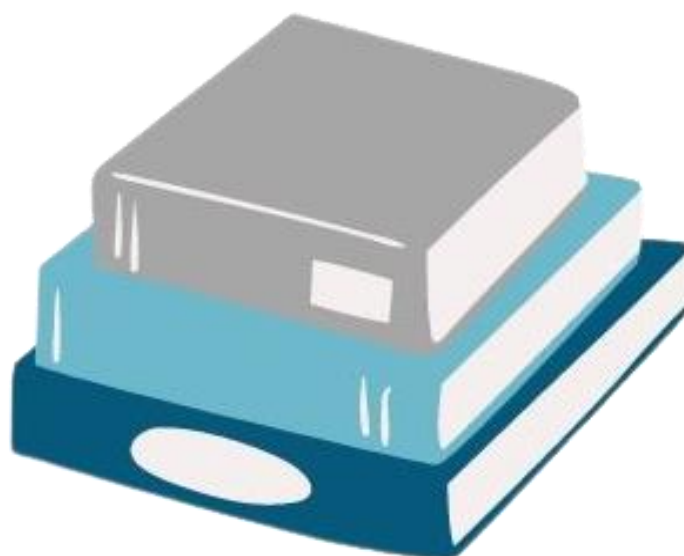




كلية الحاسبات والذكاء الاصطناعي  
Faculty of Computers & Artificial Intelligence



# RECME

*GET AND DISCOVER*

## [Recme Mobile Application]

Graduation Project 2021



*Faculty of Computers and Artificial Intelligence  
Information Systems Department*

**Supervised by:**  
**Dr. Laila Abdel-Hamed**

---

### Team Members



**Mohamed Abdel Nasser**



**Asmaa Mohamed**



**Mohamed Tarek**



**Mohamed Kamal**



**Mohamed Gamal**

## ACKNOWLEDGEMENT

First and foremost, we would like to express our great gratitude and immense joy at having the opportunity to work with such a dedicated person as **Dr. Laila Abd-El Hamed** for her support, outstanding guidance, and encouragement throughout our graduation project.

As well as **Eng. Amr Essam** for his continuous support and help as he was always there for us providing more than the needed time and effort from him since the very beginning of our project.

We would also like to thank our families, especially our parents, for their encouragement, patience, and assistance over the years. We are forever indebted to our parents, who have always kept us in their prayers and pushed us to Success.

Finally, for our faculty for providing the suitable environment that led us to represent the best image that computer science graduates of Helwan University are supposed to represent.

## Table of Contents:

<b>Chapter 1: Introduction.....</b>	<b>8</b>
1.1 Project Overview.....	9
1.2 Objectives.....	9
1.3 Purpose.....	9
1.4 Scope.....	10
1.5 General Constraints.....	12
 <b>Chapter 2: Project “Planning and Analysis” .....</b>	<b>14</b>
2.1 Project Planning.....	15
2.1.1 Feasibility study.....	15
2.1.2 Estimated Cost.....	16
2.1.3 Gantt Chart .....	17
2.2 Analysis and Limitations of existing system.....	18
2.3 Need for the new system .....	20
2.4 Analysis of the new system.....	21
2.4.1 User Requirements .....	21
2.4.2 System Requirements .....	21
2.4.3 Domain Requirements .....	22
2.4.4 Functional Requirements .....	23
2.4.5 Non-Functional Requirements .....	32
2.5 Advantages of the new system .....	33
2.6 Risk and Risk Managements .....	34

<b>Chapter 3: Software Design .....</b>	<b>37</b>
3.1 Design of Database .....	38
3.2 Class Diagram .....	39
3.3 Use-case Diagram.....	40
3.4 Sequence Diagram.....	41
3.5 Activity Diagram.....	49
 <b>Chapter 4: Implementation .....</b>	 <b>51</b>
4.1 Software Architecture .....	52
4.2 Flowchart .....	62
 <b>Chapter 5: Testing .....</b>	 <b>65</b>
5.1 Unit Testing .....	66
5.2 Integrated Testing .....	66
5.3 Additional Testing .....	68
 <b>Chapter 6: Results and Discussion.....</b>	 <b>90</b>
6.1 Results .....	91
6.1.1 Expected Results .....	91
6.1.2 Actual Results.....	91
6.2 Discussion .....	91
 <b>Chapter 7: Conclusion .....</b>	 <b>92</b>
 <b>Chapter 8: Future Work.....</b>	 <b>93</b>
 <b>Bibliography.....</b>	 <b>95</b>

## List of Tables

- Table 1: Competitors ..... Page # 20
- Table 2: Risk and Risk Management..... Page # 34
- Table 3: Test Cases..... Page # 66

## List of Figures

- Figure 1: Gantt Chart .....Page # 17
- Figure 2: Design of Database .....Page # 38
- Figure 3: Class Diagram.....Page # 39
- Figure 4: Use-case Diagram ..... Page # 40
- Figure 5: Sequence Diagram ..... Page # 41
- Figure 6: Activity Diagram..... Page # 49
- Figure 7: User Flowchart.....Page # 62
- Figure 8: Admin Flowchart..... Page # 64

## Abstract

### Main Idea:

Today the amount of information in the internet growth very rapidly and people need some instruments to find and access appropriate information. One of such tools is called recommendation system using collaborative filtering. Recommendation systems help to navigate quickly and receive necessary information Generally. Recommendation systems are extensively used for suggesting new items to users and play an important role in the discovery of relevant new items, be it books, movies, or music. This proposes a quick and intuitive book recommendation system that helps readers to find appropriate book to read next. We used a collaborative filtering method based on Pearson correlation coefficient.

### Technologies and process:

#### Structure:

Recommendation system App with K-NN Algorithm and cosine similarity. The application is an application of Book recommendation very normal, such as Amazon and Goodreads, but it solves the issue of discovering interesting books and then finding them in lists and communicating users with similar interests in chat room dedicated to sharing their interest.

#### Technologies:

Machine learning technologies and algorithms using Python and Python libraries like (NumPy, pandas, matplotlib, sklearn).

REST API technology using “Flask” to connect between flutter app and Machine learning algorithm on local server.

Firebase DB cloud service and Flutter framework using Dart language.

Depending on JASON as mediator.

We used firebase cloud services instead of web services to store data from users like Email, Images and info of Books and users.

We used Flutter framework using Dart.

The application display Books from API that It depends on the interests of each user.

## Process

### Register, login, Discover and Recommend Books, Search, Chat Room

- First: you should register by entering your info (username, email, phone number, ....).
- Then you can login to the application by E-mail & password.
- You can view your profile and can edit your data.
- You can view new books in home page.
- You can discover books based on your interests and depend on similarity for each book.
- You can search for books by book name to display all books that have the same name.
- You can communicate with users with similar interests in Chat Rooms.
- You can enter a name of any book you liked, and the app will respond using similar book.



# Chapter 1: Introduction

## 1. Introduction

In this chapter we are going to discuss and go deeper in the overview of the project and know more about its scope and limitations, and explain some terminologies we will find throughout the document.

## 1.1 Overview

In this application you can say proudly that you help readers to find appropriate book to read next. It is a mobile application that enables the user to discover books that depend on his interests, Recommendation systems are extensively used for suggesting new books to users and play an important role in the discovery of relevant new books, and it enables the users with similar book interests intelligently can communicate in chat room.

## 1.2 Objectives

- The App displays books that depend on user's interests.
- The App shows book details about everything that the user would like to see about the book such as (title, image, Author, year of publication, ...).
- The App displays books that depend on user's interests.
- The user searches for books by book name to display all books that have the same name.
- The App enables the users with similar book interests can chat in Chat Room.
- Enabling excellent user experience and user interface.

## 1.3 Project Purpose

- The user searches for books by book name to display all books that have the same name.
- The App mainly tries to help readers to find appropriate book to read next.
- The App suggesting new books to users and play an important role in the discovers of relevant new books.
- Enable people that need some instruments to find and access appropriate information through books.
- Enable users to share opinions and suggestions with each other through chat rooms to increase their information.

## 1.4 Project Scope

The Application mainly tries to help readers to discover appropriate book to read next and it suggest new books to users that what will save searching time and play an important role in the discovery of relevant new books.

The approximate work involved to finish the project is divided into these five phases: -

### 1. Planning

- Collecting datasets about the project and the lack that made us in a need to the App.
- Searching for suitable machine learning algorithm can be applied on our datasets.
- Making surveys in different ways (video, questionnaires, forms, ...).
- Determining the functional and non-functional requirements.
- Setting a Gantt chart for the project.
- Determining the resources of the team.
- Prepare a UI\UX prototype.
- Dividing the tasks on team using Trello.

### 2. Designing

- Designing recommendation algorithm: -
  1. Reading datasets from .CSV files.
  2. Weight each book based on rating count and rating average.
  3. Filtering unfamous books.
  4. Users with less than 200 ratings and books with less than 100 ratings are excluded.
  5. Drop not used columns.
  6. Combine books table with rating table to get total users rating count for each book.
  7. drop all that does not achieve the value of "popularity" (= 35 in our case)
  8. calculate cosine similarity for every book and other books on the matrix.
  9. Apply KNN algorithm on this matrix to get "K" books (15) on our case.
  10. Enter any query (Book name) And, the algorithm will respond using K nearest books on distance.

- Design API to connect between two different platforms.
- Determining the diagrams to be carried out within the project: -
  - ✓ ERD Diagram
  - ✓ Activity Diagram
  - ✓ Class Diagram
  - ✓ Use-case Diagram
  - ✓ Sequence Diagram

### 3. Coding

The main supposed functions to be coded in this App are: -

- Sign up.
- Login.
- Logout.
- Machine learning algorithm (KNN).
- View Profile
- Edit Profile
- Discover new books.
- Discover recommended books.
- Get similar books.
- View books.
- Search Bar.
- API and JASON code connection.
- Chat room.
- Add book to Favorite list
- Add book to Saved list.
- Add book to Interest book.
- (Contact us) Send Feedback
- View Feedback.
- Add Book.
- Delete Book.
- Edit Book.
- Enable Dark Mode.
- About us.

## 4. Testing

- Functional Testing: -
  - Unit testing
  - Regression testing
  - Integration testing
- Non-functional Testing: -
  - Performance testing
  - Street testing
  - Security testing

## 5. Documentation

Should mainly include these main chapters: -

- Introduction: includes an overview of the project and limitations.
- Project Planning: includes the tools and technologies as well as tasks and timeline plan.
- Project Requirements: includes the functional and nonfunctional requirements.
- Project Design: includes the diagrams.
- Project Implementation: includes user application and administrator system.
- Project Testing: includes testing types.

## 1.5 General Constraints

- Tasks division which can be not fair enough.
- Indiscipline “Human factor” like being late in delivering tasks or attending meetings.
- Hesitation, especially when it is related to taking a serious step or learning a new technique.
- Time management.
- Learning new technologies may take much time.
- Underestimating the objectives that may not lead to realistic or achievable function.

## 6.1 Software Constraints:

- make rules and conventions commonly agreed to in each programming environment explicit and automatically checkable.
- In frontend, we use Flutter Development.
- In Backend, we use Firebase DB Cloud Service.

## 6.2 Hardware Constraints:

- they enable run our application on Android, iOS, and Windows Phone devices.

# Chapter 2: Project Planning and Analysis

## 2. Project planning and analysis

In this chapter we are going to discuss and go deeper in how we plan the project and show the steps and the instructions that we have followed to plan the application.

## 2.1 Project Planning

### 2.1.1 Feasibility study

A Feasibility study is used to determine if a business or a specific project is achievable, so for determining the achievability of our project we will go deeper in the following points: -

#### 1. Market Analysis

- We can make money from our app using ADS.
- Recme mobile app help readers to find appropriate book to read next.
- Recme mobile app enables users to discover books depend on their book interests to recommend books the user would be tempted to read as they are matched with his/her choices.
- Recme mobile app users to search for books by book name to display all books that have the same name.
- Recme mobile app contains the part of social media to make users communicate with each other in chat room and exchange experience.
- Recme mobile app has the part of ranking the result of the search according to your priorities.
- Recme mobile app has property Dark mode that can prolong the battery life of your smartphone and protects eyes from bright screens that hurt the eyes from hours of looking at a bright screen.

#### 2. Organizational Analysis

- The Number of employees in our system: -  
We have 5 employees:
  - = 1 Admins.
  - = 3 Developers.
  - = 1 Testers.
- Payment for your employees?
  - 6,000 L.E for each Developer and tester.
  - 4,000 L.E for each Admin.
- The coverage we need to generate and support our sales: We upgrade the application by the developers.
- We make the application better from the rate and feedback from customers when they do the transactions.



### 3. Operational Analysis

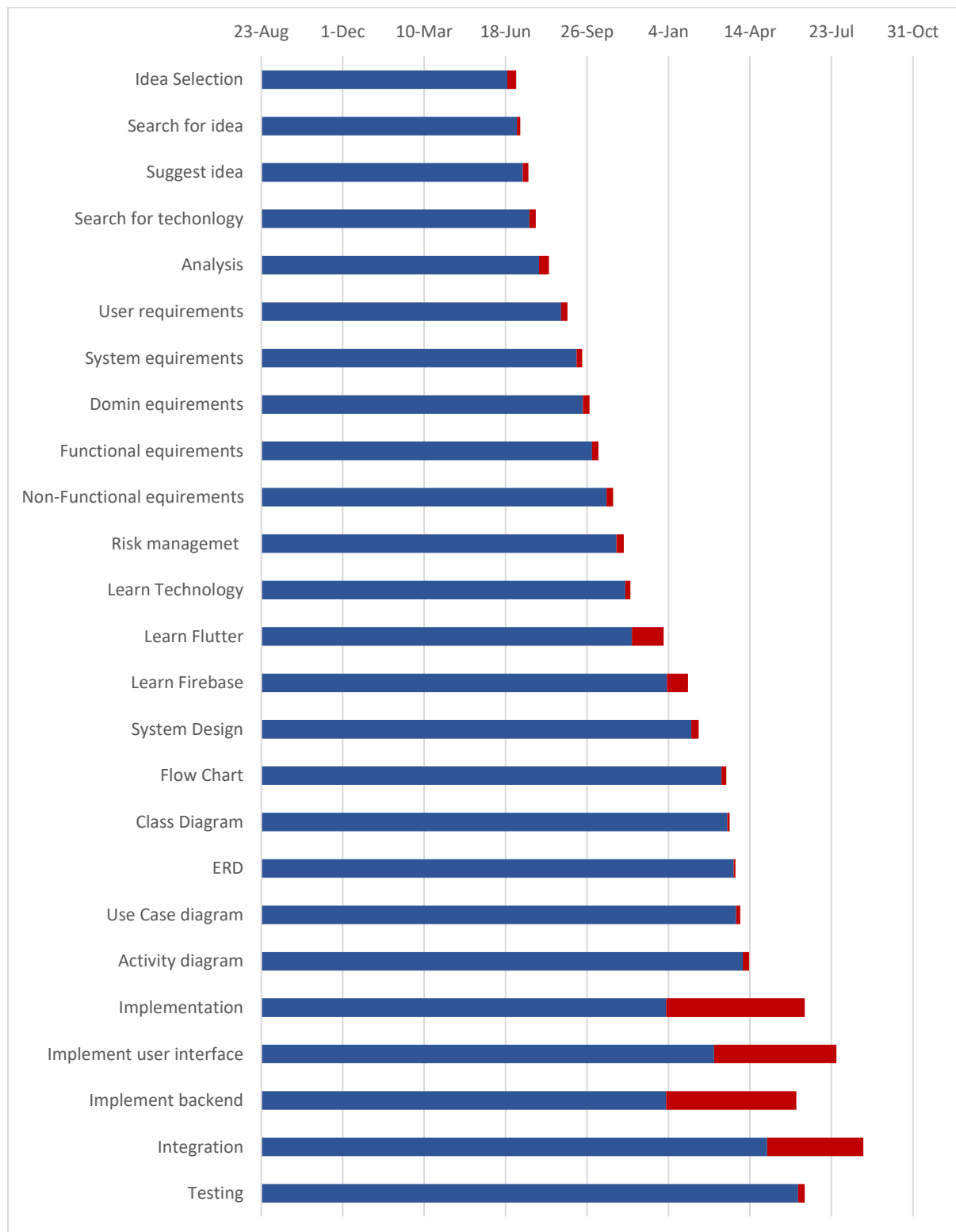
- In our project we provide some rules and services to our employees such as: -
  1. Environment suitable for work.
  2. Commitment to the deadline for delivering tasks.
  3. Good salaries.
- We deliver our service by using social media, offline marketing, and advertisements.
- we used Some equipment like:
  - Microphone: 100 LE
  - Camera: 150 LE
  - Purchasing programs like: -
    1. Adobe Premiere.
    2. Adobe Photoshop.
    3. Adobe illustrator.
    4. Obtaining a hosting server
- We plan to support and retain our customers by providing feedback and promotions offers.

#### 2.1.2 Estimated Cost

A cost estimate is approximation of the cost of a program, project, or operation. the cost estimate is the product of the cost estimating process and our estimated cost for this project comes as following: -

- In our project we had a financial issue like:
  1. Meeting once a week in a public workspace at the first three months: -  
 $(120 \text{ LE in one week} * 2 \text{ weeks in month} = 240 \text{ LE}) (240 * 3 = 720 \text{ LE})$
- We spent a lot of time and effort in self-studying about our technologies and search for the best packages and service to be included in our app.

## 2.1.3 Gantt chart



## 2.2 Analysis and Limitation of existing system

- **Amazon Kindly**



**Features: -**

1. Explore your book and recommendation.
2. Goodreads integration.
3. Read and listen with audible narration.
4. Find your next great read with Kindle.

- **Kobo**



**Features: -**

1. Search by author, title, genre, or you can check out the latest releases or some of the best editions with 'Best of' lists.
2. Get custom recommendations based on books that you read and get to read book previews and reviews that would help you decide the next book.
3. Set time to turn off the story while listening and you can move through your audiobooks with a single touch.

## • Goodreads



### Features: -

1. Find and get books.
2. Get personalized recommendations based on your tastes.
3. Browse featured books and genres to discover more books you will love.
4. See what your friends are reading.

## Our application

### • Recme App



### Features: -

1. Get personalized recommendations based on books that you read and get books.
2. It enables users to search for books by book name to display all books that have the same name.
3. It contains the part of social media to make users communicate with each other in chat rooms and exchange experience.
4. It has property Dark mode that can prolong the battery life of your smartphone and protects eyes from bright screens that hurt the eyes from hours of looking at a bright screen.

App Name	Book recommendation	Search Bar	Chat Room	Light and Dark Theme	list of books Favorite and save
Amazon Kindly	True	True	False	True	False
Kobo	True	True	False	False	False
Goodreads	True	False	False	False	True

## 2.3 Need for the new system

This program helps readers to find appropriate book to read next. It is a mobile application that enables the user to discover books that depend on his interests, find new and interesting books by browsing personalized recommendations based on books you have read and your favorite genres. Recommendation systems are extensively used for suggesting new books to users and play an important role in the discovery of relevant new books, and it enables the users with similar book interests can chat in chat rooms dedicated to each interest. It enables users to discover books depend on their book interests to recommend books the user would be tempted to read as they are matched with his/her choices.

