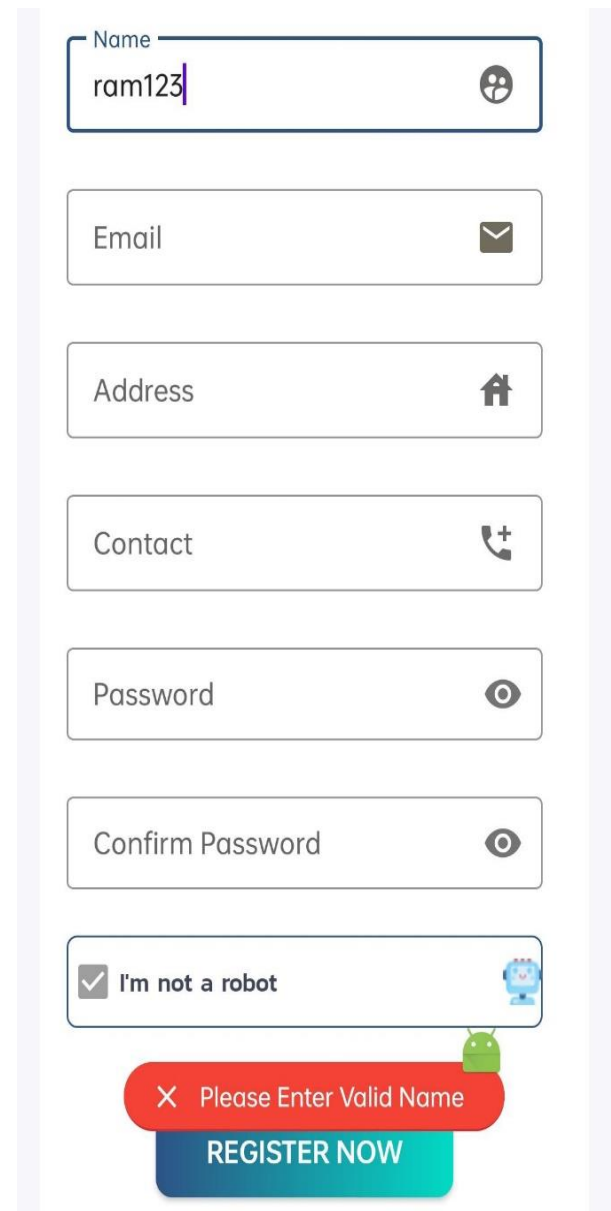
Password

Confirm Password

× Please Enter 12 Digit Enrollment Number

REGISTER NOW

Figure 39: enrolment validation testing



Name  
ram123

Email

Address

Contact

Password

Confirm Password

☒ I'm not a robot

× Please Enter Valid Name

REGISTER NOW

Figure 40: name validation testing

Figure 41 shows a mobile app registration screen. The 'Contact' field contains the number '94979797979'. Below it, the 'Password' and 'Confirm Password' fields are empty. The 'I'm not a robot' checkbox is checked. A red error message 'X Please Enter 10 Digit Contact Number' is displayed above the 'REGISTER NOW' button. The screen also includes a 'Log In' link and a bottom navigation bar with back, home, and menu icons.

Figure 41: contact no validation testing

Figure 42 shows a mobile app registration screen. The 'Password' field contains '71717'. Below it, the 'Confirm Password' field is empty. The 'I'm not a robot' checkbox is checked. A red error message 'X Password Must Be 6 Latter Long' is displayed above the 'REGISTER NOW' button. The screen also includes a 'Log In' link and a bottom navigation bar with back, home, and menu icons.

