Chapter Chatters

SYNOPSIS

By

JASON P.N KHONGWIR

Student Id: DC2021MCA0014

MASTER OF COMPUTER APPLICATIONS

DEPARTMENT OF COMPUTER APPLICATIONS ASSAM

DON BOSCO UNIVERSITY SCHOOL OF TECHNOLOGY,

AZARA, GUWAHATI

781 017,ASSAM, INDIA.

MCA (2021-2023)

Contents

Chapter 4 Conclusion

References

Chapter 1.

INTRODUCTION

The project is a book application that allows users to keep track of the books they own, join book clubs, participate in discussions and write reviews. The application will provide users with a platform to connect with other book lovers, share their thoughts and opinions on books, and find new books to read.

The application will be built using Flutter and Firebase, ensuring an efficient and scalable solution.

The aim of this project is to create a user-friendly and engaging platform that encourages book lovers to read and connect with others who share similar interests. The application will be designed to meet the needs and requirements of book lovers, making it an attractive and accessible resource for those who love to read.

1.1 PROJECT TITLE

Chapter Chatters.

1.2 OBJECTIVE

- To provide an online platform enabling users to manage their bookshelves and keep track of the books they have read, are reading, and wish to read.

- To create a community in which people may join book clubs and engage in book discussions and reviews.

- Allow people to find and read new books based on their interests and preferences.

- To provide a method for users to rank and review books, as well as to see what others are saying about a certain book.

1.3 DRAWBACKS OR PROBLEM STATEMENT

Litsy is a social media platform for book lovers. It allows users to connect with other readers, share book recommendations, and keep track of the books they have read, want to read, and are currently reading.

- **User interface**: The app's user interface may not be as clear or straightforward as some users would prefer.

- **Lack of functionality**: Some users might think the app is missing features like the capacity to establish reading challenges or connect with friends on other social networking sites.

- **Technical issues**: There may be technical issues with the app, such as crashes or slow performance, that impact the user experience.

- **Data privacy**: Litsy collects a significant amount of personal information from its users, which has raised privacy concerns among some users.

1.4 PROPOSED PLAN

The Proposed Plan of the system has the following requirements:

- Provide readers with history, and membership in book clubs.

- Improved User Experience: Create an interface with elements that make it simple and straightforward for users to find and interact with new books.

- Provide social networking tools, such as messaging, discussion boards in book clubs, so that users may get to know and communicate with other readers.

- Offer a powerful search and discovery engine for books that enables users to locate books based on a variety of factors, including genre, author, or keyword.

- Cross platform support, including desktop and mobile devices.

- Can be run on IOS and android mobile OS.

CHAPTER 2

FEASIBILITY STUDY

2.1. Technical Feasibility

User hardware requirement

The minimum hardware requirements which are needed in order to use this mobile application are listed below.

Hardware	Specifications
Processor	Clock speed with at least 1.8Ghz
Memory	4 GB RAM or more
Hard disk	3 GB or more

Table 2.1.1.1: User Hardware Requirements

Developer Hardware Requirement

The minimum hardware requirements which are needed in order to build this mobile application are listed below.

Hardware	Specifications
Processor	Clock speed with at least 2Ghz or more
Memory	8 GB RAM or more
Hard disk	12 GB or more

User Software requirement

The minimum software requirements which are needed in order to use this mobile application are listed below.

Hardware	Specifications
IDE/Editor	Android Studio, IntelliJ IDEA, or Visual Studio Code.
Operating system	Linux, macOS (64-bit), Windows 7 or later.
Software	Flutter SDK (version 3.7.1), dart(2.19.1),latest version of flutter packages, firebase sdk.

Table 2.1.2.1: User Software Requirements.

Developer Software Requirement

The minimum hardware requirements which are needed in order to build this mobile application are listed below.

Components	Specifications
IDE/Editor	Android Studio, IntelliJ IDEA, or Visual Studio Code.

Operating system	Linux, macOS (64-bit), Windows 7 or later.
Software	Flutter SDK (version 3.7.1), dart(2.19.1),latest version of flutter packages, firebase sdk.

