

ATMIYA UNIVERSITY
RAJKOT



A
Report On
CodeBook
Under subject of
MAJOR PROJECT
B. TECH Semester–VII
(Computer Engineering)

Submitted by:

Jay Mahendrabhai Malli (190002066)

Jay Kantilal Nakrani (190002070)

Prof. Nirali Borad

(Faculty Guide)

Prof. Tosal M. Bhalodia

(Head of the Department)

Academic Year

(2022-23)

CANDIDATE'S DECLARATION

We hereby declare that the work presented in this project entitled “**CodeBook**” submitted towards completion of project in **7th Semester** of B.Tech. (Computer Engineering) is an authentic record of our original work carried out under the guidance of “**Prof. Nirali Borad**”.

We have not submitted the matter embodied in this project for the award of any other degree.

Semester: 7th

Place: Rajkot

Signature:

Jay Malli (190002066)

Jay Nakrani (190002070)

ATMIYAUNIVERSITY
RAJKOT



CERTIFICATE

Date:

This is to certify that the “**CodeBook**” has been carried out by **Jay Mahendrabhai Malli** under my guidance in fulfillment of the subject Project in COMPUTER ENGINEERING (7th Semester) of Atmiya University, Rajkot during the academic year 2022-23.

Prof. NIRALI BORAD
(**Project Guide**)

Prof. Tosal M. Bhalodia
(**Head of the Department**)

ATMIYAUNIVERSITY

RAJKOT



CERTIFICATE

Date:

This is to certify that the “**CodeBook**” has been carried out by **Jay Kantilal Nakrani** under my guidance in fulfillment of the subject Project in COMPUTER ENGINEERING (7th Semester) of Atmiya University, Rajkot during the academic year 2022-23.

Prof. NIRALI BORAD
(**Project Guide**)

Prof. Tosal M. Bhalodia
(**Head of the Department**)

Sr. No.	TITLES	Page No.
	Index	I
	List of Figures	1
	List of Tables	1
	Acknowledgement	2
	Abstract	3
1.	Introduction	4
	1.1 Purpose	4
	1.2 Scope	4
	1.3 Technology and Tools	4
2.	Project Management	7
	2.1 Project Planning	7
	2.2 Project Scheduling	7
	2.3 Risk Management	7
	2.3.1 Risk Identification	7
	2.3.2 Risk Analysis	8
3.	System Requirements Study	10
	3.1 Hardware and Software Requirements	10
	3.1.1 Server-side hardware requirement	10
	3.1.2 Software requirement	10
	3.1.3 Client-side hardware requirement	10
	3.2 Constraints	11
	3.2.1 Hardware Limitation	11
	3.2.2 Reliability requirements	11
4.	System Analysis	12
	4.1 Study of Current System	12
	4.2 Problem and Weaknesses of Current System	12
	4.3 Feasibility Study	13
	4.3.1 Does the system contribute to overall objectives of the organization?	14
	4.3.2 Can the system be implemented using the current technology and within the given cost and schedule constraints	14
	4.3.3 Can the system be integrated with other system which are already in place?	14
	4.4 Technical Feasibility	14

4.5	Selection Of Hardware and Software and Justification	14
5	System Design	15
5.1	Input /output interface	15
5.2	Interface Design	17
5.2.1	Class Diagram	17
5.2.2	Use Case Diagram	18
5.2.3	Activity Diagram	19
5.2.4	Data Flow Diagram	20
6	Code Implementation	21
6.1	Implementation Environment	21
6.2	Program/Module Specification	21
6.3	Coding Standards	21
7	Testing	22
7.1	Testing Strategy	22
7.2	Testing Method	22
7.2.1	Unit Testing	22
7.2.2	Integration Testing	22
7.2.3	Validation testing	22
8	Limitations and Future Enhancement	23
8.1	Limitations	23
8.2	Future Enhancement	23
9	Conclusion	24
10	References	25

LIST OF Figures

Figure No.	Figure Title	Page No.
5.1.1	Home page	15
5.1.2	Login Page	16
5.1.3	Signup Page	16
5.2.1	Class Diagram	17
5.2.2	Use Case Diagram	18
5.2.3	Activity Diagram	19
5.2.4.1	Context Level DFD	20
5.2.4.2	First Level DFD	20

LIST OF Tables

Table No.	Table Title	Page No.
3.1.1.1	Hardware Requirements	10
3.1.2.1	Software Requirements	10
3.1.3.1	Client-Side Requirements	10

ACKNOWLEDGEMENT

We have taken many efforts in this project. However, it would not have been possible without the kind support and help of many individuals and organizations. We would like to extend our sincere thanks to all of them.

We are highly indebted to **Prof. Nirali Borad** for their guidance and constant supervision as well as for providing necessary information major regarding the Project titled “**CodeBook**”. We would like to express our gratitude towards staff members of Computer Engineering Department, Atmiya University for their kind co-operation and encouragement which helped us in completion of this project.