Figure 42: password validation testing



## 6.2.2 Module 2: Navigation and basic features

Test Case Scenario ID:		2	Test case ID:		2	
Test Case Description:		Test of different banners and buttons that navigation to new pages	Test Priority:		HIGH	
Pre-requisite:		User should be logged in	Post-requisite:		User should get logged in into the system	
Test Execution Step:		User should get logged in into the system to Test of different banners and buttons that navigation to new pages				
Sr.No	Test Action	Test Input	Browser	Expected Result	Actual Result	Pass/Fail
2.1	1.click on the book Images	Latest book Item Clicked	Any Browser	The book description page should open	As Expected,	Pass
2.2	1.click on feedback icon	Clicked on the feedback icon	Any Browser	The feedback page should open	As Expected,	Pass
2.3	1.click on navigation button	Clicked on see more	Any Browser	More items of the category should open	As Expected,	Pass
2.4	1.select menu 2.select a sub menu	Menu selected for events	Any Browser	The user should be navigated to events window	As Expected,	Pass

Table.16: Test Case of Navigation and Basic features

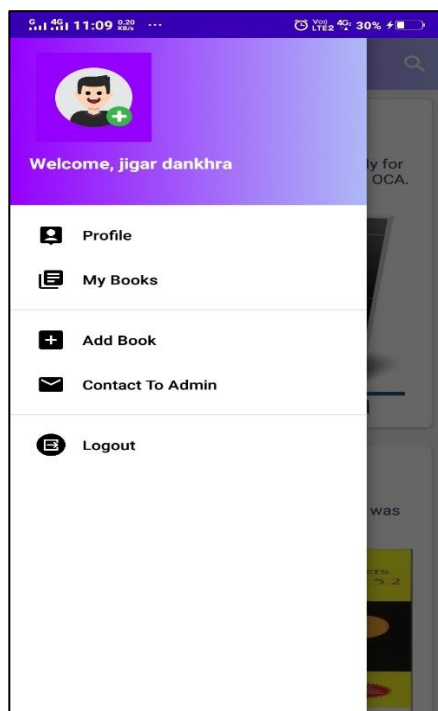


Figure 43: menu bar navigation testing



Figure 44: book image navigation testing

## 6.2.3 Module 3: Book management

<b>Test Case Scenario ID:</b>	<b>3</b>	<b>Test case ID:</b>	<b>3</b>
<b>Test Case Description:</b>	Test of add book for sale and addition/deletion of items	<b>Test Priority:</b>	Medium
<b>Pre-requisite:</b>	The user should be logged in	<b>Post-requisite:</b>	User should be able to add book and operate it
<b>Test Execution Step:</b>	User should be able to add book and addition/deletion of items		

Sr.No	Test Action	Test Input	Browser	Expected Result	Actual Result	Pass/Fail
3.1	1. Book details 2. Click on Add book Check in the my books	Entered book detail Add book clicked My checked	Any Browser	book should be there in my books	As Expected,	Pass
3.2	1. Go to my books 2. Select book 3. Click on delete	Navigated to my book book selected Delete clicked	Any Browser	book should be deleted from my books	As Expected,	Pass
3.3	1. Go to my books 2. Select book 3. Click on update	Navigated to my book book selected update clicked	Any Browser	Book information should be updated from my books	As Expected,	Pass