Table 2.1.2.2: Developer Software Requirements

2.2. Operational Feasibility

It has been established that the book app project is operationally possible after careful analysis and evaluation of several criteria, including resource availability and realistic timescale. The project's timeframe can be completed within the organization's limits since all essential resources, including persons, money, and technology, have been identified and secured.

2.3. Economic Feasibility

Economic feasibility analysis is the process of determining whether developing the system is worth the cost and time investment.

The costs of developing an app would include the initial development costs, any necessary equipment or software, and any ongoing maintenance costs. It would also include the cost of any additional features, such as integration with third-party payment systems, and the cost of any necessary updates or bug fixes.

- The project is academic in nature so the cost incurred will be minimum.

- The software used to produce this app is available for free online.

2.3.1 COCOMO MODEL:

The basic constructive cost model (COCOMO) model is a static, single-valued model that computes software development effort (and cost) as a function of a program size expressed in estimated lines of code (LOC).

The basic COCOMO equations take the form:

Effort Applied (E) = ab (KLOC)^{bb}, [person-months]

Development Time (D) = cb(Effort Applied)^{db} [months]

People Required (P) = Effort Applied / Development Time [Count].

Where KLOC is the estimated number of delivered lines (expressed in thousands) of code. The coefficients ab, bb,cb, and db are given in the following table:

Software Project	ab	bb	cb	db
Organic	2.4	1.05	2.5	0.38
Semi-Detached	3.0	1.12	2.5	0.35
Embedded	3.6	1.20	2.5	0.32

Table 2.3.1.1: Developer Software Requirements

Our project is Organic project, Estimate LOC = 2000

Now the basic COCOMO equation of our project is

Effort Applied (E) = ab (KLOC)^{bb}, [person-months]

$= 2.4(2K)^{1.05}$ [person-months]

$= 4.96$ [person - months]

Development Time (D) $= cb(\text{Effort Applied})^{db}$ [months]

$= 2.5(4.96)^{0.38}$ [person - months]

$= 4.59$ [person - months]

People Required (P) $= \text{Effort Applied} / \text{Development Time}$ [Count].

$= 4.96/4.59$ [count]

$= 1.08$ [count] $= 1$ approximately

The development time for this project is 5 months with 1 person.

As we have a limited time of approximately 5 months to complete this project, this project will be completed within the timeframe.

2.4. Schedule Feasibility

The main part of this project is to build a book forum mobile application with a user-friendly interface with multiple functions which will be used by customers to interact with people they want. The project is set to be completed within the due deadline of the given time frame.

2.4.1. Work Breakdown Structure

A work breakdown structure provides the necessary framework for detailed cost estimating and control along with guiding schedule development and control.

The total project development estimation time (in hours), for our application is **660 hours**.

To further explain our calculation of ours,

The start date of the project is 09/01/2023.