- It enables users to search for books by book name to display all books that have the same name.
- It contains the part of social media to make users communicate with each other in chat rooms and exchange experience.
- It has the part of ranking the result of the search according to your priorities.

- It has property Dark mode that can prolong the battery life of your smartphone and protects eyes from bright screens that hurt the eyes from hours of looking at a bright screen.

## 2.4 Analysis of a new systems

### 2.4.1 User Requirements

#### 1. Mandatory Requirements

- The user needs to discover books that play an important role in the discovery of relevant books. so, we will provide Search bar in our app.
- The user not only needs to discover interested books or search about new books but also to share opinions and suggestions with each other through chat rooms.

#### 2. Desirable Requirements

- The user needs to use Search Bar to make the process easier so we will provide this feature in our app.
- The user uses the app with him at any place and any time so our app should be available in any place and any time.

#### 3. Optional Requirements

- The users prefer registration with special accounts like Facebook or G-mail so we will provide this feature in our app.
- The users prefer Dark mode theme instead of Light mode theme, so we will provide this feature in our app.

### 2.4.2 System Requirements

#### 1. Android

As the application works on the least versions of the android ice cream sandwich because we use a working framework which is responsive to the lowest versions of Android.

#### 2. IOS

The application works on IOS 9 as we use Flutter Framework that support android and iOS and it is the minimum stable version working on it.

### 3. Camera

This app needs a mobile camera as it is very useful in taking pictures for listings to make it easy to the user to choose the suitable listing and it does not matter which version of camera you will use, but it will be awesome if you use a good one.

### 4. Internet Connections

This app uses a cloud server to store data, so you need to be connected to the internet to fetch and view it.

A part of it considered a social media network so the whole data will be on the server [firebase] also you need it to enable Storage service to upload photos.

### 5. Storage space

To keep the user's data in addition to the application itself, we need a fine storage.

## 2.4.3 Domain Requirements

1. Multiple users must be able to use the application simultaneously without corrupting the database (whatever form it may be).
2. The application must verify all values before making the change in the database.
3. The database should be backed up occasionally in case the original does become corrupt.
4. The necessary software required to run the application (Installation for both Android and IOS).
5. The application must have update capabilities for future models and accessories.

## 2.4.4 Functional Requirements

### 1. Signup

<b>Name</b>	Signup
<b>Actor</b>	Client
<b>Input</b>	Fill registration form (Email, username, password, Address, phone...etc.)
<b>output</b>	User will go directly to this profile page
<b>Description</b>	user can create an account to be a part of the application, and login to the application for the second time he opens the application without the need of sign up again.
<b>preconditions</b>	open the application, client should enter a valid data to create an account such username, email, password, and phone.
<b>post conditions</b>	the user should have an account and enter to home page and his profile that has its data which saved when the account created.

### 2. Login

<b>Name</b>	Login
<b>Actor</b>	Client, Admin
<b>Input</b>	e-mail & password
<b>output</b>	Send the user into the home page
<b>Description</b>	user should enter his email and his password for authorizing that he has an account
<b>preconditions</b>	open the application, enter login Button, and enter a valid e-mail and password.
<b>post conditions</b>	Logged into the system then enter to home page then he can go to his profile that has its data which saved in the firebase.



### 3. Update Profile

<b>Name</b>	Update profile
<b>Actor</b>	Client
<b>Input</b>	the updated information
<b>output</b>	Update new data in DB
<b>Description</b>	If user wants to change any data in his profile the app will update Them and save in the database
<b>preconditions</b>	There is actual account sign up or login, open his profile.
<b>post conditions</b>	the updated information will be displayed in the profile and saved in the firebase.

### 4. Logout

<b>Name</b>	Logout
<b>Actor</b>	Client, Admin
<b>Input</b>	User token in data base
<b>output</b>	End user session
<b>Description</b>	After the user login to the application, they can logout easily from account.
<b>preconditions</b>	open the application, login, enter log out Button you have been logged in first.
<b>post conditions</b>	user will log out successfully and could not use the application without doing login again

## 5. Forget Password

<b>Name</b>	Forget Password
<b>Actor</b>	Client
<b>Input</b>	Change password
<b>output</b>	Password changed
<b>Description</b>	help the user to change password when he forgets it, the user should click on forget password and then enter his e-mail to make sure that he is the correct user and sent the change password form to his mail.
<b>preconditions</b>	open the application, user should have an account, click on forget password button.
<b>post conditions</b>	this password should save in the firebase to use it when you make login before.

## 6. Discover Books

<b>Name</b>	Discover Books
<b>Actor</b>	Client
<b>Input</b>	Enter to home page
<b>output</b>	The books display.
<b>Description</b>	The user Discover books depends on his interests and can see details it.
<b>preconditions</b>	Sign up or login and enter to home page then to recommend page and put the book name in text Field to see books.
<b>post conditions</b>	The user will discover the books depends on his interests and can see details it.

## 7. View Books

<b>Name</b>	View Books
<b>Actor</b>	Client, Admin
<b>Input</b>	Press on any book
<b>output</b>	Send the user into book details
<b>Description</b>	The user choose book depend on his interests and want to see book details to discover information about this book.
<b>preconditions</b>	You must have an account and logged in successfully then go to home page and press on your books to see details.
<b>post conditions</b>	The books' details will display.

## 8. Search

<b>Name</b>	Search
<b>Actor</b>	Client
<b>Input</b>	Write Book's name
<b>output</b>	The books and details about the books.
<b>Description</b>	Help the user to search about books that have the same book name.
<b>preconditions</b>	Sign up or login and go to search icon then write book's name in text field.
<b>post conditions</b>	All the books that have same book name will display.

## 9. Send Feedback

<b>Name</b>	Send Feedback
<b>Actor</b>	Client
<b>Input</b>	e-mail, username, and message
<b>output</b>	The message sent to Firebase and to admin
<b>Description</b>	If the user wants to any feedback about the application, the user enters email, password, and Message.
<b>preconditions</b>	You must have an account and logged in successfully then go to contact us page and enter a valid e-mail and password and enter the message.
<b>post conditions</b>	Show message show that “The message sent successfully” and feedback will be sent to the admin.

## 10. Send Message

<b>Name</b>	Send Message
<b>Actor</b>	Client
<b>Input</b>	Type a message
<b>output</b>	Chatting with users in chat room
<b>Description</b>	Share opinions and suggestions with each other through chat rooms
<b>preconditions</b>	Open application, you must have an account and logged in successfully then go to chat room and enter your message.
<b>post conditions</b>	Your messages sent successfully, and the user now can be chatting with users in chat room.

## 11. Add to favorite

<b>Name</b>	Add to favorite
<b>Actor</b>	Client
<b>Input</b>	Click on heart icon
<b>output</b>	The book display in favorite list.
<b>Description</b>	If user likes the book and want to save it to see it later, he will click on heart icon (the symbol of favorites).
<b>preconditions</b>	open the application, login, user must have an account then the client will press on any book he wants.
<b>post conditions</b>	The book will be saved to his favorites to see it whenever he wants.

## 12. Add to saved

<b>Name</b>	Add to Saved
<b>Actor</b>	Client
<b>Input</b>	Click on save icon
<b>output</b>	The book display in save list.
<b>Description</b>	If he saves the book and want to save it to see it later, he will click on save icon (the symbol of saves).
<b>preconditions</b>	open the application, login, user must have an account then the client will press on any book he wants.
<b>post conditions</b>	The book will be saved to his saves to see it whenever he wants.

### 13. Add to interest

<b>Name</b>	Add to Interest
<b>Actor</b>	Client
<b>Input</b>	Click on interest icon
<b>output</b>	The book display in home page.
<b>Description</b>	If user click on interest icon in any book that recommended to user, the book will display in user's interest books home page.
<b>preconditions</b>	Open the application, login, user must have an account then the client will press on any book that is interest for user.
<b>post conditions</b>	The book will be saves to his interest books to see it whenever he wants.

### 14. View Feedback

<b>Name</b>	Add to favorite
<b>Actor</b>	Admin
<b>Input</b>	Click on view feedback button
<b>output</b>	All feedback will display
<b>Description</b>	The users sent messages to admin and the admin want to see these message
<b>preconditions</b>	Open the application, login as admin with valid email and password and enter to view feedback page.
<b>post conditions</b>	The admin will see All Feedback.

## 15. Add book

<b>Name</b>	Add book
<b>Actor</b>	Admin
<b>Input</b>	Fill Add book Form (Title, description, Author, ISBN, Category, Year of publication, etc..).
<b>output</b>	The data saved in DB
<b>Description</b>	The admin wants to display books in home page to discover it the users.
<b>preconditions</b>	Open the application, login as admin with valid email and password and enter to Add book page.
<b>post conditions</b>	The book saved in DB and sent to User homepage.

## 16. View book

<b>Name</b>	View books
<b>Actor</b>	Admin
<b>Input</b>	Press on any book
<b>output</b>	Sent the admin into book details.
<b>Description</b>	The admin wants to see the book details that he adds it to user and if he wants to update or delete it.
<b>preconditions</b>	Open the application, login as admin with valid email and password and enter to view book page then press on any book.
<b>post conditions</b>	The book details will display

## 17. Delete book

<b>Name</b>	Delete books
<b>Actor</b>	Admin
<b>Input</b>	Click on delete icon
<b>output</b>	The book will delete.
<b>Description</b>	The admin wants to delete any book.
<b>preconditions</b>	Open the application, login as admin with valid email and password, enter to view book page, press on any book then click on delete icon.
<b>post conditions</b>	Show message say, "the book is deleted successfully".

## 18. Update Books

<b>Name</b>	Update books
<b>Actor</b>	Admin
<b>Input</b>	Click on edit icon
<b>output</b>	Sent the admin to update book page
<b>Description</b>	The admin wants to update any book.
<b>preconditions</b>	Open the application, login as admin with valid email and password, enter to view book page, press on any book then click on edit icon then go to update home.
<b>post conditions</b>	Show message say, "the book is updated successfully".



## 2.4.5 Nonfunctional Requirement

### 1. Availability:

the system should be available all the time in case of a hardware failure or database or delay a replacement page will be shown, and in case of hardware or database delay backups will be retrieved from application data folder -it means 24/7 available.

### 2. Flexibility:

if the team intends to increase the application functionality and extends it, that would be easy as the application is prepared from the beginning to be expanded for future.

### 3. Robustness:

Our application can handle error conditions gracefully, without failure. This includes a tolerance of invalid data, software defects, and unexpected operating conditions.

### 4. Scalability:

Our application is scalable can handle a wide variety of system configuration sizes. By adding new features and new machines. Usability: Our application is Ease to use requirements address the factors that constitute the capacity of the software to be understood, learned, and used by its intended users.

### 5. Reliability:

the system provides the right tools for problem solving it is made in such as the way that the system is reliable in its operations and for securing the sensitive details.

## 2.5 Advantages of the new system

- We can make money from our app using ADS.
- Recme mobile app help readers to find appropriate book to read next.
- Get personalized recommendations and discover new books based on your tastes.
- Recme mobile app enables users to discover books depend on their book interests to recommend books the user would be tempted to read as they are matched with his/her choices.
- Recme mobile app enables user to search about books that play an important role in the discovery of relevant books.
- Recme mobile app contains the part of social media to make users communicate with each other in chat rooms and exchange experience.
- Recme mobile app has the part of ranking the result of the search according to your priorities.
- Recme mobile app has property Dark mode that can prolong the battery life of your smartphone and protects eyes from bright screens that hurt the eyes from hours of looking at a bright screen.

## 2.6 Risk and Risk Management

### Risk Management Process:

#### 1. Risk identification:

Risk Type	Possible Risk
<b>Technology</b>	The database used in the system cannot process as many transactions per second as expected. Software components that should be reused contain defects that limit their functionality.
<b>People</b>	It is impossible to recruit staff with the skills required. Key staff are ill and unavailable at critical times. Required training for staff is not available.
<b>Organizational</b>	The organization is restructured so that different management are responsible for the project.
<b>Tool</b>	The code generated by CASE tools is inefficient. CASE tools cannot be integrated.
<b>Requirement</b>	Changes to requirements that require major design rework are proposed. Customers fail to understand the impact of requirements changes
<b>Estimation</b>	The time required to develop the software is underestimated. The rate of defect repair is underestimated. The size of the software is underestimated

## 2. Risk analysis

<b>Risk</b>	<b>Probability</b>	<b>Effects</b>
<b>Organizational financial problems force reductions in the project budget.</b>	low	Catastrophic
<b>It is impossible to recruit staff with the skills required for the project</b>	high	Catastrophic
<b>Key staff are ill at critical times in the project</b>	moderate	Serious
<b>Software components that should be reused contain defects which limit them functionality.</b>	moderate	Serious
<b>Changes to requirements that require major design rework is proposed</b>	moderate	Serious
<b>The organization is restructured so that different manage met are responsible for the project</b>	high	Serious
<b>The database used in the system cannot process as many transactions per second as expected</b>	moderate	Serious
<b>The time required to develop the software is underestimated</b>	high	Serious
<b>CASE tools cannot be integrated</b>	High	Tolerable
<b>Customers fail to understand the impact of requirements changes.</b>	moderate	Tolerable
<b>The rate of defect repair is underestimated</b>	moderate	Tolerable
<b>The size of the software is underestimated</b>	high	Tolerable
<b>There are many competitors.</b>	high	Tolerable

### 3. Risk planning

Risk	Strategy
<b>Organizational financial problems</b>	Prepare a briefing document for senior management showing how the project is making a very important contribution to the goals of the business.
<b>Recruitment problems</b>	Alert customer of potential difficulties and the possibility of delays, investigate buying-in components
<b>Staff illness</b>	Reorganize team so that there is more overlap of work and people therefore understand each other's jobs.
<b>Defective components</b>	Replace potentially defective components with bought-in components of known reliability
<b>Requirements changes</b>	Derive traceability information to assess requirements change impact, maximize information hiding in the design
<b>Organizational restructuring</b>	Prepare a briefing document for senior management showing how the project is making a very important contribution to the goals of the business.
<b>Database performance</b>	Investigate the possibility of buying a higher performance database

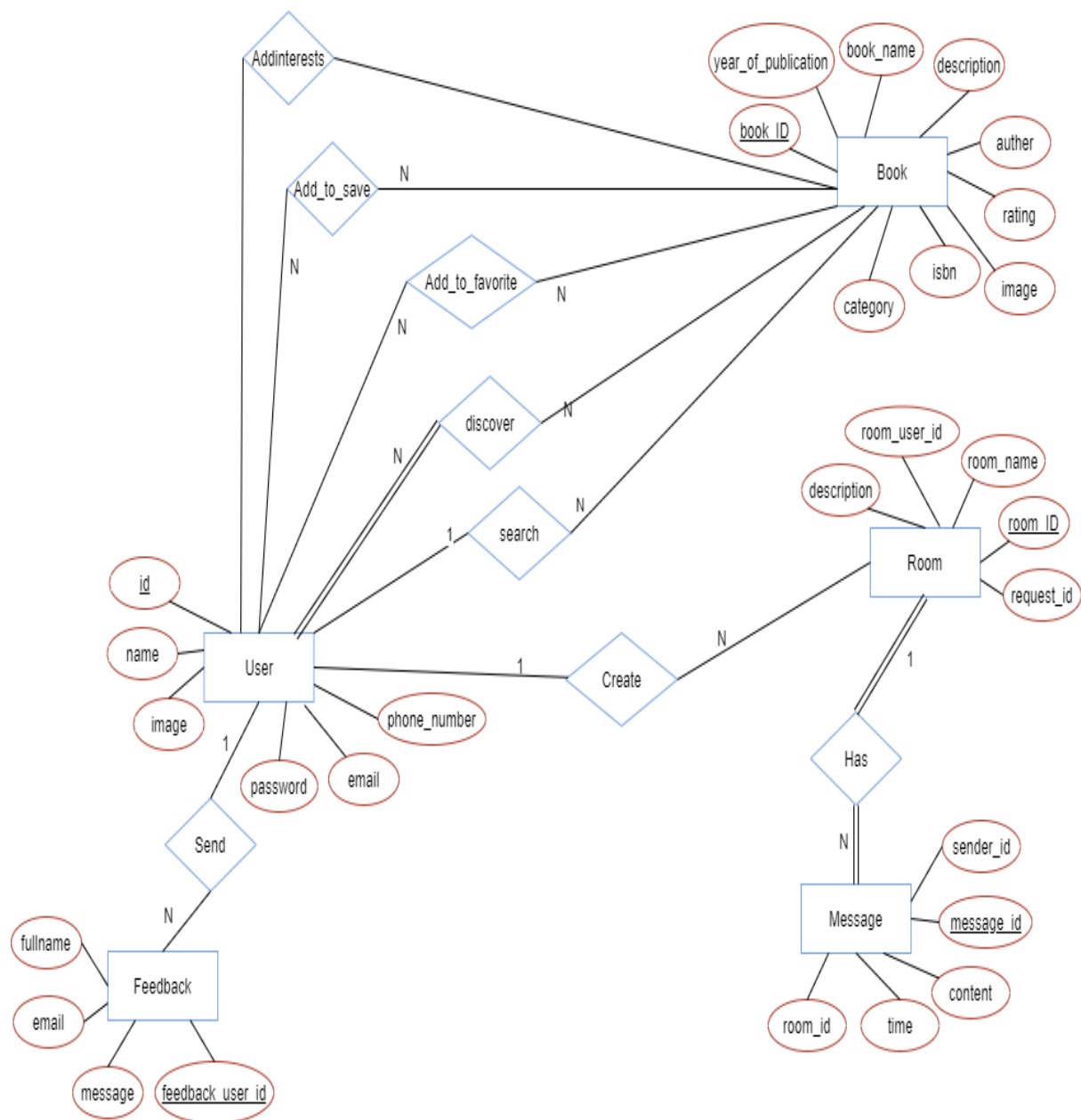
# Chapter 3:

## Software Design

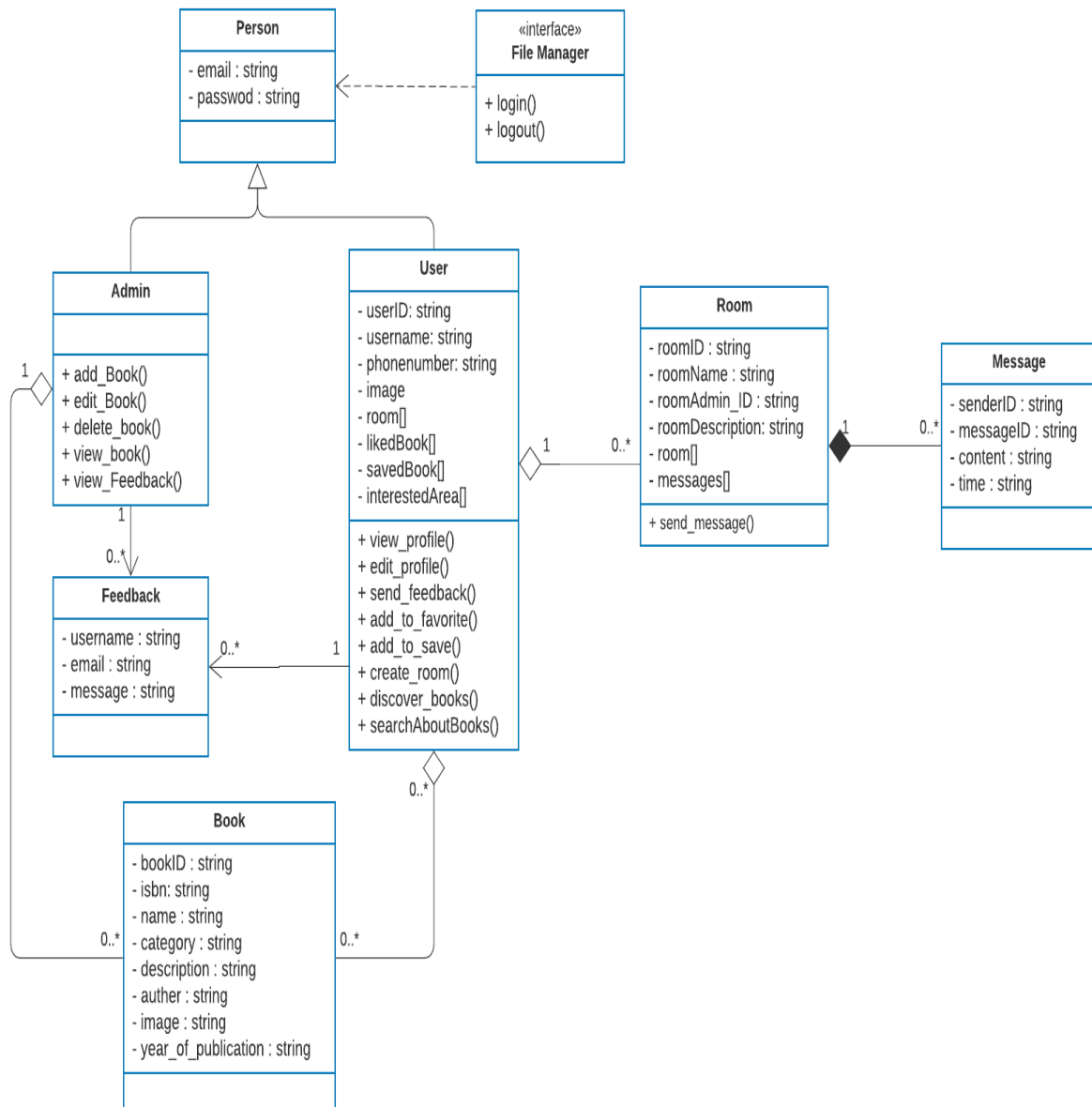
### 3. Software Design

In this chapter we are going to discuss and go deeper in Settle mobile application's design, and present its diagrams and database.