ABSTRACT

The CodeBook provides virtual notebook, so user can prepare notes of his learning. As well as user can access his notes anywhere anytime. User can add images texts and code in his notes. Also, it's provided features like highlighting text, quick links, code formatting. Also, user can share notes with others and he can also provide rating to other's notes

1. INTRODUCTION

1.1. Purpose

- The purpose of this project is user prepare notes of his daily learning in a good way and he can access or share his notes anywhere , anytime.

1.2. Scope

- We cannot carry our hard copy with along us whole the time. So, it's very coinvent to use CodeBook for preparing and maintaining notes with the help of CodeBook.

1.3. Technology and tools

We hope to use following frameworks and techniques through the implementation of our proposed work.

- Frontend

- Html
- Javascript
- ReactJs
- Chakra UI
- React Animation
- Progressive Web App

- Backend

- NodeJs
- Express

- Firebase

1. ReactJs:

The user interaction with our website is considerable therefore; we should guarantee that it ensures all aspects that it should be enriched with. Therefore, we thought to adapt ReactJs for developing our client-side application that provides a single page application framework by JSX is an extension to the JavaScript language syntax, similar in appearance to HTML.

2. Chakra UI:

Chakra UI is a simple, modular and accessible component library that gives you the building blocks you need to build your React applications.

3. Firebase:

Firebase is a no-SQL database which is supported by Google and here it's used as a database for storing data.

- Firestore: it's used for store data about users.
- Authentication: it's used for check whether user is authenticated or not.
- Hosting: it's used for deploy a website.

4. Html:

HTML, in full hypertext mark-up language, a formatting system for displaying material retrieved over the Internet. Each retrieval unit is known as a Web page (from World WideWeb), and such pages frequently contain hypertext links that allow related pages to be retrieved. HTML is the mark-up language for encoding Web pages.

5. JavaScript:

JavaScript is a dynamic computer programming language. It is lightweight and most commonly used as a part of web pages, whose implementations allow client-side script to interact with the user and make dynamic pages. It is an interpreted programming language with object-oriented capabilities.

6. React Animation:

A collection of animations that can be used with any inline style library that supports using objects to define keyframe animations

7. Progressive Web App:

Progressive Web Apps (PWAs) are web apps that use service workers, manifests, and other web-platform features in combination with progressive enhancement to give users an experience on par with native apps.

8. NodeJS:

NodeJS (Node) is an open-source development platform for executing JavaScript code server-side. Node is useful for developing applications that require a persistent connection from the browser to the server and is often used for real-time applications such as chat, news feeds and web push notifications.

9. Firebase:

Firebase is a set of hosting services for any type of application. It offers NoSQL and real-time hosting of databases, content, social authentication, and notifications, or services, such as a real-time communication server.

1. PROJECT MANAGEMENT

2.1 Project Planning:

Project Planning is concerned with identifying and measuring the activities, milestones and deliverables produced by the project. Project planning consists of following essential activities:

- Scheduling manpower and other resources needed to develop the system like API needed for fetching data.
- Staff organization and staffing plans.
- Risk identification, analysis, and accurate planning.
- Estimating some of the basic attributes of the project like cost, duration and efforts.

2.2 Project Scheduling:

The scheduling is the peak of a planning activity, a primary component of software project management. When combined with estimation methods and risk analysis, scheduling establishes a roadmap for project management.

2.3 Risk Management:

Risk management consists of a series of steps that help a software development team to understand and manage uncertain problems that may arise during the course of software development and can plague a software project.

2.3.1 Risk Identification:

Risk identification is a first systematic attempt to specify risks to project plan, scheduling resources, project development. It may be carried out as a team process using brainstorming approach.

Technology risk:

- Technical risks concern implementation, potential design, Interfacing, testing, and maintenance problems.
- API Corruptness
- Issue with database

People Risks:

- These risks are concerns with the team and its members who are taking part in developing the system.
- Leaking an important data
 - Failure of the administration
 - Lack of knowledge.
- Lack of clear product vision.

Tools Risks:

- These are more concerned with tools used to develop the system.
- Tools containing virus.

General Risks:

- General Risks are the risks, which are concerned with the mentality and resources.
- Lack of resources can cause great harm to efficiency and timely productivity.
- Rapidly changing requirements.

2.3.2 Risk Analysis:

“Risk analysis = risk assessment + risk management + risk communication.” Risk analysis is employed in its broadest sense to include:

Risk assessment:

- Involves identifying sources of potential harm, assessing the likelihood that harm will occur and the consequences if harm does occur.
- For this project It might be - Some issue with fetching of weather data from API.

Risk management:

- Evaluates which risks identified in the risk assessment process require management and selects and implements the plans or actions that are required to ensure that those risks are controlled.
- Precautions taken to make risks minimal are as under.
 - Check user authentication at some interval after the user is logged in.
 - Throw an exception if any issue is generated from the server side during fetch data.