The total number of weeks: 21 weeks and 4 days approximately

Thus,

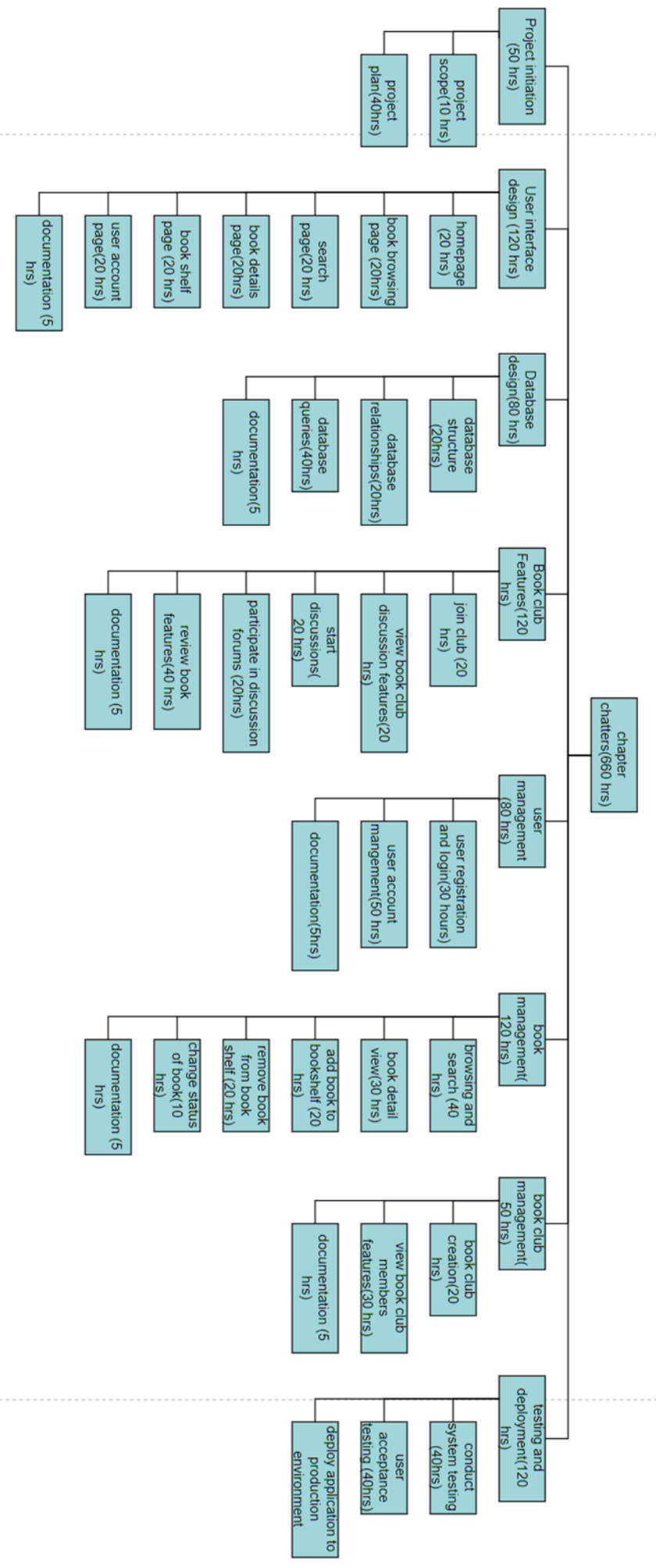the total number of working days is 110 days.

One day = 6 hours of work approximately.

So now, total hours = a total number of days (110) x Number of hours worked per day (6)

= 660 hours.

Therefore, the total number of hours applicable for this project according to the schedule allotted is 660.

The Work breakdown structure diagram shown below further describes the work hour load allotted for each task.

chapter chatters (660 hrs)

- **Project initiation (50 hrs)**
  - project scope (10 hrs)
  - project plan (40hrs)

- **User interface design (120 hrs)**
  - homepage (20 hrs)
  - book browsing page (20hrs)
  - search page (20hrs)
  - book details page (20hrs)
  - book shelf page (20 hrs)
  - user account page (20 hrs)
  - documentation (5 hrs)

- **Database design (80 hrs)**
  - database structure (20hrs)
  - database relationships (20hrs)
  - database queries (40hrs)
  - documentation (5 hrs)

- **Book club Features (120 hrs)**
  - join club (20 hrs)
  - view book club discussion features (20 hrs)
  - start discussions (20 hrs)
  - participate in discussion forums (20hrs)
  - review book features (40 hrs)
  - documentation (5 hrs)

- **user management (80 hrs)**
  - user registration and login (30 hours)
  - user account mangement (50 hrs)
  - documentation (5hrs)

- **book management (120 hrs)**
  - browsing and search (40 hrs)
  - book detail view (30 hrs)
  - add book to bookshelf (20 hrs)
  - remove book from book shelf (20 hrs)
  - change status of book (10 hrs)
  - documentation (5 hrs)

- **book club management (50 hrs)**
  - book club creation (20 hrs)
  - view book club members features (30 hrs)
  - documentation (5 hrs)

- **testing and deployment (120 hrs)**
  - conduct system testing (40hrs)
  - user acceptance testing (40 hrs)
  - deploy application to production environment