### ER Diagram

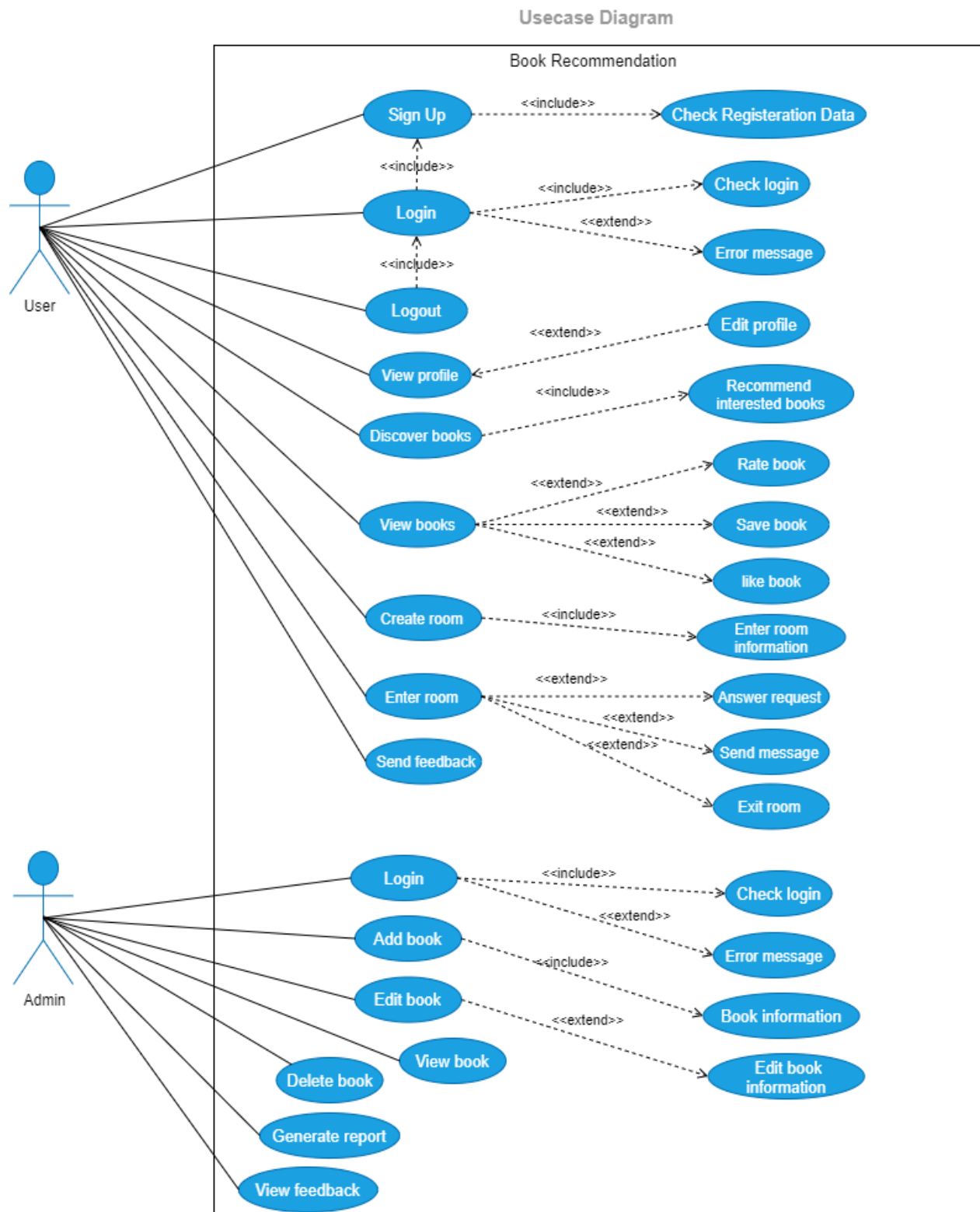


## 3.2 Class Diagram



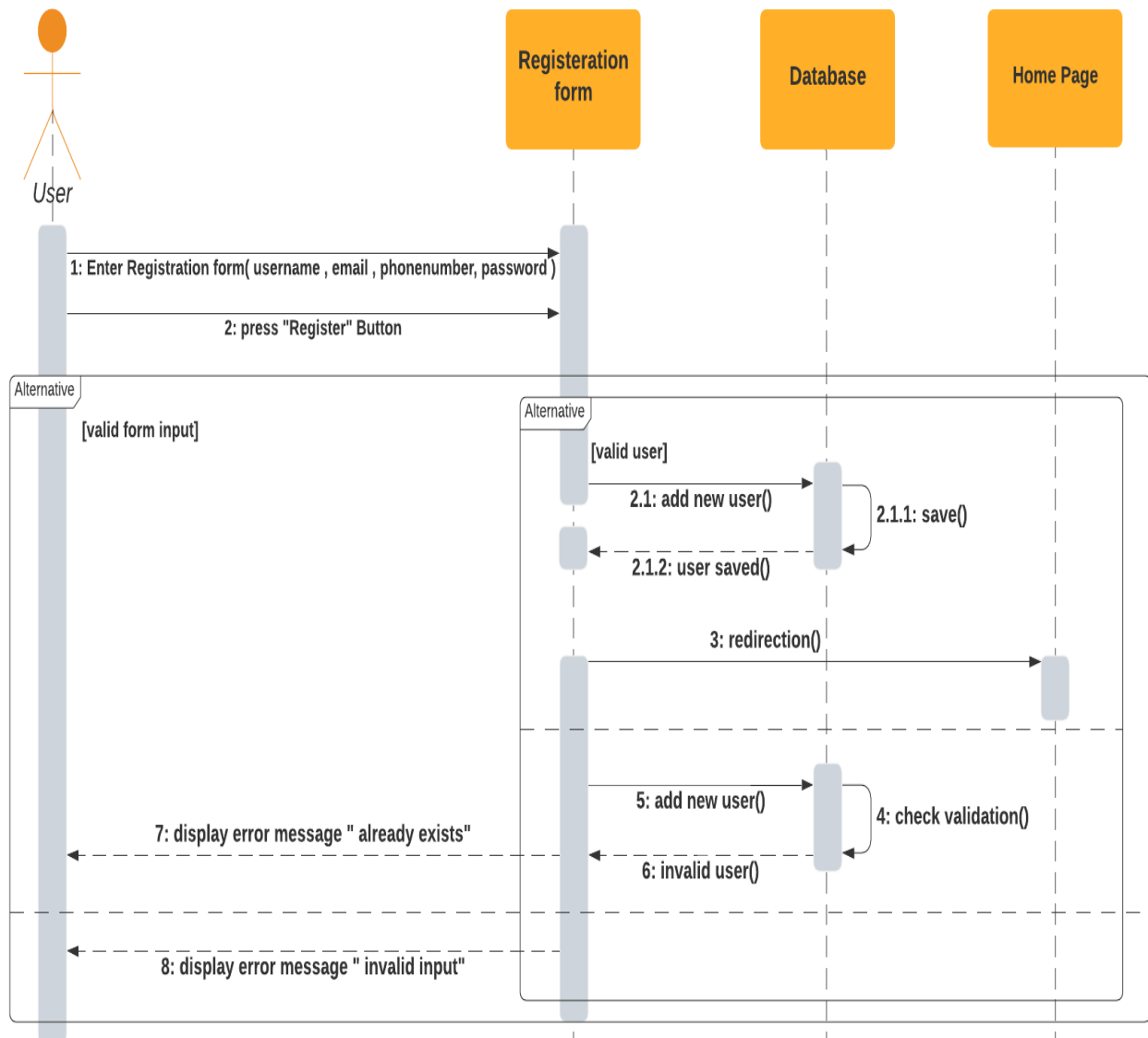


## 3.3 Use case Diagram

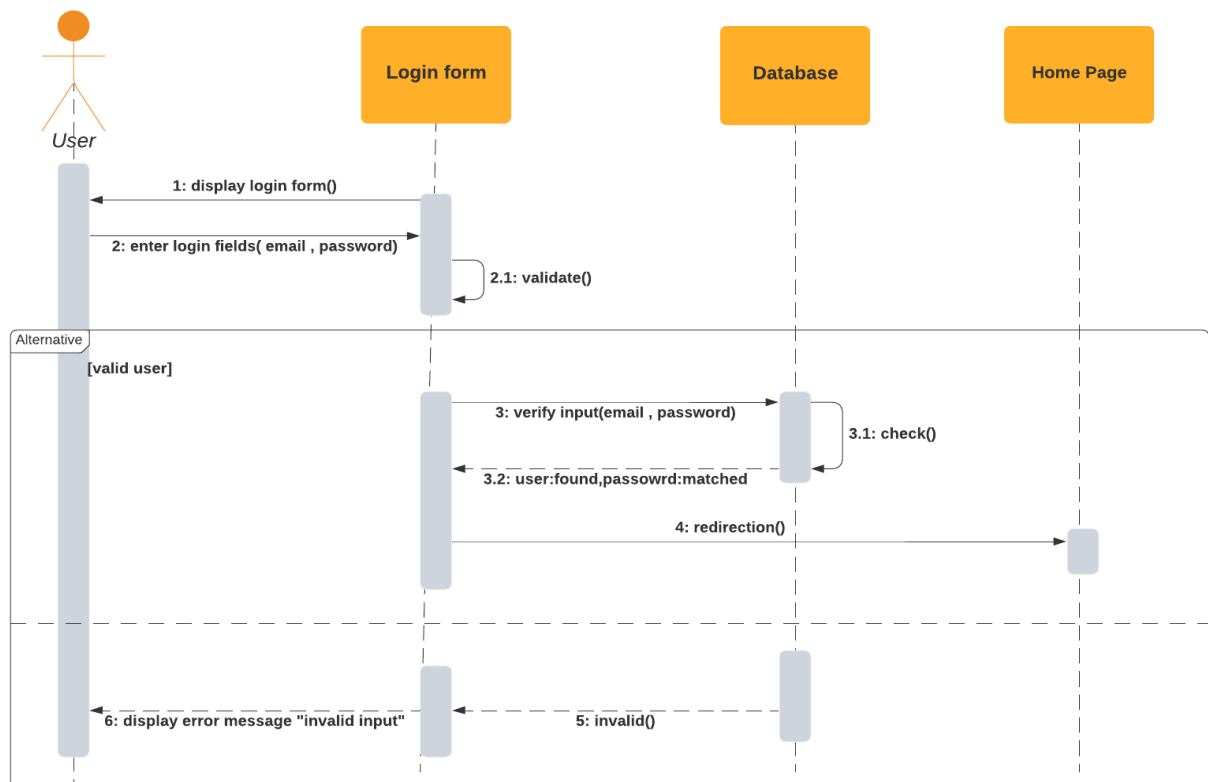


## 3.4 Sequence Diagram

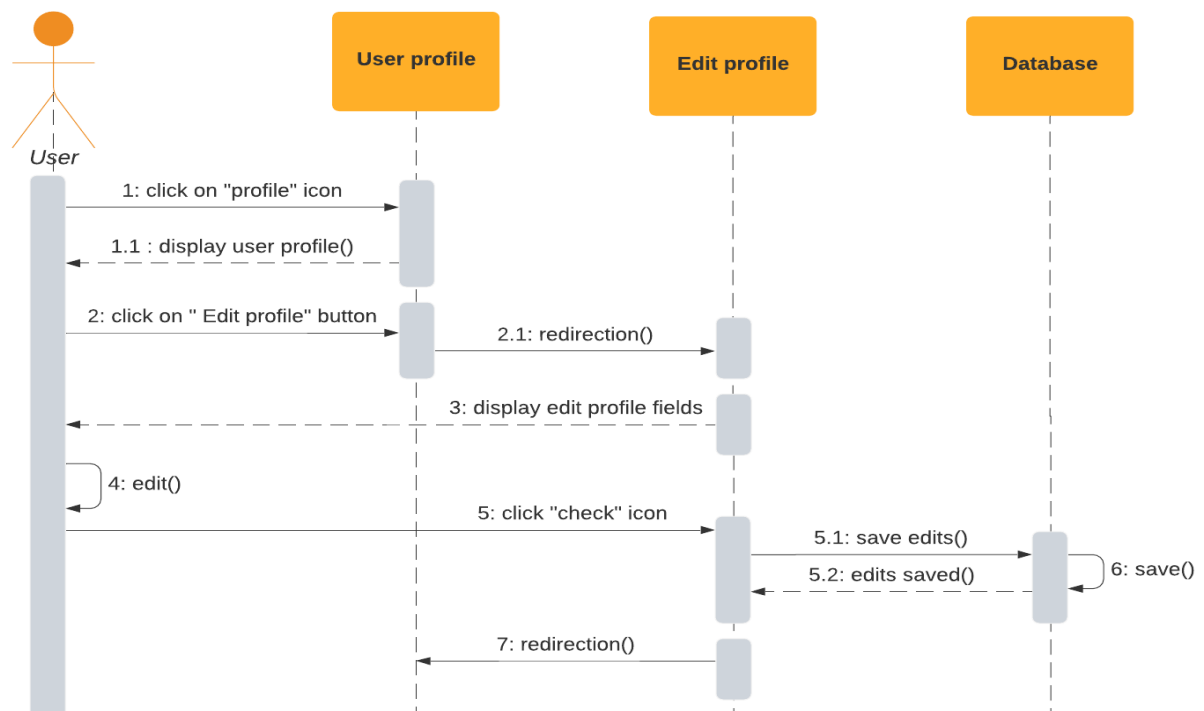
### 3.4.1 Signup



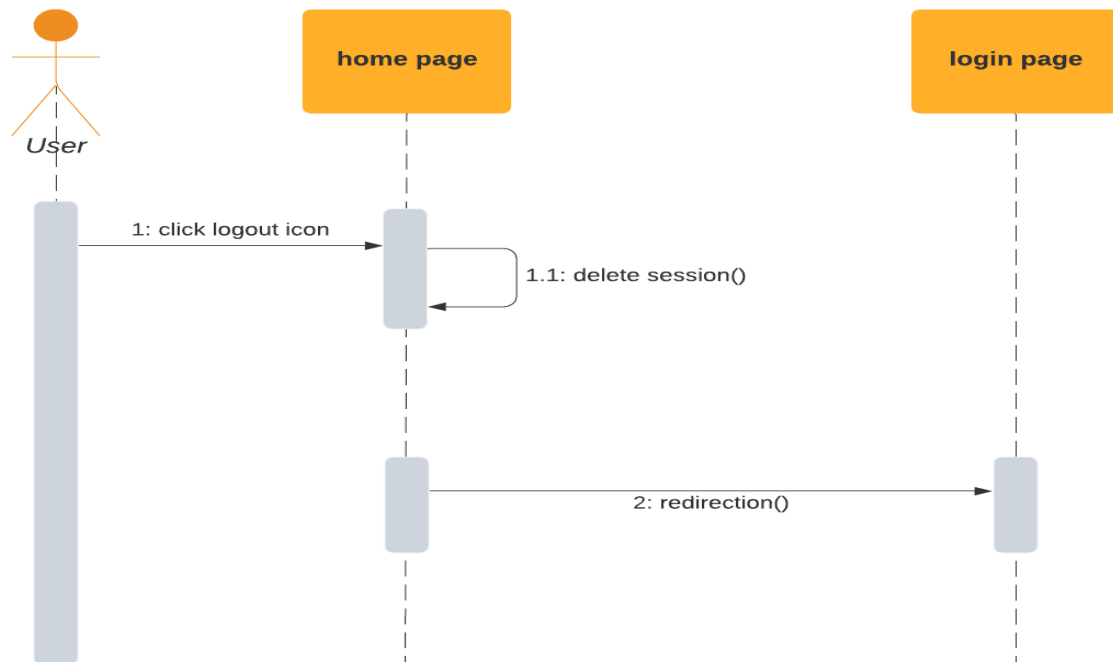
### 3.4.2 Login



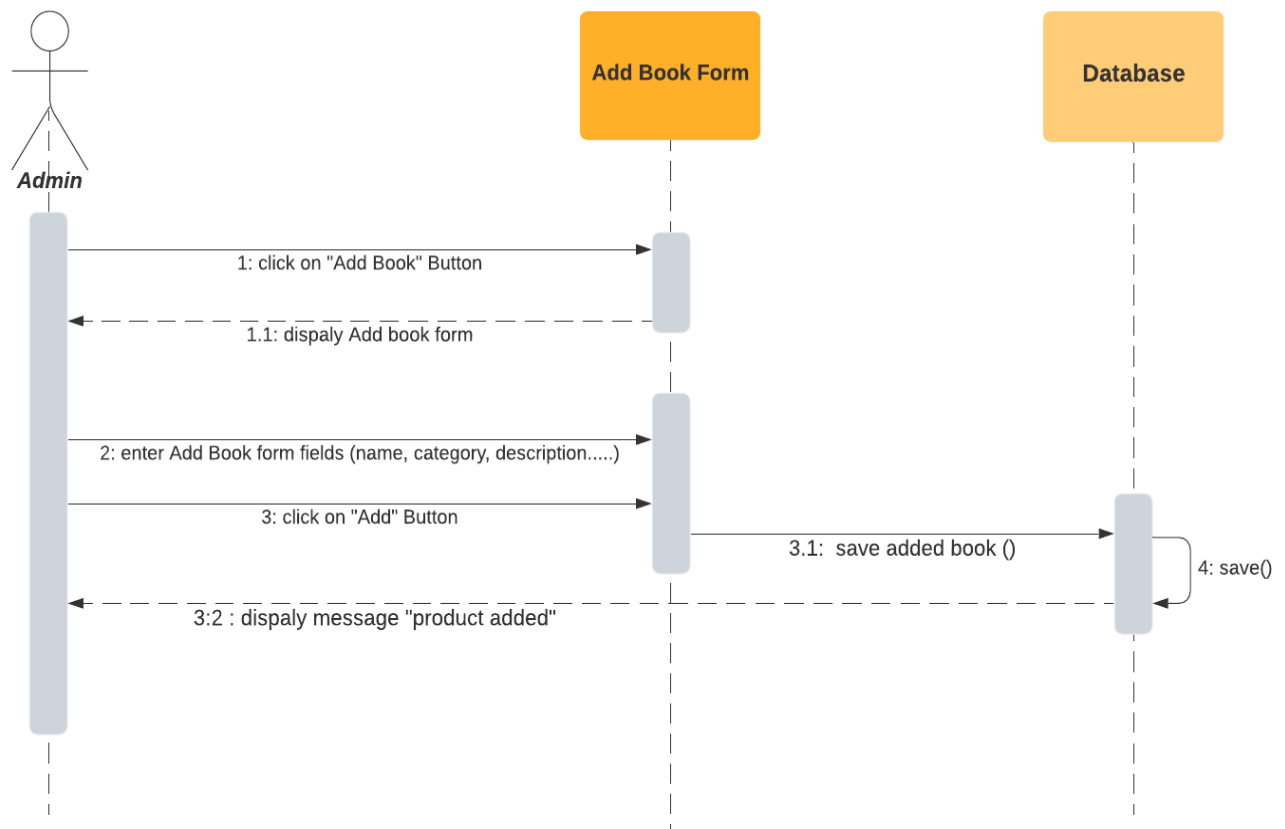
### 3.4.3 Edit Profile



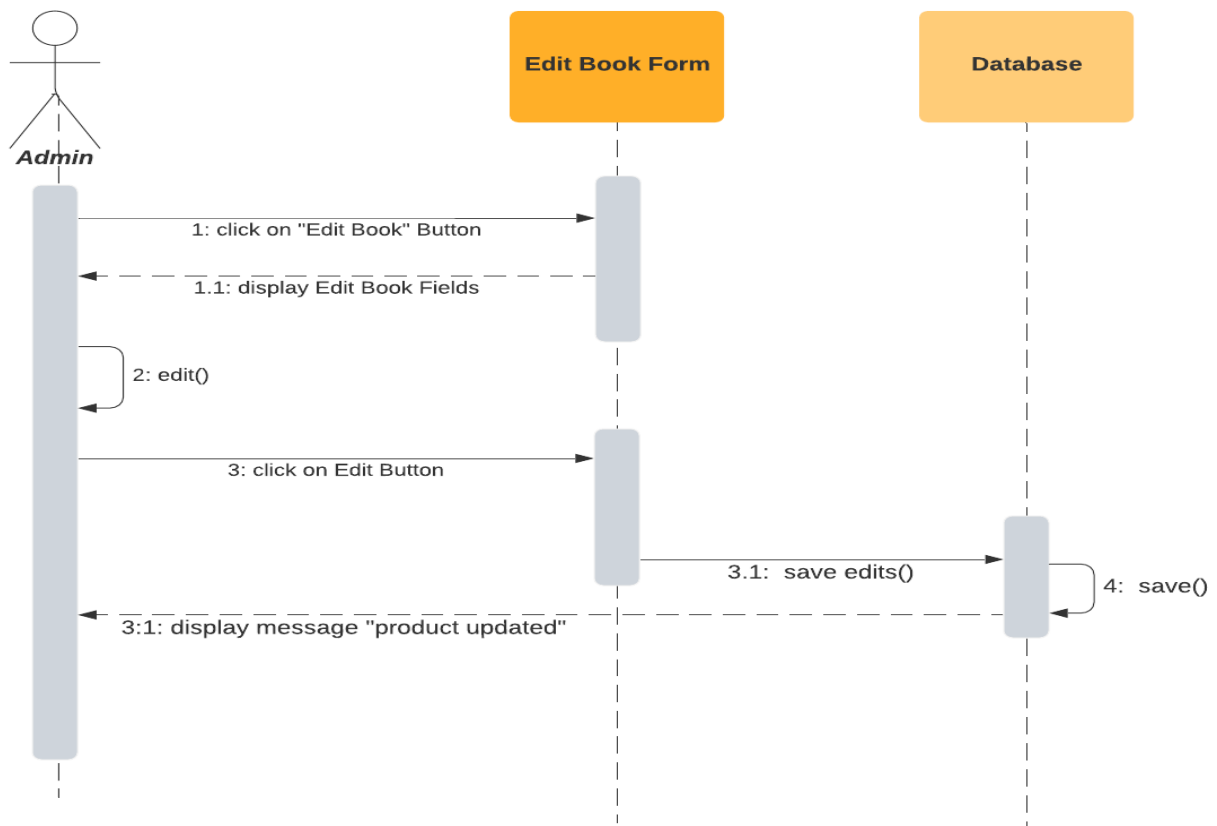
### 3.4.4 Logout



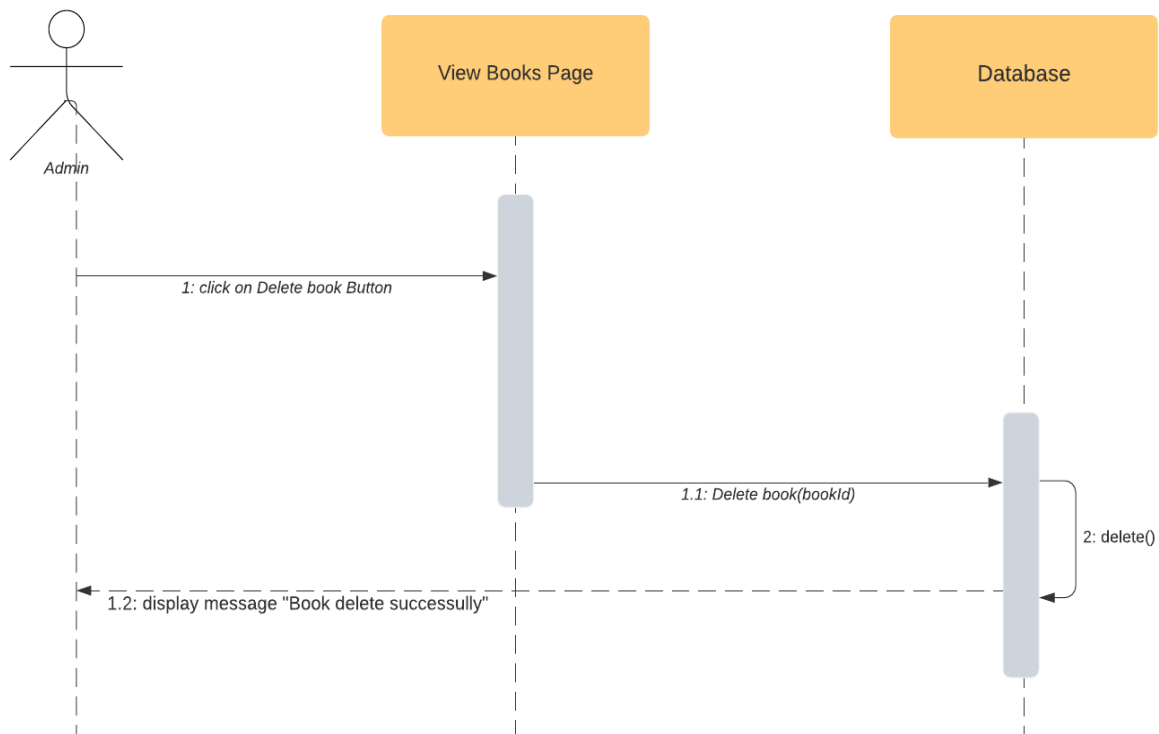
### 3.4.5 Add Book



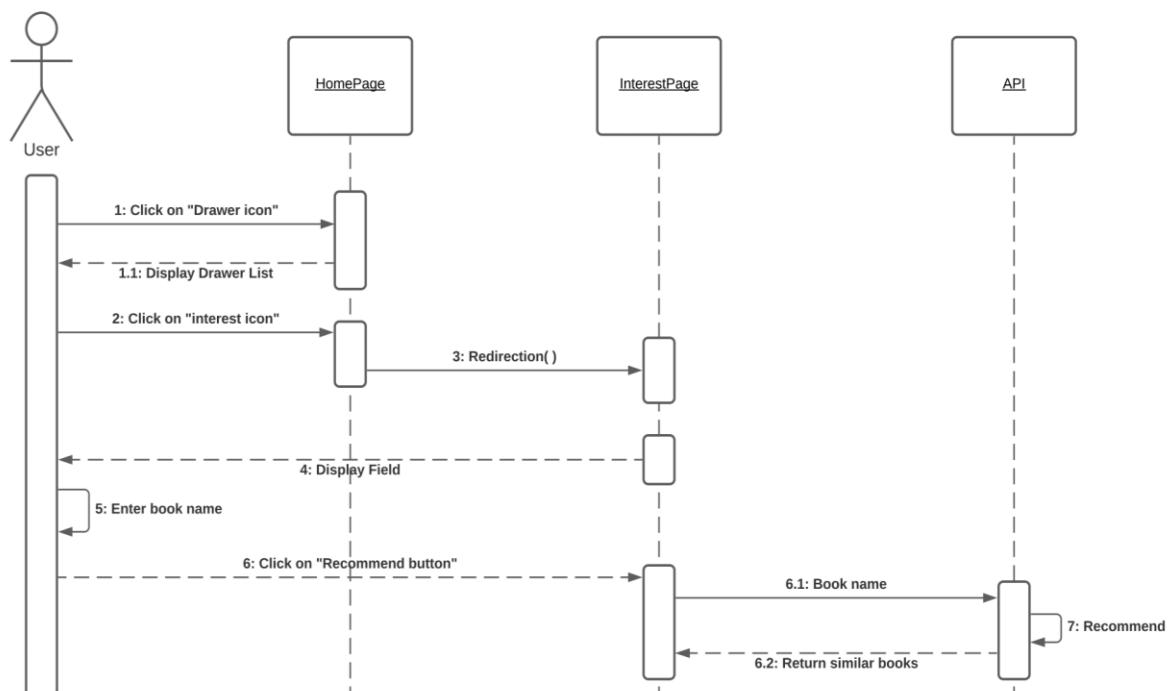
### 3.4.6 Edit Book



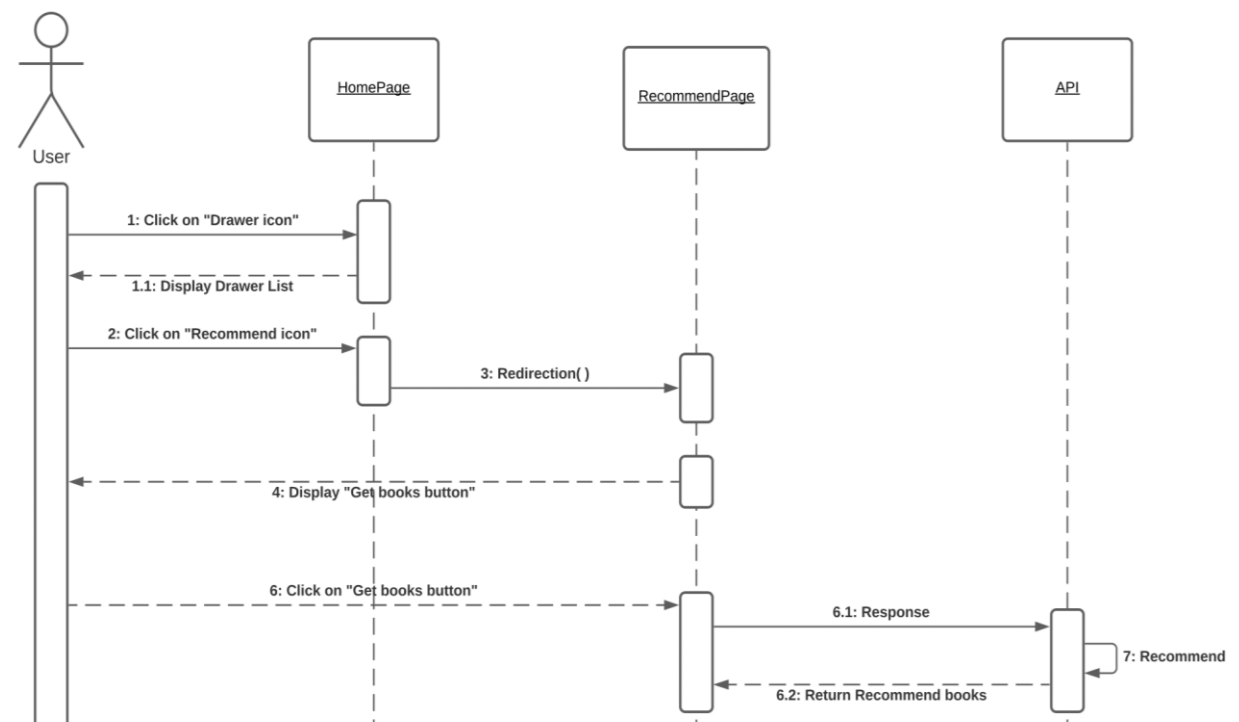
### 3.4.7 Delete Book



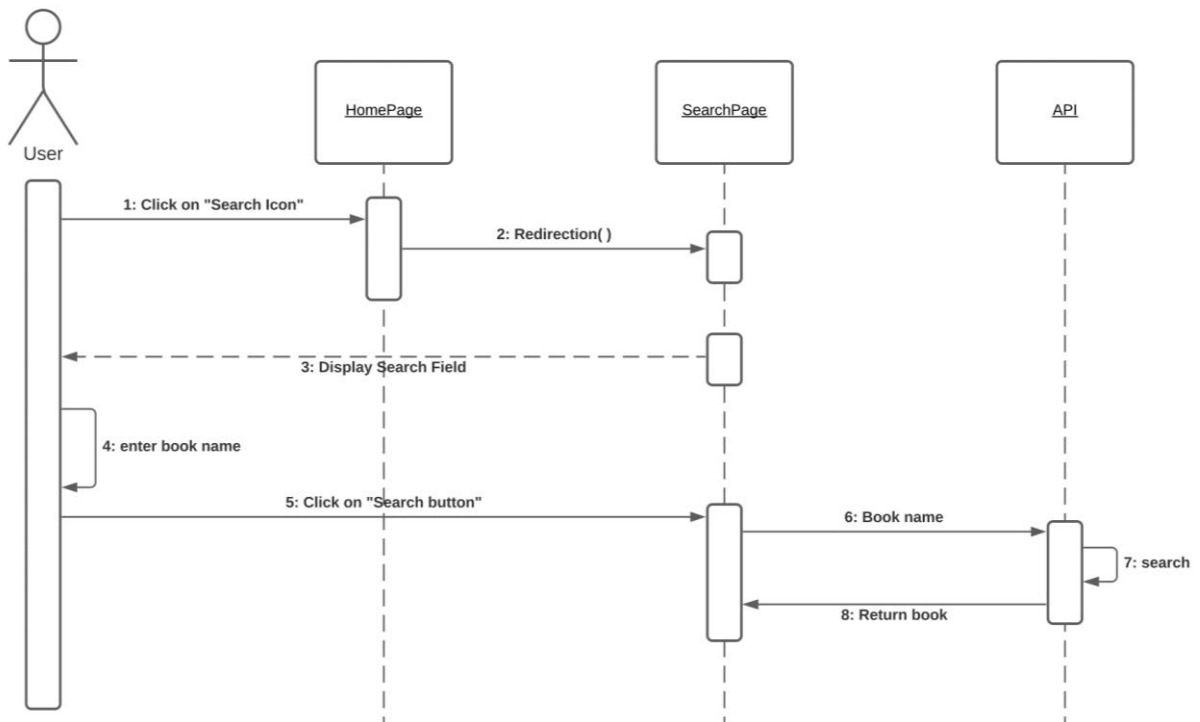
### 3.4.8 Interest Book



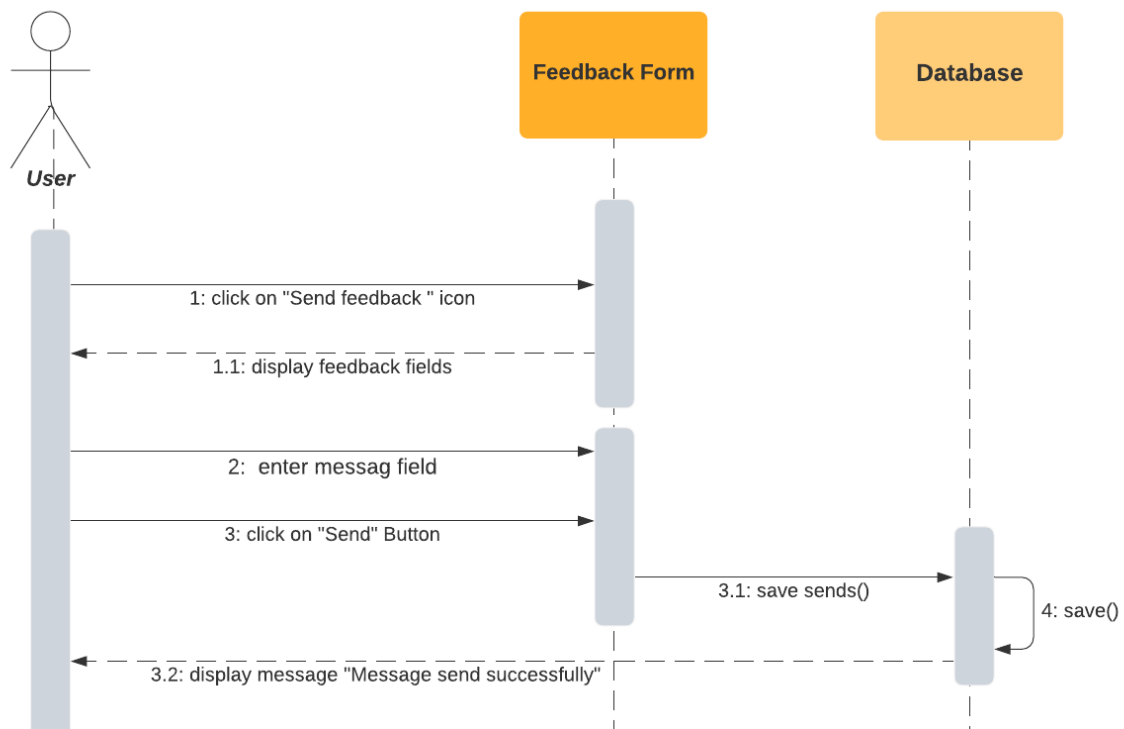