Table.17 Test Case of add book



Figure 45: Add book page

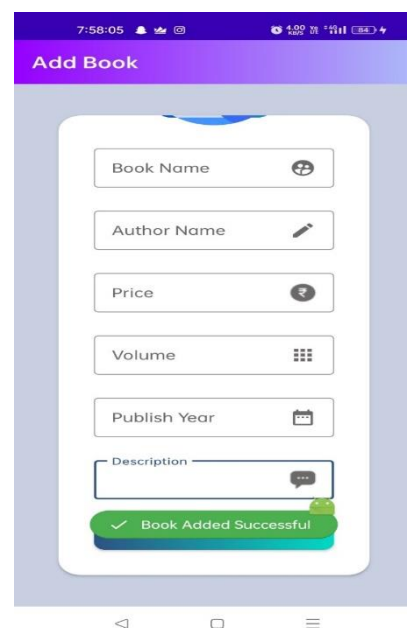


Figure 46: successfully book added testing

## 6.2.4 module 4: Contact to admin

<b>Test Case Scenario ID:</b>		<b>4</b>		<b>Test case ID:</b>	<b>4</b>	
<b>Test Case Description:</b>		Test of contact to admin, for user can communicate with admin		<b>Test Priority:</b>	High	
<b>Pre-requisite:</b>		The user should have logged in		<b>Postrequisite:</b>	User should be able to navigate to contact to admin page	
<b>Test Execution Step:</b>		User should be able to navigate to contact to admin page to have message of success				
Sr.No	Test Action	Test Input	Browser	Expected Result	Actual Result	Pass/ Fail
4.1	1. Go to menu bar Click on contact to admin button and Enter message for admin	Navigated to contact to admin page Enter message Clicked on button	Any Browser	User should get message box	As Expected,	Pass
4.2	1. Go to menu bar Click on contact to admin button and Enter message for admin	Navigated to contact to admin page Clicked on button	Any Browser	The data should be not submitted and give message of missing data	As Expected,	Pass

Table.18: test case for contact to admin

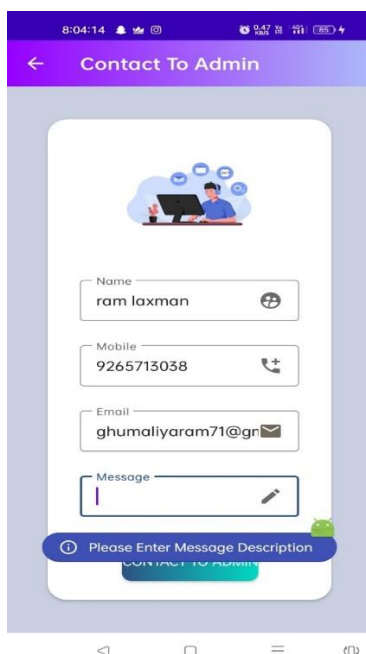


Figure 47: Contact to admin test page

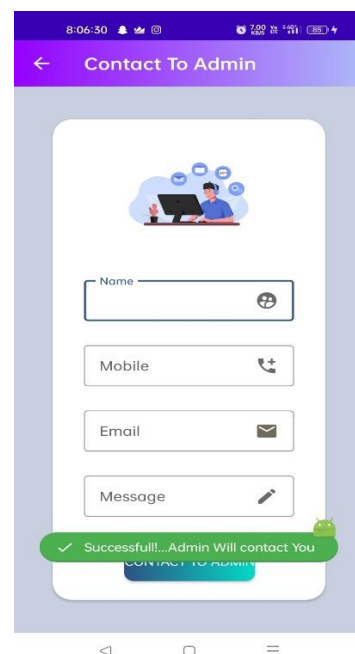


Figure 48: successfully contact with admin test

## 7. Future Enhancement

For future enhancement we would like to add more costly and study related things. Different Payment gateways, different reports. And save history or recently visited books in our application. This application can be compatible with IOS and android both as it is developed using .net core and angular which supports cross platforms.

In this application we can add map facility to reach at seller place to have a look for book. university can take little profit from this application after charge on every transaction. We can add online auction to sell a book.

## 8. Bibliography

### 8.1 Website and documentation

1. <https://developer.android.com/docs>
2. <https://firebase.google.com/docs>
3. <https://www.tutlane.com/tutorial/android>
4. <https://www.javatpoint.com/android-tutorial>

### 8.2 Book

1. “Android Application Development All in one for Dummies” by Barry Burd.
2. “Teach Yourself Android Application Development In 24 Hours”, Sams teach yourself, 3rd edition.
3. “Mobile Apps Development” by Anubhav Pradhan, Anil V Deshpande.
4. “Android Wireless Application Development” by Lauren Darcey and Shane Conder, Pearson Education, 2nd Edition, (2011).
5. “Professional Android 2.0 Application Development” by Reto Meier, Wiley India Pvt. Ltd., (2011).

### 8.3 Video tutorial

1. <https://www.youtube.com/c/CodingCafe>
2. <https://nullclass.com/project/learn-to-build-real-time-video-conferencing-app-like-zoom/>
3. <https://www.youtube.com/c/PapayaCoders>
4. <https://www.youtube.com/c/AndroidMasterApp>

### 8.4 API and library

1. <https://github.com/1902shubh/SendMail>
2. <https://api.whatsapp.com/send>