Figure 2.4.1 Work Breakdown Structure

2.4.2. Gantt Chart

ID	Name	Jan, 23	Feb, 23	Mar, 23	Apr, 23	May, 23	Jun, 23	Jul, 23

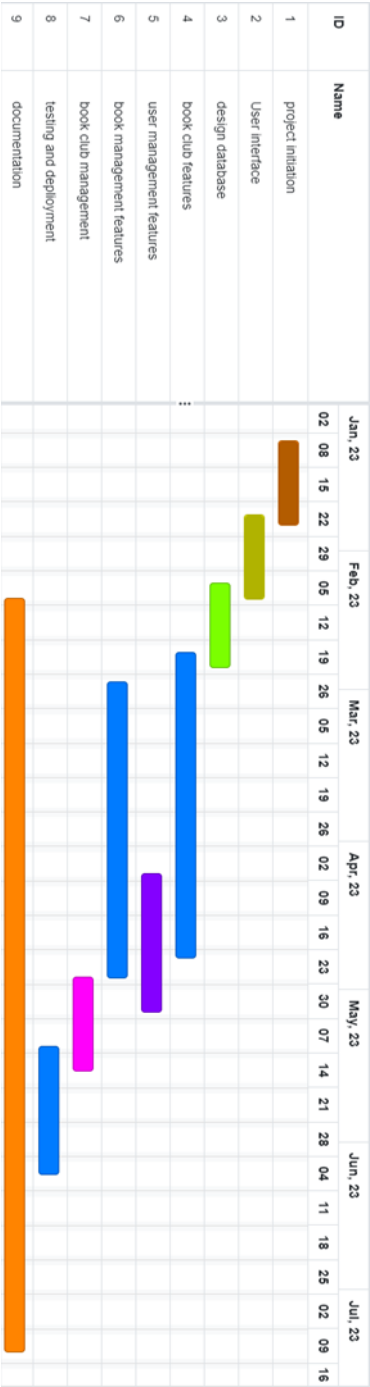| ID	Name
1	project initiation
2	User interface
3	design database
4	book club features
5	user management features
6	book management features
7	book club management
8	testing and deployment
9	documentation

Figure 2.4.2 Gantt Chart

CHAPTER 3

DESIGN DIAGRAMS

3. Design Diagram

Design Diagrams help visualize software programs using various diagrams. The diagram illustrates the relation, data flow and different modules or features of the software.