### 3.4.9 Recommend Books



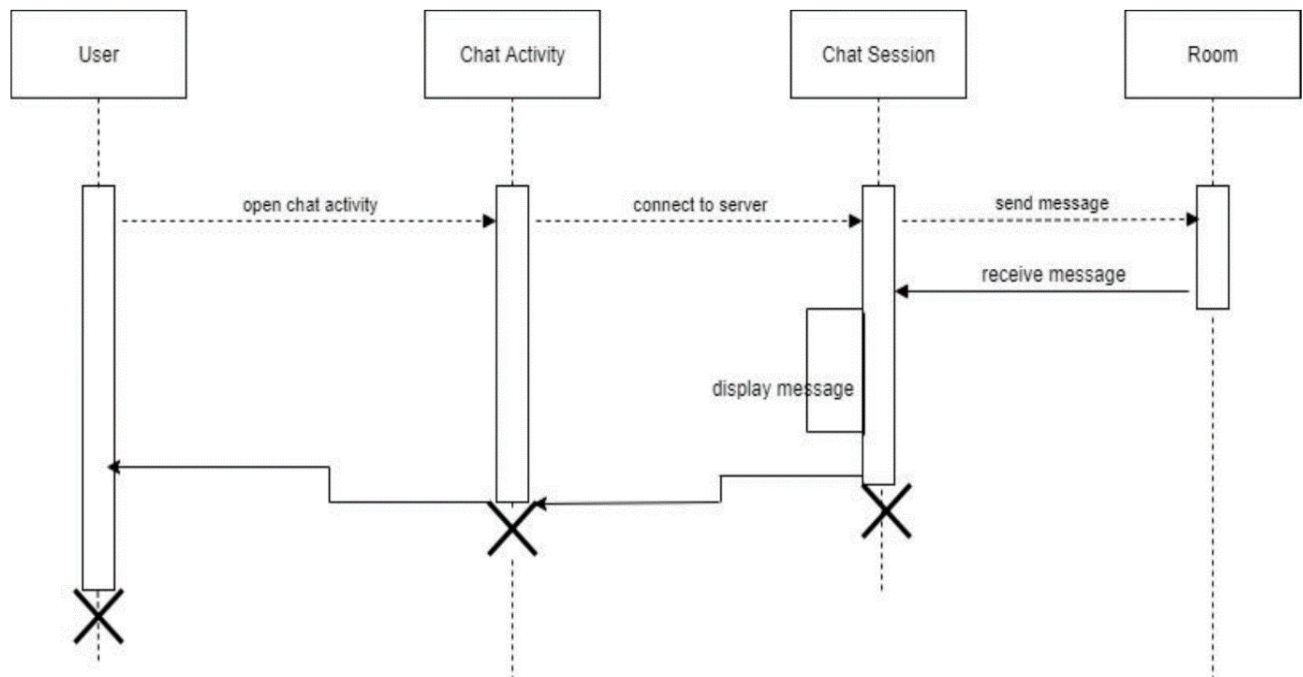
### 3.4.10 Search



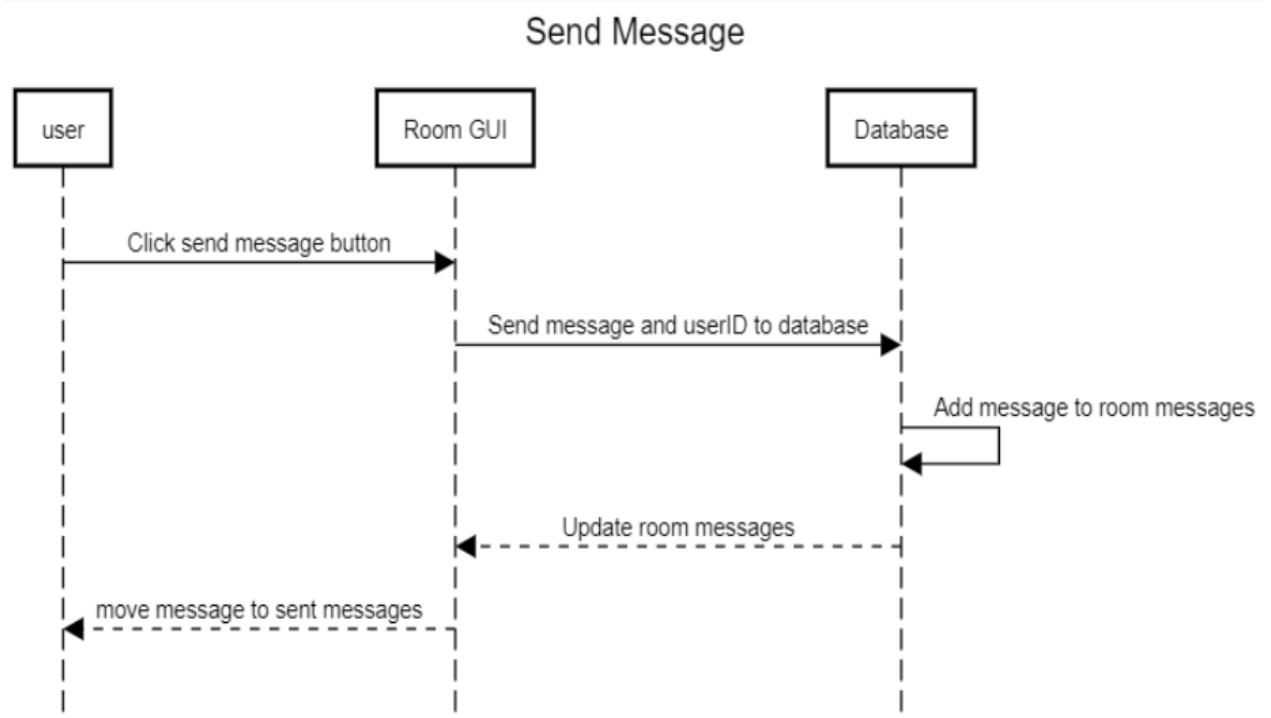
### 3.4.11 Send Feedback



### 3.4.12 Chat Room

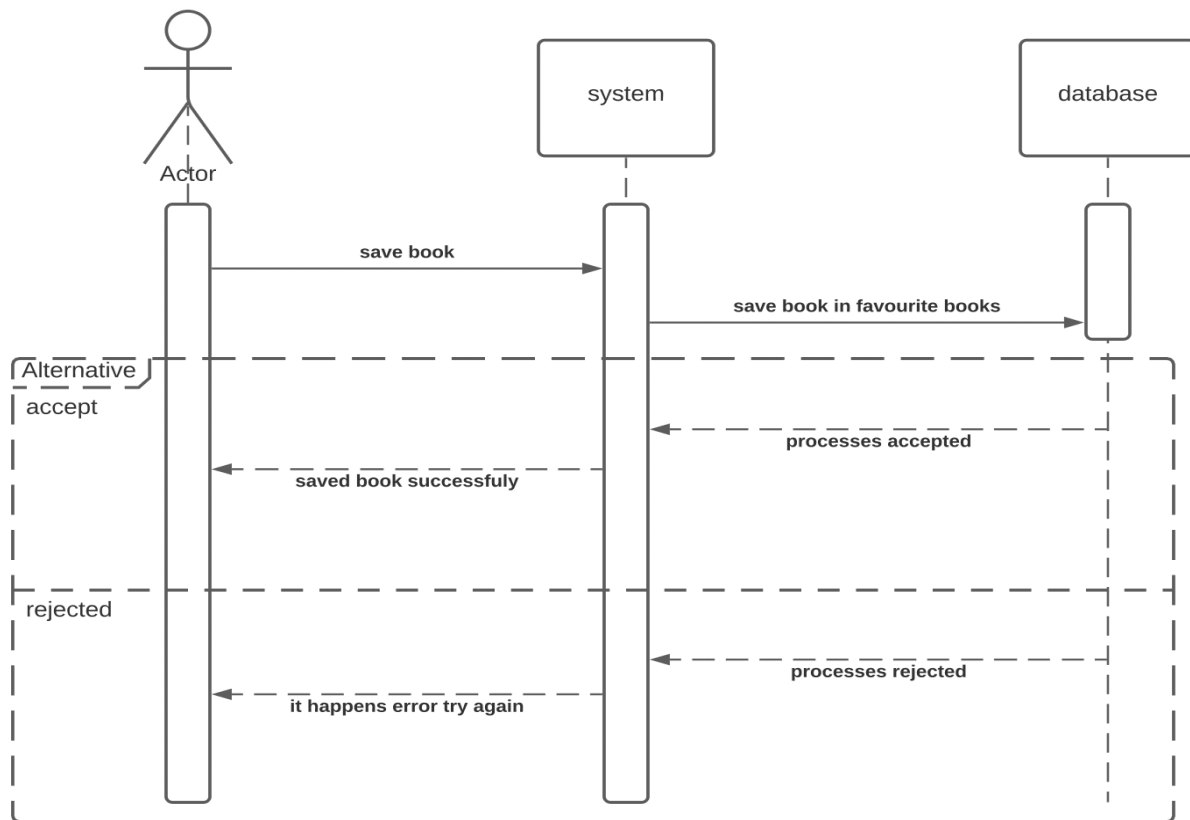


### 3.4.13 Send Message





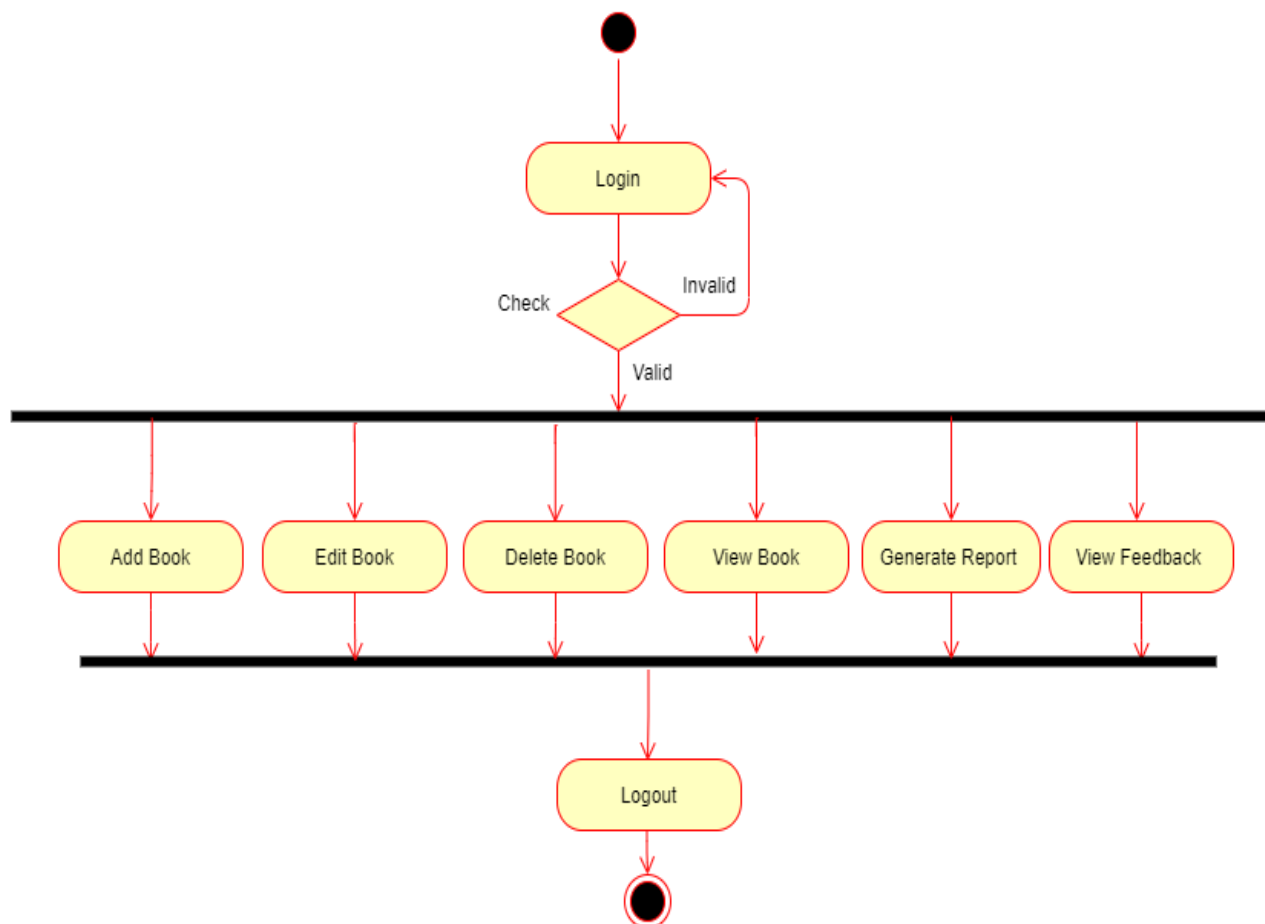
### 3.4.14 Save Books



## 3.5 Activity Diagram

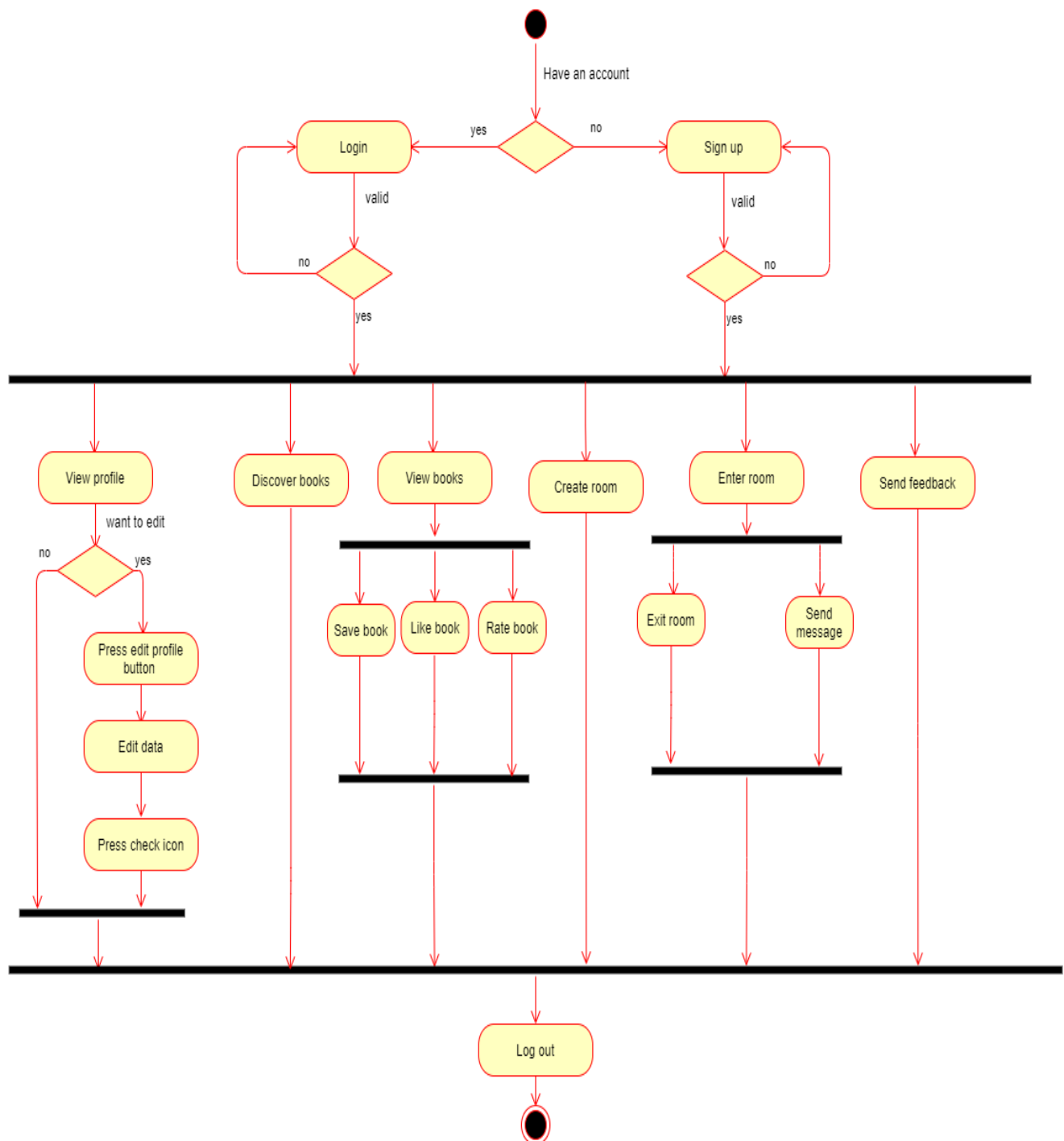
### 3.5.1 Admin Activity

Admin Activity Diagram



## 3.5.2 User Activity

User Activity Diagram



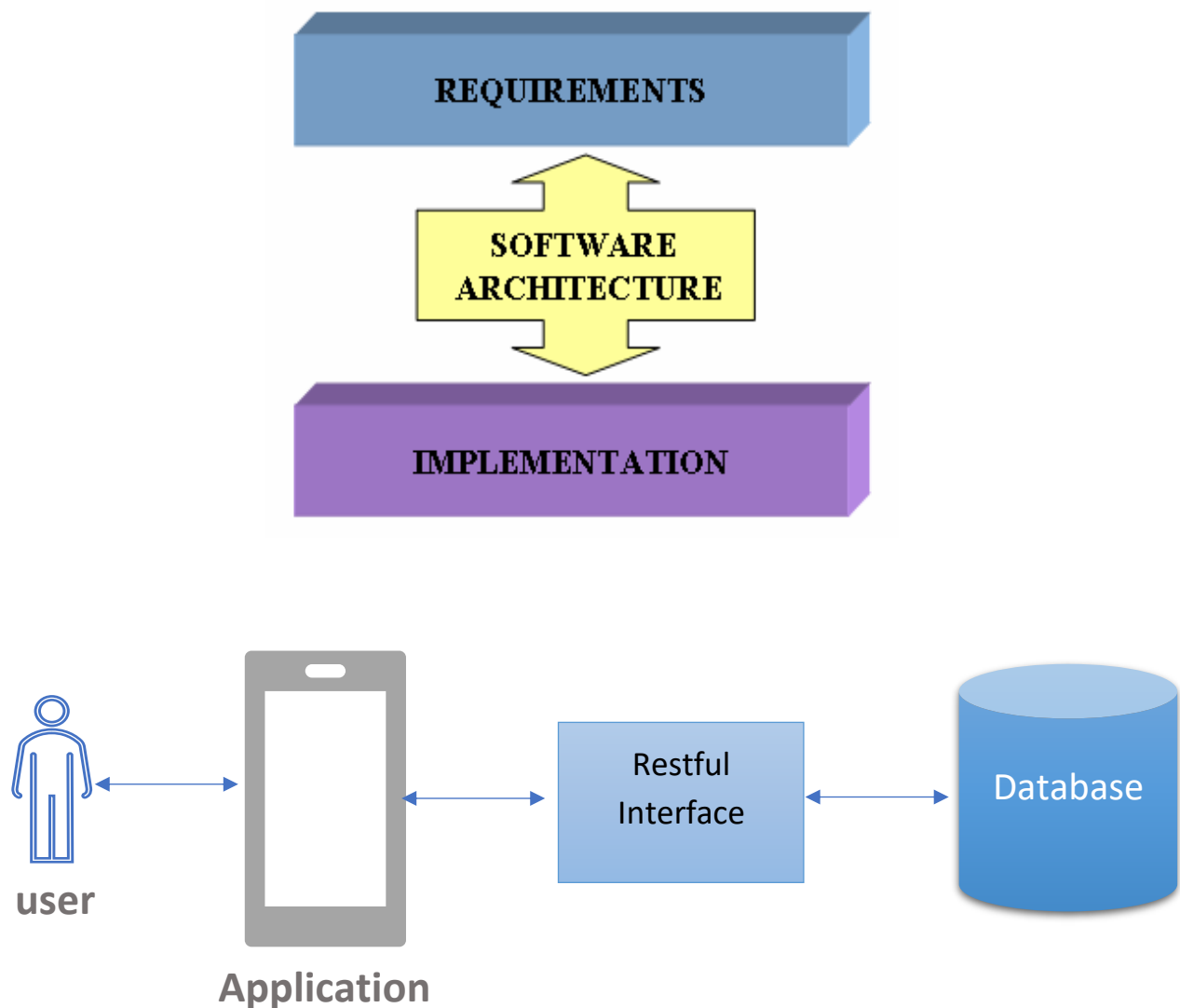
# Chapter 4:

# Implementation

## 4. Implementation

In this chapter we are going to discuss and go deeper in Recme mobile application's implementation, and present its code and the algorithms used to build it.