3.1 Use case diagram

A use case diagram is a graphical depiction of a user's possible interactions with a system. A use case diagram shows various use cases and different types of users the system has and will often be accompanied by other types of diagrams as well. The use cases are represented by either circles or ellipses. The actors are often shown as stick figures.
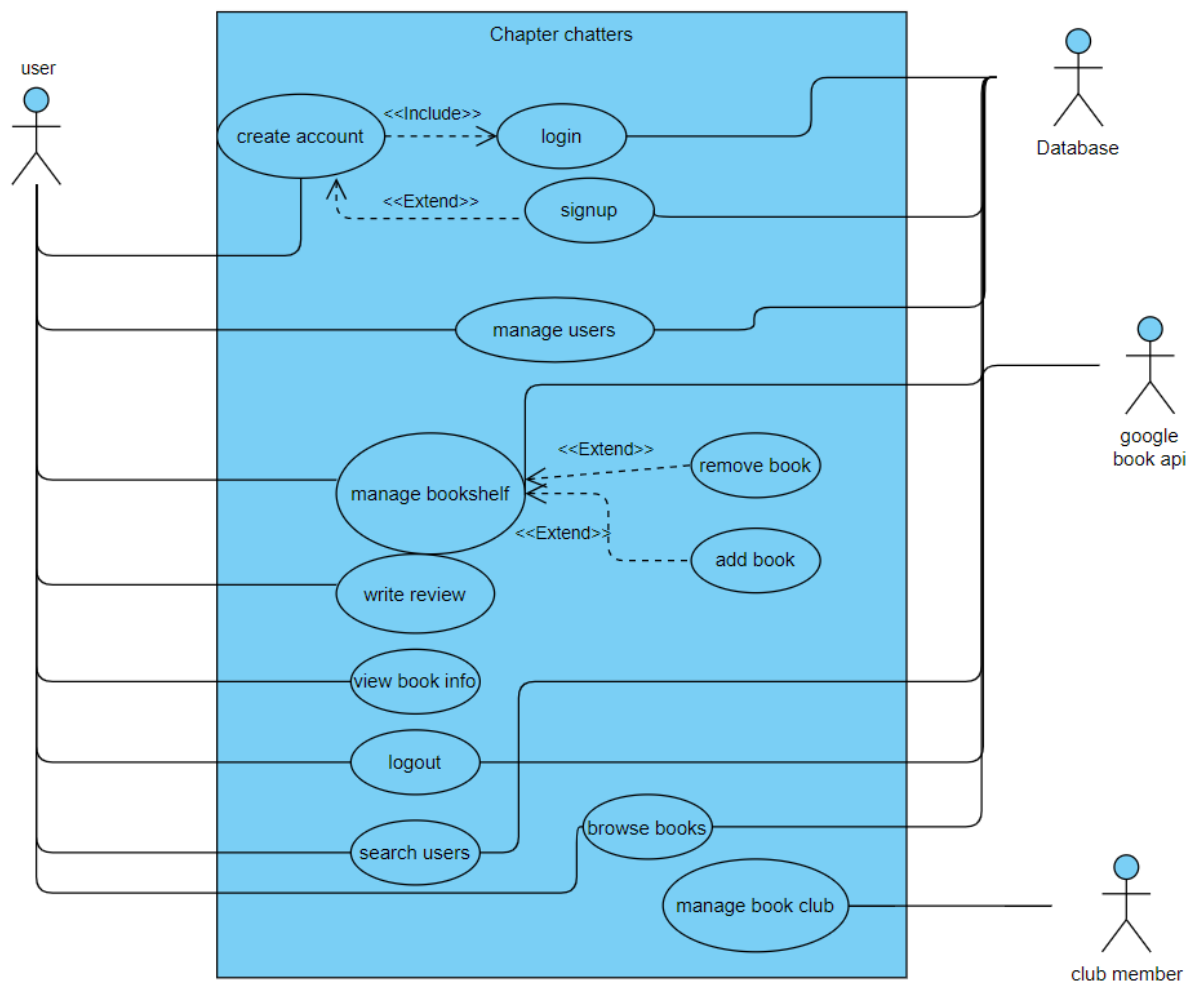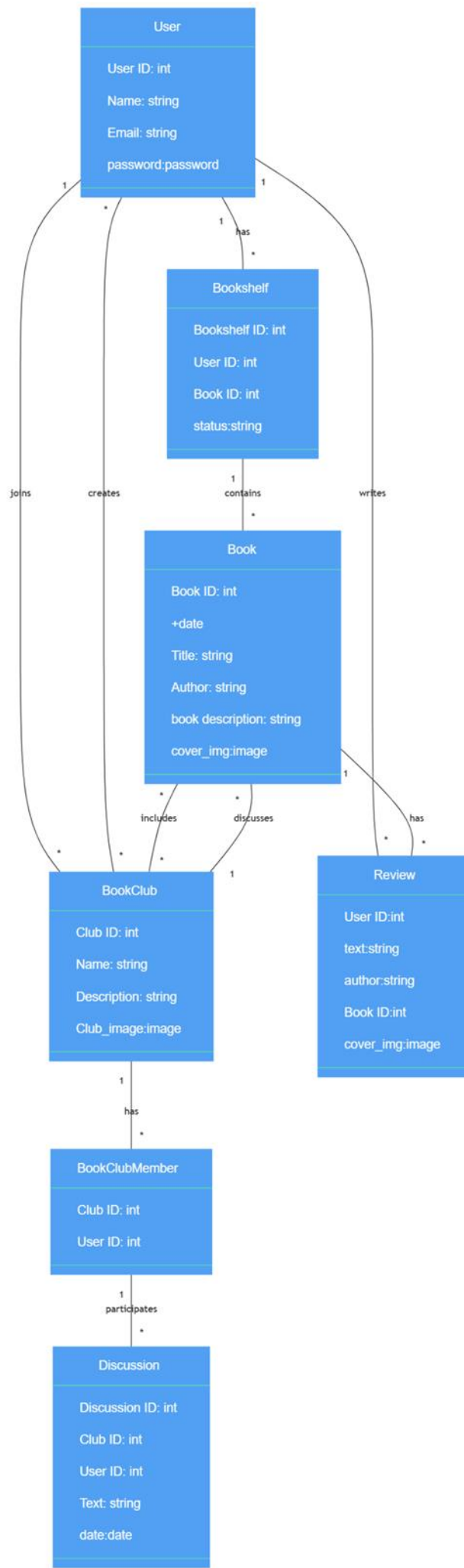
Figure 3.1 Use case Diagram

3.2. Class diagram

The class diagrams are use to describe structural components, representing the components Activity and Service, classes, its attributes, and methods.

User

User ID: int

Name: string

Email: string

password:password

Bookshelf

Bookshelf ID: int

User ID: int

Book ID: int

status:string

has

contains

Book

Book ID: int

+date

Title: string

Author: string

book description: string

cover_img:image

joins

creates

writes

includes

discusses

has

BookClub

Club ID: int

Name: string

Description: string

Club_image:image

Review

User ID:int

text:string

author:string

Book ID:int

cover_img:image

has

BookClubMember

Club ID: int

User ID: int

participates

Discussion

Discussion ID: int

Club ID: int

User ID: int

Text: string

date:date

14

3.3. Activity diagram

Activity Diagrams is used to illustrate the flow of control in a system and refer to the steps involved in the execution of a use case. We model sequential and concurrent activities using activity diagrams. So, we basically depict workflows visually using an activity diagram. An activity diagram focuses on condition of flow and the sequence in which it happens. We describe or depict what causes a particular event using an activity.
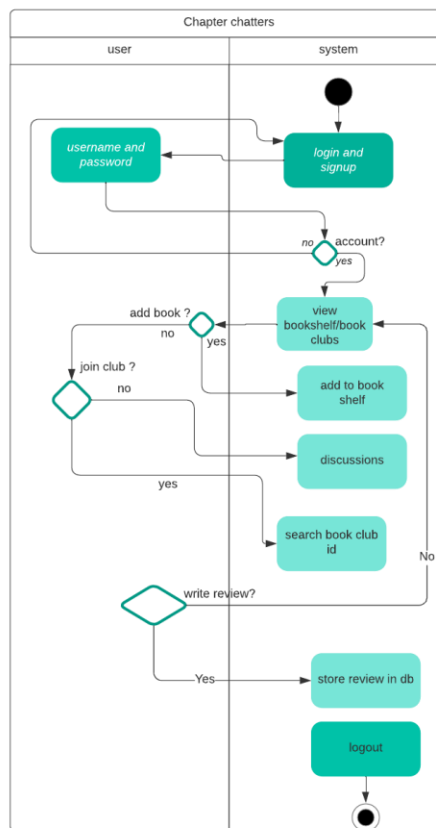
Figure 3.3 Activity Diagram

CONCLUSION

The book application project has the potential to give book lovers a forum for interaction, sharing of reading experiences, and participation in book clubs. It can provide an effective, user-friendly, and scalable solution to manage bookshelves, books, reviews, discussions, and book clubs with the integration of flutter and firebase. Cardinality relationships, activity diagrams, and decision nodes can all be used to organize and streamline the system's operations. This project has the potential to provide book lovers with an original and comprehensive solution, and it could be further improved with new functions and features.

REFERENCES

[1] Feasibility study website: https://www.geeksforgeeks.org/types-of-feasibility-study-in-software-project-development/ accessed on 29/01/2023

[2] Gantt chart website: https://www.investopedia.com/terms/g/gantt-chart.asp accessed on 29/01/2023

[3] COCOMO model website: https://www.geeksforgeeks.org/software-engineering-cocomo-model/ accessed on 29/01/2023

[4] Use case video: https://www.youtube.com/watch?v=zid-MVo7M-E accessed on 5/02/2023

[5] Activity diagram video: https://www.youtube.com/watch?v=_tCedK0CfMk accessed on 5/02/2023

[6] Class diagram video: https://www.youtube.com/watch?v=UI6lqHOVHic&t=7s accessed on 5/02/2023

[7] Creating the diagrams: https://www.lucidchart.com/pages/ accessed on 29/01/2023

[8] study on book clubs: https://time.com/5809322/social-distancing-book-clubs/ accessed on 29/01/2023

[9] Existing system: https://www.lucidchart.com/pages/ accessed on 29/01/2023