## 4.1 Software Architecture



## Login

```

32 void submit(context) async {
33   try {
34     setState(() {
35       isLoading = true;
36     });
37
38     if (password.text == adminPassword) {
39       try {
40         setState(() {
41           isLoading = true;
42         });
43
44         UserCredential result = await FirebaseAuth.instance
45           .signInWithEmailAndPassword(
46             email: email.text, password: password.text);
47         print(result);
48         Navigator.pushReplacementNamed(context, Dashboard.id);
49       } on PlatformException catch (error) {
50         var message = "Please Check Your Internet Connection ";
51         if (error.message != null) {
52           message = error.message;
53         }
54         _scaffoldKey.currentState.showSnackBar(
55           SnackBar(
56             content: Text(message.toString()),
57             duration: Duration(milliseconds: 800),
58           ), // SnackBar
59         );
60         setState(() {
61           isLoading = false;
62         });
63       }
64     } else {
65       setState(() {
66         isLoading = true;
67       });
68       UserCredential result = await FirebaseAuth.instance
69         .signInWithEmailAndPassword(
70           email: email.text, password: password.text);
71       print(result);
72       Navigator.pushReplacement(
73         context,
74         MaterialPageRoute(
75           builder: (ctx) => HomeScreen(),
76         ), // MaterialPageRoute
77       );
78     }
79   } catch (error) {
80     setState(() {
81       isLoading = false;
82     });
83     _scaffoldKey.currentState.showSnackBar(SnackBar(
84       content: Text(error.toString()),
85       duration: Duration(milliseconds: 800),
86     )); // SnackBar

```

## Reset Password

```

66 | | | | | MyButton(
67 | | | | |   name: 'Send Request',
68 | | | | |   onPressed: () {
69 | | | | |     auth.sendPasswordResetEmail(email: _email);
70 | | | | |     Navigator.of(context).pop();
71 | | | | |   },
72 | | | | | ) // MyButton

```

## Register

```

32 void submit(context) async {
33   UserCredential result;
34   try {
35     setState(() {
36       isLoading = true;
37     });
38     result = await FirebaseAuth.instance.createUserWithEmailAndPassword(
39       email: email.text, password: password.text);
40     print(result);
41   } on PlatformException catch (error) {
42     var message = "Please Check Your Internet Connection ";
43     if (error.message != null) {
44       message = error.message;
45     }
46     _scaffoldKey.currentState.showSnackBar(SnackBar(
47       content: Text(message.toString()),
48       duration: Duration(milliseconds: 600),
49     )); // SnackBar
50     setState(() {
51       isLoading = false;
52     });
53   } catch (error) {
54     setState(() {
55       isLoading = false;
56     });
57     _scaffoldKey.currentState.showSnackBar(SnackBar(
58       content: Text(error.toString()),
59       duration: Duration(milliseconds: 600),
60     )); // SnackBar
61     print(error);
62   }
63   FirebaseFirestore.instance.collection("User").doc(result.user.uid).set({
64     "UserName": userName.text,
65     "UserId": result.user.uid,
66     "UserEmail": email.text,
67     "UserNumber": phoneNumber.text,
68     "UserPassword": password.text,
69   });
70   Navigator.pushReplacement(
71     context, MaterialPageRoute(builder: (ctx) => HomeScreen()));
72   setState(() {
73     isLoading = false;
74   });
75 }
76
77 void validation() async {
78   if (userName.text.isEmpty &&
79     email.text.isEmpty &&
80     password.text.isEmpty &&
81     phoneNumber.text.isEmpty) {
82     _scaffoldKey.currentState.showSnackBar(
83       SnackBar(
84         content: Text('All fields Are Empty'),
85       ), // SnackBar
86     );
87   } else if (userName.text.length < 6) {
88     _scaffoldKey.currentState.showSnackBar(

```

## Update Profile

```

90   File _pickedImage;
91
92   PickedFile _image;
93   Future<void> getImage({ImageSource source}) async {
94     _image = await ImagePicker().getImage(source: source);
95     if (_image != null) {
96       setState(() {
97         _pickedImage = File(_image.path);
98       });
99     }
100  }
101
102  String userUid;
103
104  Future<String> _uploadImage({File image}) async {
105    StorageReference storageReference =
106      | FirebaseStorage.instance.ref().child("UserImage/$userUid");
107    StorageUploadTask uploadTask = storageReference.putFile(image);
108    StorageTaskSnapshot snapshot = await uploadTask.onComplete;
109    String imageUrl = await snapshot.ref.getDownloadURL();
110    return imageUrl;
111  }
112
113  void getUserUid() {
114    User myUser = FirebaseAuth.instance.currentUser;
115    userUid = myUser.uid;
116  }
117
118  bool centerCircle = false;
119  var imageMap;
120  void userDetailsUpdate() async {
121    setState(() {
122      centerCircle = true;
123    });
124    _pickedImage != null
125      ? imageMap = await _uploadImage(image: _pickedImage)
126      : Container();
127    FirebaseFirestore.instance.collection("User").doc(userUid).update({
128      "UserName": userName.text,
129      "UserEmail": userEmail.text,
130      "UserNumber": phoneNumber.text,
131      "UserPassword": userPassword.text,
132      "UserImage": imageMap,
133    });
134    setState(() {
135      centerCircle = false;
136    });
137    setState(() {
138      edit = false;
139    });
140  }

```



## Interest Book

```

14 class _InterestBookState extends State<InterestBook> {
15   int bookcount;
16   var jsonresponse;
17   String bookname;
18   Future<void> getBooks(bookname) async {
19     //List<Book> books = [];
20     //String url = "http://10.0.2.2:5000/api/get_rec?books={1:\'$bookname\'"}";
21     http.Response response = await http.get(
22       Uri.https(
23         '192.168.1.13:5000',
24         "/api/get_rec",
25         {'books': "{1:\'$bookname\'"}},
26       ),
27     );
28     if (response.statusCode == 200) {
29       setState(() {
30         jsonresponse = jsonDecode(response.body);
31         bookcount = jsonresponse.length;
32       });
33     } else {
34       print("Request failed with status: ${response.statusCode}");
35     }
36   }
37 }

```

## Recommend Book

```

18 class _RecommendationState extends State<Recommendation> {
19   int bookcount;
20   var jsonresponse;
21   String bookname;
22   String queryDec;
23   int count = 0;
24   Future<void> callUri(decString) async {
25     print("hello fro calURI");
26     print(decString);
27     http.Response response = await http.get(
28       Uri.https(
29         '192.168.1.13:5000',
30         "/api/get_rec",
31         {'books': "{$decString}"},
32       ),
33     );
34     if (response.statusCode == 200) {
35       setState(() {
36         jsonresponse = jsonDecode(response.body);
37         bookcount = jsonresponse.length;
38       });
39     } else {
40       print("Request failed with status: ${response.statusCode}");
41     }
42   }
43 }

```

## Search bar

```

16 class _SearchScreenState extends State<SearchScreen> {
17   int bookcount;
18   var jsonresponse;
19   String bookname;
20   Future<void> getBooks(bookname) async {
21     //List<Book> books = [];
22     //String url = "http://10.0.2.2:5000/api/get_rec?books={1:\'$bookname\'"}";
23     http.Response response = await http.get(
24       Uri.https(
25         '192.168.1.13:5000',
26         "/api/search",
27         {'bookName': "\'$bookname\'"},
28       ),
29     );
30     if (response.statusCode == 200) {
31       setState(() {
32         jsonresponse = jsonDecode(response.body);
33         bookcount = jsonresponse.length;
34       });
35     } else {
36       print("Request failed with status: ${response.statusCode}");
37     }
38   }
39 }

```

## Add to Favorite

```

166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200

```

```

onPressed: () {
  int bookIndex = allBooksToFav.indexWhere(
    (element) =>
      element["bookTitle"] == book.bTitle);
  if (bookIndex != -1) {
    _scaffoldKey.currentState.showSnackBar(
      SnackBar(
        content: Text(
          "Book already exist in favorite list"),
        ), // SnackBar
    );
  } else {
    _store
      .addBookToFavoriteList(
        bookIsbn: book.bIsbn,
        bookTitle: book.bTitle,
        bookImage: book.bImage,
        bookDescription: book.bDescription,
        bookAuthor: book.bAuthor,
        bookPublisher: book.bPublisher,
        bookAuthorImage: book.bAuthorImage,
        bookLanguage: book.bLanguage,
        bookYearOfPublication:
          book.byear_of_publication,
      )
      .then((value) {
        _scaffoldKey.currentState.showSnackBar(
          SnackBar(
            content:
              Text("Book Added Successfully"),
            ), // SnackBar
          );
      });
  }
},

```

## Add to Save

```

116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
onPressed: () {
  int bookIndex = allBooksToSave.indexWhere(
    (element) =>
      element["bookTitle"] == book.bTitle);
  if (bookIndex != -1) {
    _scaffoldKey.currentState.showSnackBar(
      SnackBar(
        content: Text("This Book already saved"),
      ), // SnackBar
    );
  } else {
    _store
      .addBookToSaveList(
        bookIsbn: book.bIsbn,
        bookTitle: book.bTitle,
        bookImage: book.bImage,
        bookDescription: book.bDescription,
        bookAuthor: book.bAuthor,
        bookPublisher: book.bPublisher,
        bookAuthorImage: book.bAuthorImage,
        bookLanguage: book.bLanguage,
        bookYearOfPublication:
          book.byear_of_publication,
      )
      .then((value) {
        _scaffoldKey.currentState.showSnackBar(
          SnackBar(
            content:
              Text("Book Saved Successfully"),
          ), // SnackBar
        );
      });
  }
},

```

## Send Feedback

```

141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
child: MyButton(
  name: "Send",
  onPressed: () async {
    if (!_formKey.currentState.validate()) {
      return;
    } else {
      _formKey.currentState.save();
      _store1
        .addFeedback(Contact(
          fName: name,
          fEmail: email,
          fMessage: message,
        )) // Contact
        .then((value) {
          _scaffoldKey.currentState.showSnackBar(
            SnackBar(
              content:
                Text("Message Send successfully"),
            ), // SnackBar
          );
        });
      feedbackProvider
        .addNotification("Notification");
      feedbackProvider.addCount();
    }
  },
), // MyButton

```

## Chat Room

```

9      return StreamBuilder(
10        stream: FirebaseFirestore.instance
11          .collection('Chat')
12          .orderBy('createdAt', descending: true)
13          .snapshots(),
14        builder: (context, snapshot) {
15          if (snapshot.connectionState == ConnectionState.waiting) {
16            return Center(child: CircularProgressIndicator());
17          }
18          final docs = snapshot.data.docs;
19          final user = FirebaseAuth.instance.currentUser;
20          return ListView.builder(
21            reverse: true,
22            itemCount: docs.length,
23            itemBuilder: (context, index) => MessageBubble(
24              docs[index]['text'],
25              docs[index]['username'],
26              docs[index]['userimage'],
27              docs[index]['userId'] == user.uid,
28              key: ValueKey(docs[index].documentID),
29            ), // MessageBubble
30          ); // ListView.builder
31        },
32      ); // StreamBuilder

```

## Add Book

```

294      MyButton(
295        name: 'Add Book',
296        onPressed: () async {
297          if (!_formKey.currentState.validate()) {
298            return;
299          } else {
300            _formKey.currentState.save();
301            _store
302              .addBook(Book(
303                bIsbn: _isbn,
304                bImage: _image,
305                bTitle: _title,
306                bDescription: _description,
307                bAuthor: _author,
308                byear_of_publication: _year_of_publication,
309                bPublisher: _publisher,
310                bLanguage: _language,
311                bAuthorImage: _authorImage,
312              )) // Book
313              .then((value) {
314                _scaffoldKey.currentState.showSnackBar(
315                  SnackBar(
316                    content: Text("Book added successfully"),
317                  ), // SnackBar
318                );
319              });
320          }
321        },
322      ), // MyButton

```

## Edit Book

```

309 |
310 | MyButton(
311 |   name: 'Edit Book',
312 |   onPressed: () async {
313 |     if (!_formKey.currentState.validate()) {
314 |       return;
315 |     } else {
316 |       _formKey.currentState.save();
317 |       _store
318 |         .editBook(
319 |           ({
320 |             kBookIsbn: _isbn,
321 |             kBookImage: _image,
322 |             kBookTitle: _title,
323 |             kBookDescription: _description,
324 |             kBookAuthor: _author,
325 |             kBookYearOfPublication:
326 |               _year_of_publication,
327 |             kBookPublisher: _publisher,
328 |             kBookLanguage: _language,
329 |             kBookAuthorImage: _authorImage,
330 |           }),
331 |           book.bId)
332 |         .then((value) {
333 |           _scaffoldKey.currentState.showSnackBar(
334 |             SnackBar(
335 |               backgroundColor: Colors.black45,
336 |               content: Text("Book edited successfully"),
337 |             ), // SnackBar
338 |           );
339 |         });
340 |     },
341 |   ), // MyButton

```

## Delete Book

```

98 |
99 |   onPressed: () {
100 |     _store.deleteBook(book.bId);
101 |     Navigator.pushNamed(context, ViewBook.id)
102 |       .then((value) {
103 |         _scaffoldKey.currentState.showSnackBar(
104 |           SnackBar(
105 |             content:
106 |               Text("Book deleted successfully"),
107 |           ), // SnackBar
108 |         );
109 |       });
110 |   },

```

## API

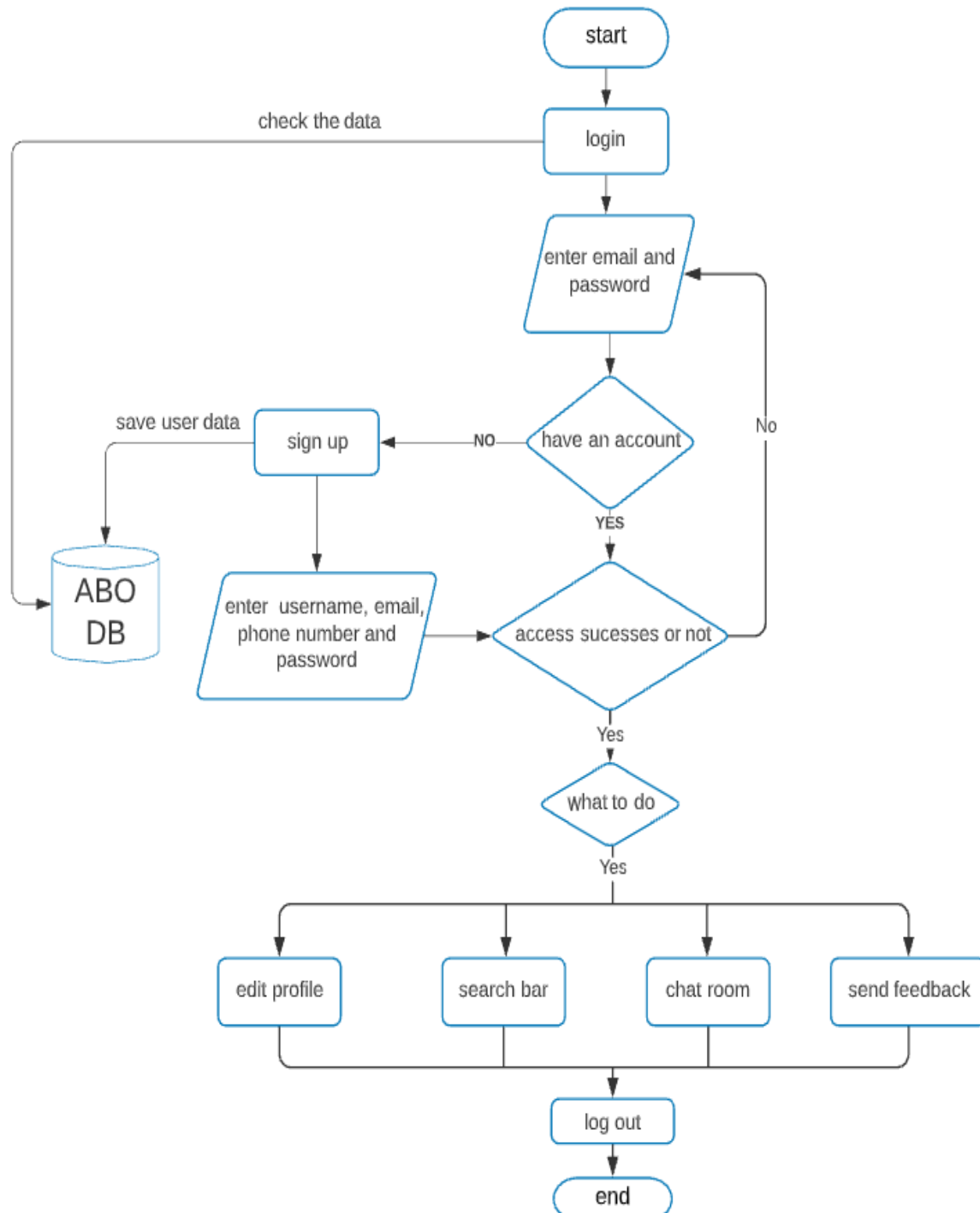
```

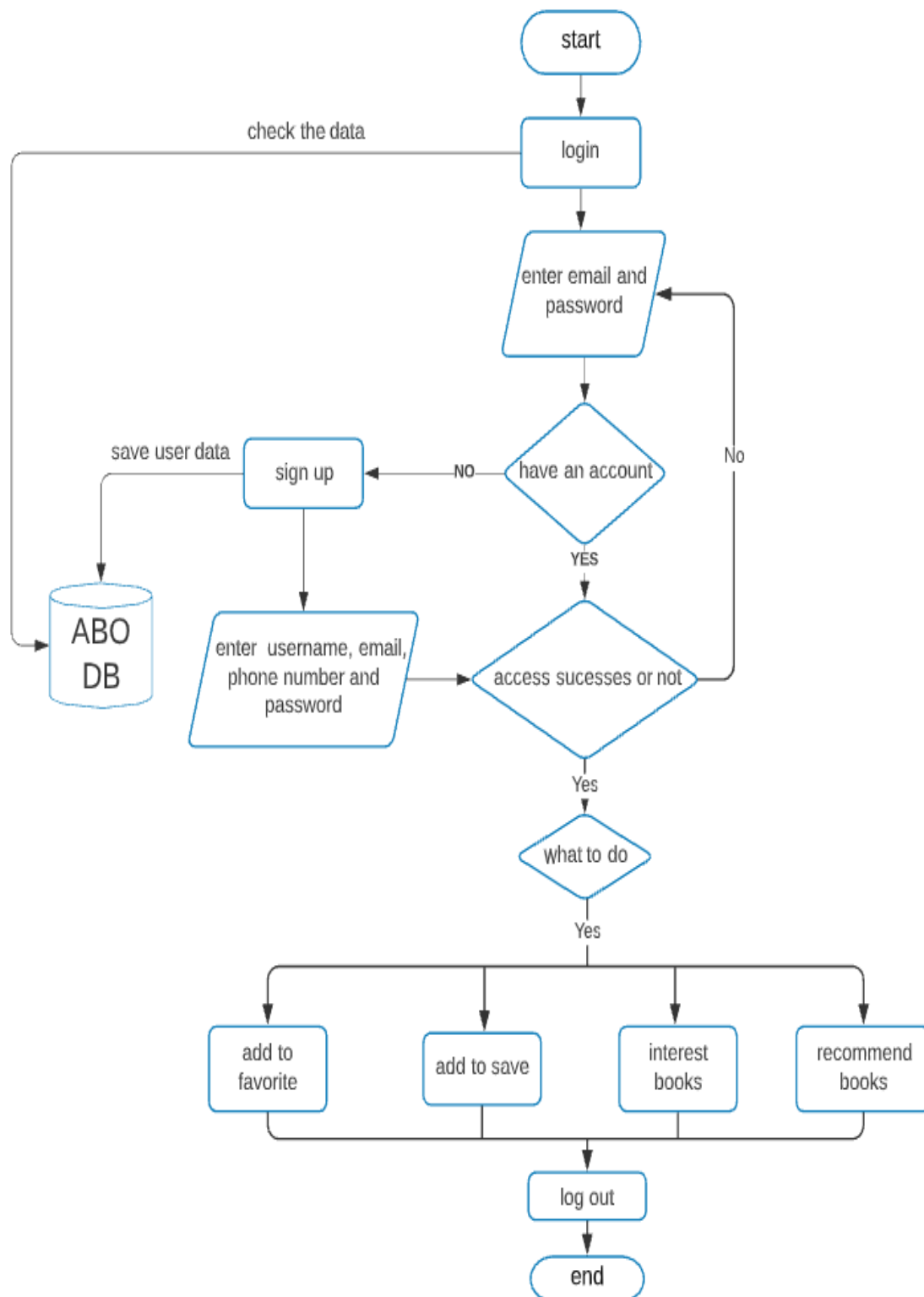
82     plt.show()
83     users.Age.hist(bins=[0, 10, 20, 30, 40, 50, 100])
84     plt.title('Age Distribution\n')
85     plt.xlabel('Age')
86     plt.ylabel('Count')
87     #plt.savefig('system2.png', bbox_inches='tight')
88     #plt.show()
89     counts1 = ratings['userID'].value_counts()
90     ratings = ratings[ratings['userID'].isin(counts1[counts1 >= 200].index)]
91     counts = ratings['bookRating'].value_counts()
92     ratings = ratings[ratings['bookRating'].isin(counts[counts >= 100].index)]
93     combine_book_rating = pd.merge(ratings, books, on='ISBN')
94     columns = ['yearOfPublication', 'publisher', 'bookAuthor', 'imageUr15', 'imageUr1M', 'imageUr1L']
95     combine_book_rating = combine_book_rating.drop(columns, axis=1)
96     combine_book_rating.head()
97     combine_book_rating = combine_book_rating.dropna(axis = 0, subset = ['bookTitle'])
98
99     book_ratingCount = (combine_book_rating.
100         groupby(by = ['bookTitle'])['bookRating'].
101         count().
102         reset_index().
103         rename(columns = {'bookRating': 'totalRatingCount'}))
104     [['bookTitle', 'totalRatingCount']]
105
106     book_ratingCount.head()
107     rating_with_totalRatingCount = combine_book_rating.merge(book_ratingCount, left_on = 'bookTitle', right_on = 'bookTitle', how = 'left')
108     rating_with_totalRatingCount.head()
109     pd.set_option('display.float_format', lambda x: '%.3f' % x)
110     popularity_threshold = 35
111     rating_popular_book = rating_with_totalRatingCount.query('totalRatingCount >= @popularity_threshold')
112     rating_popular_book.head()
113     rating_popular_book.shape
114     combined = rating_popular_book.merge(users, left_on = 'userID', right_on = 'userID', how = 'left')
115     from scipy.sparse import csr_matrix
116     us_canada_user_rating = combined[combined['Location'].str.contains("canada|india|egypt")]
117     us_canada_user_rating=us_canada_user_rating.drop('Age', axis=1)
118     us_canada_user_rating.head()
119     us_canada_user_rating = us_canada_user_rating.drop_duplicates(['userID', 'bookTitle'])
120     us_canada_user_rating_pivot = us_canada_user_rating.pivot(index = 'bookTitle', columns = 'userID', values = 'bookRating').fillna(0)
121     us_canada_user_rating_matrix = csr_matrix(us_canada_user_rating_pivot.values)
122     #print(us_canada_user_rating_matrix)
123
124     #getting recommendation for each book
125     for j in range(0,len(test_array)):
126         query_index = test_array[j]
127         model_knn = NearestNeighbors(metric = 'cosine', algorithm = 'brute')
128         model_knn.fit(us_canada_user_rating_matrix)
129         us_canada_user_rating_pivot.iloc[0,:].values.reshape(1,-1)
130         distances, indices = model_knn.kneighbors(us_canada_user_rating_pivot.loc[query_index,:].values.reshape(1, -1), n_neighbors = 20)
131         us_canada_user_rating_pivot.loc[query_index]
132         for i in range(0, len(distances.flatten())):
133             respond =np.append(respond,indices)
134
135     respond = np.unique(respond, axis=0)
136     #convert respond list to dictionary of books
137     bookslist = pd.read_csv('BX-Books.csv', sep=';', error_bad_lines=False, encoding="latin-1")
138     bookslist.columns = ['ISBN', 'bookTitle', 'bookAuthor', 'yearOfPublication', 'publisher', 'imageUr15', 'imageUr1M', 'imageUr1L']
139     #bookslist=bookslist.drop_duplicates(subset="bookTitle")
140     print("this is test")
141
142     #testDic={}
143     rows = pd.DataFrame()
144     for i in range (0,len(respond)):
145
146         row=(bookslist.loc[bookslist['bookTitle'] == us_canada_user_rating_pivot.index[respond.flatten()[i]]])
147         row.reset_index(drop=True, inplace=True)
148         #testDic[i]=row.to_dict('index')
149         rows = rows.append(row)
150         #print(rows)
151
152     else:
153         return "Error: No id field provided. Please specify an id."
154
155     #query = {'book'+str(i): query_list[i] for i in range(0, 2, 1)}
156     #testDic[re] =rows.to_dict('index')
157     #print(rows)
158     rows = rows.sample(frac=1)
159     rows.reset_index(drop=True, inplace=True)
160
161     return jsonify(rows.to_dict('records'))
162     #return jsonify(query)

```

## 4.2 Flow Chart

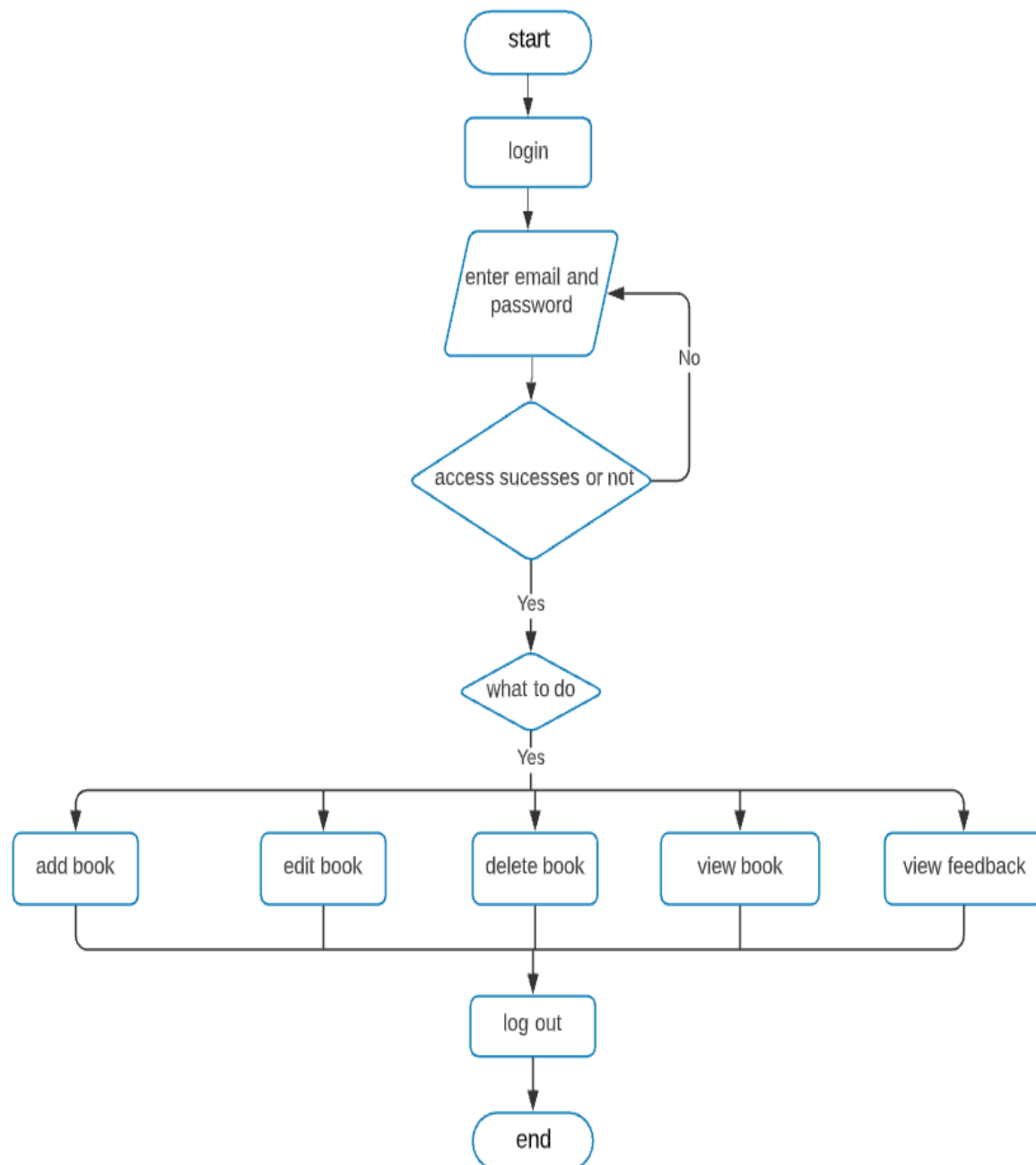
### User flow chart







## Admin flow chart

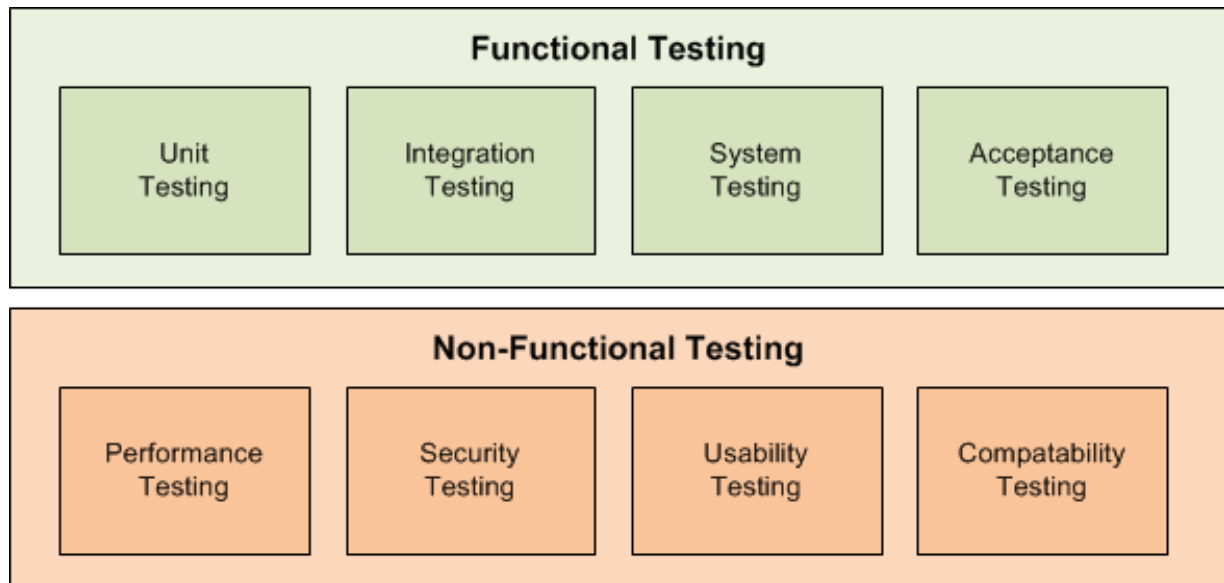


# Chapter 5:

# Testing

## 5. Testing

In this chapter we are going to discuss and go deeper in Recme mobile application's testing, and present the types of testing to be used and test cases we examined our application through.



## Functional Testing:

### 5.1 Unit Testing

Testing of individual items (e.g., modules, programs, objects, classes, etc.) Usually as part of the coding phase, in isolation from other development item sand the system as a whole.

### 5.2 Integration Testing

Testing the interfaces between major (e.g., systems level application modules) and minor (e.g., individual programs or components) items within an application which must interact with each other.

### System Testing

Testing a system behavior as a whole when development is finished, and the system can be tested as complete entity.

## **Regression Testing**

To check older functionality after integrating new functionality.

## **Acceptance Testing**

Testing to ensure that a development is ready to be deployed into the business, operational or production environment.

## **Non-Functional Testing:**

### **Performance Testing**

Accomplished a designated function regarding processing time and through put rate.

### **Load Testing**

Measuring the behavior of within creasing load which can be handled by the component or system.

### **Stress Testing**

Evaluate a system or component at or beyond the limits of its specified requirements.

### **Security Testing**

Testing how well the system protects against unauthorized internal or external access.

## 5.3 Additional Testing

### Test Cases:

- Test scenario objective:  
Verify log in successfully with email and password
- Assumptions/Dependencies  
Valid email: [asmaa12@gmail.com](mailto:asmaa12@gmail.com)  
Valid password: 123456789

Step#	Description	Expected Result	Actual Result	Error Type
1	Go to log in screen	User can go to log in screen through splash screen		
2	Enter email: asmaa12@gmail.com	User can enter email		
3	Press Tab	User can move to the text field of password		
4	Enter password: 123456789	User can enter password		
5	Click on log in button	Log in is successfully done And Home page will open	Passed	

- **Test scenario objective:**

A message should appear to user to tell him/her to enter valid email when the entered email is not in firebase

- **Assumptions/Dependencies**

Invalid email: [v@gamil.com](mailto:v@gamil.com)

Valid password: 123456789

Step#	Description	Expected Result	Actual Result	Error Type
1	Go to log in screen	User can go to log in screen through splash screen		
2	Enter email: v@gmail.com	User can enter email		
3	Press Tab	User can move to the text field of password		
4	Enter password: 123456789	User can enter password		
5	Click on log in button	Message will appear to user telling him/her that the email entered is not valid And User is still in log in page as there is no user with this email in firebase.	Passed	

- **Test scenario objective:**

A message should appear to user to tell him/her to enter valid password when the entered password is not in firebase.

- **Assumptions/Dependencies**

Valid email: [asmaa12@gamil.com](mailto:asmaa12@gamil.com)

Invalid password: 232342134

Step#	Description	Expected Result	Actual Result	Error Type
1	Go to log in screen	User can go to log in screen through splash screen		
2	Enter email: asmaa12@gmail.com	User can enter email		
3	Press Tab	User can move to the text field of password		
4	Enter password: 232342134	User can enter password		
5	Click on log in button	Message will appear to user telling him/her that the password entered is not valid User is still in log in page as there is no user with this password in firebase	Passed	

- **Test scenario objective:**  
Change password if user forget it through forget password
- **Assumptions/Dependencies**  
Valid email: [asmaa12@gmail.com](mailto:asmaa12@gmail.com)

Step#	Description	Expected Result	Actual Result	Error Type
1	Go to log in screen	User can go to log in screen through splash screen		
2	Select forget password	Forget password screen appears		
3	Enter valid email: asmaa12@gmail.com	User can enter email		
4	Press send request	Request will be sent to the entered mail to tell him to enter new password	passed	



- Test scenario objective:  
Register successfully
- Assumptions/Dependencies  
Valid Username: Asmaamohamed  
Valid Phone number: 01123456789  
Valid email: [asmaa12@gamil.com](mailto:asmaa12@gamil.com)  
Valid password: 123456789

Step#	Description	Expected Result	Actual Result	Error type
1	Go to Registration screen	User can go to registration screen		
2	Enter username: Asmaamohamed	User can enter username		
3	Press tap	User can go to the next field		
4	Enter email: asmaa12@gmail.com	User can enter email		
5	Press tap	User can go to the next field		
6	Enter phone number: 01123456789	User can enter phone number		
7	Press tab	User can go to the next field		
8	Enter password: 123456789	User can enter password		
9	Click on register button	User will be successfully registered, and home screen will be opened	Passed	

- **Test scenario objective:**

Register with username with more than 8 characters

- **Assumptions/Dependencies**

Invalid Username: Asmaamohamedd

Valid Phone number: 01123456789

Valid email: [asmaa12@gamil.com](mailto:asmaa12@gamil.com)

Valid password: 123456789

Step#	Description	Expected Result	Actual Result	Error type
1	Go to Registration screen	User can go to registration screen		
2	Enter username: Asmaamohamedd	User can enter username		
3	Press tap	User can go to the next field		
4	Enter email: asmaa12@gmail.com	User can enter email		
5	Press tap	User can go to the next field		
6	Enter phone number: 01123456789	User can enter phone number		
7	Press tab	User can go to the next field		
8	Enter password: 123456789	User can enter password		
9	Click on register button	User will not be registered, and message appeared to tell user to enter username with characters not more than 8 and less than 6	Passed	

- **Test scenario objective:**

Register with username with less than 6 characters

- **Assumptions/Dependencies**

Invalid Username: Asma

Valid Phone number: 01123456789

Valid email: [asmaa12@gamil.com](mailto:asmaa12@gamil.com)

Valid password: 123456789

Step#	Description	Expected Result	Actual Result	Error type
1	Go to Registration screen	User can go to registration screen		
2	Enter username: Asma	User can enter username		
3	Press tap	User can go to the next field		
4	Enter email: asmaa12@gmail.com	User can enter email		
5	Press tap	User can go to the next field		
6	Enter phone number: 01123456789	User can enter phone number		
7	Press tab	User can go to the next field		
8	Enter password: 123456789	User can enter password		
9	Click on register button	User will not be registered, and message appeared to tell user to enter username with characters not more than 8 and less than 6	Passed	

- **Test scenario objective:**

Register with phone number with more than 11 numbers

- **Assumptions/Dependencies**

Valid Username: Asmaamohamed

invalid Phone number: 01123456789

Valid email: [asmaa12@gamil.com](mailto:asmaa12@gamil.com)

Valid password: 123456789

Step#	Description	Expected Result	Actual Result	Error type
1	Go to Registration screen	User can go to registration screen		
2	Enter username: Asmaamohamed	User can enter username		
3	Press tap	User can go to the next field		
4	Enter email: asmaa12@gmail.com	User can enter email		
5	Press tap	User can go to the next field		
6	Enter phone number: 011123456789	User can enter phone number		
7	Press tab	User can go to the next field		
8	Enter password: 123456789	User can enter password		
9	Click on register button	User will not be registered, and message appeared to tell user to enter phone number which is equal to 11 number	Passed	

- **Test scenario objective:**

Register with phone number with less than 11 numbers

- **Assumptions/Dependencies**

Valid Username: Asmaamohamed

invalid Phone number: 01123456789

Valid email: [asmaa12@gamil.com](mailto:asmaa12@gamil.com)

Valid password: 123456789

Step#	Description	Expected Result	Actual Result	Error type
1	Go to Registration screen	User can go to registration screen		
2	Enter username: Asmaamohamed	User can enter username		
3	Press tap	User can go to the next field		
4	Enter email: asmaa12@gmail.com	User can enter email		
5	Press tap	User can go to the next field		
6	Enter phone number: 01123456789	User can enter phone number		
7	Press tab	User can go to the next field		
8	Enter password: 123456789	User can enter password		
9	Click on register button	User will not be registered, and message appeared to tell user to enter phone number which is equal to 11 number	Passed	

- **Test scenario objective:**

Register with invalid email format

- **Assumptions/Dependencies**

Valid Username: Asmaamohamed

Valid Phone number: 01123456789

invalid email: asmaa122134

Valid password: 123456789

Step#	Description	Expected Result	Actual Result	Error type
1	Go to Registration screen	User can go to registration screen		
2	Enter username: Asmaamohamed	User can enter username		
3	Press tap	User can go to the next field		
4	Enter email: asmaa122134	User can enter email		
5	Press tap	User can go to the next field		
6	Enter phone number: 01123456789	User can enter phone number		
7	Press tab	User can go to the next field		
8	Enter password 123456789	User can enter password		
9	Click on register button	User will not be registered, and message appeared to tell user to enter valid email	Passed	

- **Test scenario objective:**  
Register with short password (less than 8)
- **Assumptions/Dependencies**  
Valid Username: Asmaamohamed  
Valid Phone number: 01123456789  
Valid email: [asmaa12@gmail.com](mailto:asmaa12@gmail.com)  
invalid password: 1234567

Step#	Description	Expected Result	Actual Result	Error type
1	Go to Registration screen	User can go to registration screen		
2	Enter username: Asmaamohamed	User can enter username		
3	Press tap	User can go to the next field		
4	Enter email: asmaa12@gmail.com	User can enter email		
5	Press tap	User can go to the next field		
6	Enter phone number: 01123456789	User can enter phone number		
7	Press tab	User can go to the next field		
8	Enter password:1234567	User can enter password		
9	Click on register button	User will not be registered, and message appeared to tell user to enter password with 8 or more characters	Passed	

- **Test scenario objective:**

User updates his profile data

- **Assumptions/Dependencies**

Choose a profile picture

Valid Username: Asmaamohamed

Valid Phone number: 01123456789

Valid password: 123456789

Step#	Description	Expected Result	Actual Result	Error type
1	Select profile from slide list	User can go to update profile screen		
2	Press on edit profile button	Edit screen appeared		
3	Press on camera icon	2 Choices appeared		
4	Select take photo from camera/gallery	User can choose whether to choose new photo from camera or gallery		
5	Enter new username: Asmaamohamed	User can enter new username		
6	Press tab	User can go to the next field		
7	Enter new phone number: 01123456789	User can enter new Phone number		
8	Press tab	User can go to the next field		
9	Enter new password: 123456789	User can enter new password		
10	Press on mark button	User profile updated successfully	passed	



- **Test scenario objective:**

Message appeared to tell user that he entered a username less than 6 characters

- **Assumptions/Dependencies**

Choose a profile picture

Invalid Username: Asma

Valid Phone number: 01123456789

Valid password: 123456789

Step#	Description	Expected Result	Actual Result	Error type
1	Select profile from slide list	User can go to update profile screen		
2	Press on edit profile button	Edit screen appeared		
3	Press on camera icon	2 Choices appeared		
4	Select take photo from camera/gallery	User can choose whether to choose new photo from camera or gallery		
5	Enter new username: Asma	User can enter new username		
6	Press tab	User can go to the next field		
7	Enter new phone number: 01123456789	User can enter new Phone number		
8	Press tab	User can go to the next field		
9	Enter new password: 123456789	User can enter new password		
10	Press on mark button	User profile not updated as new username is less than 6 characters	passed	

- **Test scenario objective:**

Message appeared to tell user that he entered a phone number less than 11 numbers

- **Assumptions/Dependencies**

Choose a profile picture

Valid Username: Asmaamohamed

invalid Phone number: 0111233

Valid password: 123456789

Step#	Description	Expected Result	Actual Result	Error type
1	Select profile from slide list	User can go to update profile screen		
2	Press on edit profile button	Edit screen appeared		
3	Press on camera icon	2 Choices appeared		
4	Select take photo from camera/gallery	User can choose whether to choose new photo from camera or gallery		
5	Enter new username: Asmaamohamed	User can enter new username		
6	Press tab	User can go to the next field		
7	Enter new phone number: 0111233	User can enter new Phone number		
8	Press tab	User can go to the next field		
9	Enter new password: 123456789	User can enter new password		
10	Press on mark button	User profile not updated as new phone number is less than 11 numbers	passed	

- **Test scenario objective:**

Message appeared to tell user that he entered a password less than 8 numbers

- **Assumptions/Dependencies**

Choose a profile picture

Valid Username: Asmaamohamed

Valid Phone number: 01123456789

invalid password: 12345

Step#	Description	Expected Result	Actual Result	Error type
1	Select profile from slide list	User can go to update profile screen		
2	Press on edit profile button	Edit screen appeared		
3	Press on camera icon	2 Choices appeared		
4	Select take photo from camera/gallery	User can choose whether to choose new photo from camera or gallery		
5	Enter new username: Asmaamohamed	User can enter new username		
6	Press tab	User can go to the next field		
7	Enter new phone number: 0112345678	User can enter new Phone number		
8	Press tab	User can go to the next field		
9	Enter new password: 12345	User can enter new password		
10	Press on mark button	User profile not updated as new password is less than 8 characters	passed	

- **Test scenario objective:**

User can send feedback to admin through contact us feature

- **Assumptions/Dependencies**

Name: Asmaamohamed

Email: [asmaa12@gmail.com](mailto:asmaa12@gmail.com)

Message: this app is awesome

Step#	Description	Expected Result	Actual Result	Error type
1	Select contact us from slide list	User can go to contact us screen		
2	Enter Name: Asmaamohamed	User can enter his name		
3	Press tab	User can go to the next field		
4	Enter email: asmaa12@gmail.com	User can enter his/her email		
5	Press tab	User can go to the next field		
6	Enter message: This app is awesome	User can enter message		
7	Press send button	Feedback will be sent to admin successfully	passed	

- **Test scenario objective:**

Open interest page to enter the book you want to get similar books to it

- **Assumptions/Dependencies**

Valid book name: Animal Farm

Step#	Description	Expected Result	Actual Result	Error type
1	Select interest from slide list	User can go to interest screen		
2	Enter book name: Animal Farm	User can enter book name		
3	Press Recommend button	Similar books to the entered book name will appear to user	passed	

- **Test scenario objective:**

Open search page to search for a specific book

- **Assumptions/Dependencies**

Valid book name: Animal Farm

Step#	Description	Expected Result	Actual Result	Error type
1	Press on search icon in the home page	User can go to search screen		
2	Enter book name: Animal Farm	User can enter book name		
3	Press Search button	Results of search will appear to user	passed	

- **Test scenario objective:**  
Login as admin successfully
- **Assumptions/Dependencies**  
Valid email: [admin@gmail.com](mailto:admin@gmail.com)  
Valid password: admin1234

Step#	Description	Expected Result	Actual Result	Error Type
1	Go to log in screen	Admin can go to log in screen through splash screen		
2	Enter email: admin@gmail.com	Admin can enter email		
3	Press Tab	Admin can move to the text field of password		
4	Enter password: admin1234	Admin can enter password		
5	Click on log in button	Log in is successfully done And admin page will be successfully opened	Passed	

- **Test scenario objective:**

Admin can add new book to the firebase

- **Assumptions/Dependencies**

**Valid Imageurl:**

<http://images.amazon.com/images/P/0195153448.01.LZZZZZZZ.jpg>

**Valid ISBN:** 0195153448

**Valid Title:** Classical Mythology

**Valid Description:** Building on the bestselling tradition of previous editions, *Classical Mythology*, Tenth Edition, is the most comprehensive survey of classical mythology available--and the first full-color textbook of its kind.

**Valid Author:** Mark P. O. Morford

**Valid Year of Publication:** 2002

**Valid Publisher:** Oxford University

**Valid Language:** English

Step#	Description	Expected Result	Actual Result	Error type
1	Click on add book button	Admin can go to add book screen		
2	Enter imageurl: <a href="http://images.amazon.com/images/P/0195153448.01.LZZZZZZZ.jpg">http://images.amazon.com/images/P/0195153448.01.LZZZZZZZ.jpg</a>	Admin can enter Imageurl		
3	Press tap	Admin can go to the next field		
4	Enter ISBN: 0195153448	Admin can enter ISBN		
5	Press tap	Admin can go to the next field		

6	Enter title: Classical Mythology	Admin can enter title		
7	Press tab	Admin can go to the next field		
8	Enter Description: Building on the bestselling tradition of previous editions, <i>Classical Mythology</i> , Tenth Edition, is the most comprehensive survey of classical mythology available-- and the first full-color textbook of its kind.	Admin can enter description		
9	Press tab	Admin can go to the next field		
10	Enter Author: Mark P. O. Morford	Admin can enter author		
11	Press tab	Admin can go to the next field		
12	Enter Year of Publication: 2002	Admin can enter Year of Publication		
13	Press tab	Admin can go to the next field		
14	Enter publisher: Oxford University	Admin can enter publisher		
15	Press tab	Admin can go to the next field		
16	Enter language: English	Admin can enter language		
17	Press Add book button	Book added successfully to firebase	passed	



- **Test scenario objective:**

Admin can edit existing book in firebase

- **Assumptions/Dependencies**

**Valid Imageurl:**

<http://images.amazon.com/images/P/0195153448.01.LZZZZZZZ.jpg>

**Valid ISBN:** 0195153448

**Valid Title:** Classical Mythology

**Valid Description:** Building on the bestselling tradition of previous editions, *Classical Mythology*, Tenth Edition, is the most comprehensive survey of classical mythology available--and the first full-color textbook of its kind.

**Valid Author:** Mark P. O. Morford

**Valid Year of Publication:** 2002

**Valid Publisher:** Oxford University

**Valid Language:** English

Step#	Description	Expected Result	Actual Result	Error type
1	Click on add view book button	Admin can go to view book screen		
2	Select book to edit	Book details page appear		
2	Enter new imageurl: <a href="http://images.amazon.com/images/P/0195153448.01.LZZZZZZZ.jpg">http://images.amazon.com/images/P/0195153448.01.LZZZZZZZ.jpg</a>	Admin can enter new Imageurl		
3	Press tap	Admin can go to the next field		
4	Enter new ISBN: 0195153448	Admin can enter new ISBN		
5	Press tap	Admin can go to the next field		

6	Enter new title: Classical Mythology	Admin can enter new title		
7	Press tab	Admin can go to the next field		
8	Enter new Description: Building on the bestselling tradition of previous editions, <i>Classical Mythology</i> , Tenth Edition, is the most comprehensive survey of classical mythology available-- and the first full-color textbook of its kind.	Admin can enter new description		
9	Press tab	Admin can go to the next field		
10	Enter new Author: Mark P. O. Morford	Admin can enter new author		
11	Press tab	Admin can go to the next field		
12	Enter new Year of Publication: 2002	Admin can enter new Year of Publication		
13	Press tab	Admin can go to the next field		
14	Enter new publisher: Oxford University	Admin can enter new publisher		
15	Press tab	Admin can go to the next field		
16	Enter new language: English	Admin can enter new language		
17	Press Edit book button	Book edited successfully	passed	

# Chapter 6:

## Result and Discussion

### 6. Result and Discussion

In this chapter we are going to find out the results of the project whether they are achieved or not and the differences between the desired results and the actual ones.

## 6.1 Results

### 6.1.1 Expected Result

- Find new and interesting books by browsing personalized recommendations based on books you have read and your favorite genres.
- The application enables user to search about books that play an important role in the discovery of relevant books.
- The application contains the part of social media to make users communicate with each other in chat rooms and exchange experience.

### 6.1.2 Actual Result

- The application help readers to find appropriate book to read next.
- The application enables users to discover books depend on their book interests to recommend books the user would be tempted to read as they are matched with his/her choices.
- The application enables user to search about books by book name to display all books that have the same name that play an important role in the discovery of relevant books.
- The application contains the part of social media to make users communicate with each other in chat rooms and exchange experience.
- It has the part of ranking the result of the search according to your priorities.
- The application delays some second during getting respond from local server because of weak capabilities.
- It has property Dark mode that can prolong the battery life of your smartphone and protects eyes from bright screens that hurt the eyes from hours of looking at a bright screen.

## 6.2 Discussion

There is a little difference between the expected and actual result except because of weakness of local server for allowing to reading books and see what your friends are interesting.

We have added new features into our application such as: Chatting and reporting.

# Chapter 7: Conclusion

## 7. Conclusion

Eventually, during this project we tried to create the most effective we will avoid exploitation and it is simply the way of expressing our duty towards our society as we felt accountable enough to participate in creating this community a stronger place with one amongst the foremost powerful tools we can use and that we selected as our finding out field years past that is **TECHNOLOGY**.

In summary, our society would be a lot of convenient if we tend to get obviate exploitation and more individuals used this application. we should always still try and facilitate individuals create the most effective decisions with no got to any reasonably blackmail and that we tried to try and do that with our restricted fund and resources, and we are certain we are planning to expand this application and enhance its performance and fame if we are simply given the correct resources and fund

Finally, realizing the fact that “**a few clicks**” will do tons to a society, will certainly create US a lot of responsive to the technology role and become higher versions of ourselves.

# Chapter 8:

## Future Work

### 8. Future Work

We are going to add more and more features to our application such as:

1. Recommend a book to a friend by Seeing what your friends are interesting to allowing to you virtually peruse your friends' bookshelves. The app will allow you to save and share the recommendations of your friends so neither of you will ever be at a loss for what to read next.
2. Chat Room for each book category to make it easier for users to chat in the chat room designated for each book category.
3. Using a new and updated datasets to discover new books.
4. Owning our data instead of using historical amazon datasets.

5. Having rights to show books content to users.
6. Enabling reading and buying books inside our app.
7. Increase information about recommendation books.
8. We also are going to enhance its performance and try deploying it on any of the available stores.

So, we will not miss any chance to keep working as a team on that project and enhancing it after graduation as we are looking forward to turning this application into a start-up.

Our dream started with this idea months ago, and we will insist on not letting go of it and not letting it be just a dream.

## Bibliography

<https://www.kdnuggets.com/2019/09/machine-learning-recommender-systems.html>

[https://en.wikipedia.org/wiki/Collaborative\\_filtering](https://en.wikipedia.org/wiki/Collaborative_filtering)

<https://firebase.flutter.dev/docs/overview/>

<https://flutter.dev/>

<https://api.flutter.dev/>

<https://www.fullstackpython.com/flask.html>

<https://www.amazon.com/kindle-dbs/fd/kcp>

<https://www.goodreads.com/>

<http://rushrash.com/domain-requirement/>

[https://en.wikipedia.org/wiki/User\\_requirements\\_document](https://en.wikipedia.org/wiki/User_requirements_document)

<http://www2.informatik.uni-freiburg.de/~chiegler/BX/>

[https://www.youtube.com/channel/UCNU\\_IIfiiWBdtULKOW6X0Dig](https://www.youtube.com/channel/UCNU_IIfiiWBdtULKOW6X0Dig)

<https://programminghistorian.org/en/lessons/creating-apis-with-python-and-flask>

<https://www.smartdraw.com/sequence-diagram/>

[https://en.m.wikipedia.org/wiki/Cost\\_estimate](https://en.m.wikipedia.org/wiki/Cost_estimate)



**“If you do not like to read  
you have not found the right book.”**



**RECME**  
*GET AND DISCOVER*

**Thank You!**