Risk communication:

- Involves an interactive dialogue between stakeholders and risk assessors and risk managers which actively informs the other processes.
- Probability of certain risks is negotiated with client

3. SYSTEM REQUIREMENTS STUDY

3.1 Hardware and Software Requirement

- This shows minimum requirements to carry on running this system efficiently.

3.1.1 Hardware Requirements

Table 3.1.1.1 Hardware Requirement

Devices	Description
Processor	Intel Core Duo 2.0 GHz or more
RAM	2 GB or more
Hard Disk	256 GB or more

3.1.2 Software Requirements

Table 3.1.2.1 Software Requirements

For which	Software
Operating System	Windows 7/8/10, android [for mobile]
Front End	HTML, Chakra Ui, ReactJs
Back End	Firebase Database
Coding Language	JavaScript

3.1.3 Client-side Requirements

Table 3.1.3.1 Client-side Requirements

For which	Requirements
Browser	Any Web Browser [chrome, Microsoft Edge]

3.2 Constraints

3.2.1 Hardware Limitations

The major hardware limitations faced by the system are as follows:

- If the appropriate hardware is not there like processor, RAM, hard disks
- The problem in processing requests of client
- If appropriate storage is not there our whole database will crash due to less storage because our main requirement is large storage.

3.2.2 Reliability Requirements

Since many users can access the server simultaneously, load on the server becomes very high. Hence, the server should be of enough high configurations. There should be high back up storage and management of huge data for overall ideas, videos, images, multiple countries, and multiple user profile. The Reliability requirements are the validations used to protect the system against one or more incorrect activities. Without proper validation of the system, the failure possibilities of it grow higher so it is must to understand the proper validation of the system and must implement them. All the required validator controls spend very good role to keep the system secure from any unauthorized or incorrect information. In all these validation actions if system found one or more entries violating validation rules, then user will be warned by proper error messages and the details or the record is not going to be saved until corrections are made to them.

4. SYSTEM ANALYSIS

4.1 Study Current System

Previously built notebook project web based application was compatible with system and every time users start this application, they have to maintain their notes like pdf. Due to complex coding, system responding time was high and requires more memory to get startup.

4.1 Problem and weakness of current system

- Requires more memory for app.
- page is not attractive
- Time consuming

4.2 Requirements of New System

4.2.1 User Requirements:

The user requirement for this system is mobile phone or computer with flexible RAM, so system can run on it very efficiently.

4.2.2 System Requirements:

- **Functional System Requirement:**

This section gives a functional requirement that applicable to the Weather WebApp.

There are two sub modules in this phase.

1. User module.
2. Admin module.

1. **User module:** A user must login with his user name and password to the system after registration. If they are invalid, the user not allowed entering the system.

- Username and password will be provided after user registration is confirmed.
- A new user will have to register in the system by providing essential details in order to view weather details.
- The system must encrypt the password of the customer to provide security.

- The user can access notes.

2. **Admin module:** The administrator can add user, delete user, view user and block user.

- The administrator can update the user interface.
- The administrator can view user's details like email, username etc.

• **Non-Functional System Requirements:**

i. RELIABILITY REQUIREMENT:

The system should provide a reliable environment to the user

ii. USABILITY REQUIREMENT:

The website is designed for user friendly environment and ease of use.

iii. IMPLEMENTATION REQUIREMENT:

Implementation of the system using Chakra UI and ReactJs in front end with firebase as back end and it will be used for database connectivity. Responsive web designing is used for making the website compatible for any type of screen. As well this website also contains the PWA feature.

iv. DELIVERY REQUIREMENT:

The whole system is expected to be delivered in four months of time with a weekly evaluation by the project guide.

4.3 Feasibility Study

The feasibility study of any system is mainly intended to study and analyze the proposed system and to decide whether the system under consideration will be viable or not after implementation. That is, it determines the usability of the project after deployment. To come to result a set of queries is answered keeping the efficiency of the software and its impact on the domain for which it was developed.

4.4 Technical Feasibility:

In technical feasibility, we study all technical issues regarding the proposed system. It is mainly concerned with the specifications of the equipment and the software, which successfully satisfies the end-user's requirement. The technical needs of the system may vary accordingly but include:

- The feasibility to fetch the weather data in a minimal time.
 - Response time under certain conditions.
 - Ability to change the notes layout dynamically according to the current request.

4.5 Selection of Hardware and Software and Justification

- The configuration of the existing systems is:
- Processor : i3 500 MHz (or above)
- Memory : 128 MB (or above)
- For Software there are following alternatives:
- Operating System : Window 7/8/9/10, Linux, android [for mobile]
- Development tools : JavaScript, CSS, HTML
- Database : Firebase
- Documentation tool: MS-Word.

5. System Design

5.1 Input/output interface

- Home page

Image Optimization

Images have always been a key part of the web. Our brains are able to interpret images much faster than text, which is why high-quality visuals drive conversions and user engagement.

That said, loading images efficiently at scale isn't a little project for a quiet afternoon. It requires understanding of compression techniques, loading behavior, image decoding and image CDNs, adaptive media loading and caching.



Figure 5.1.1 Home page

- Login page

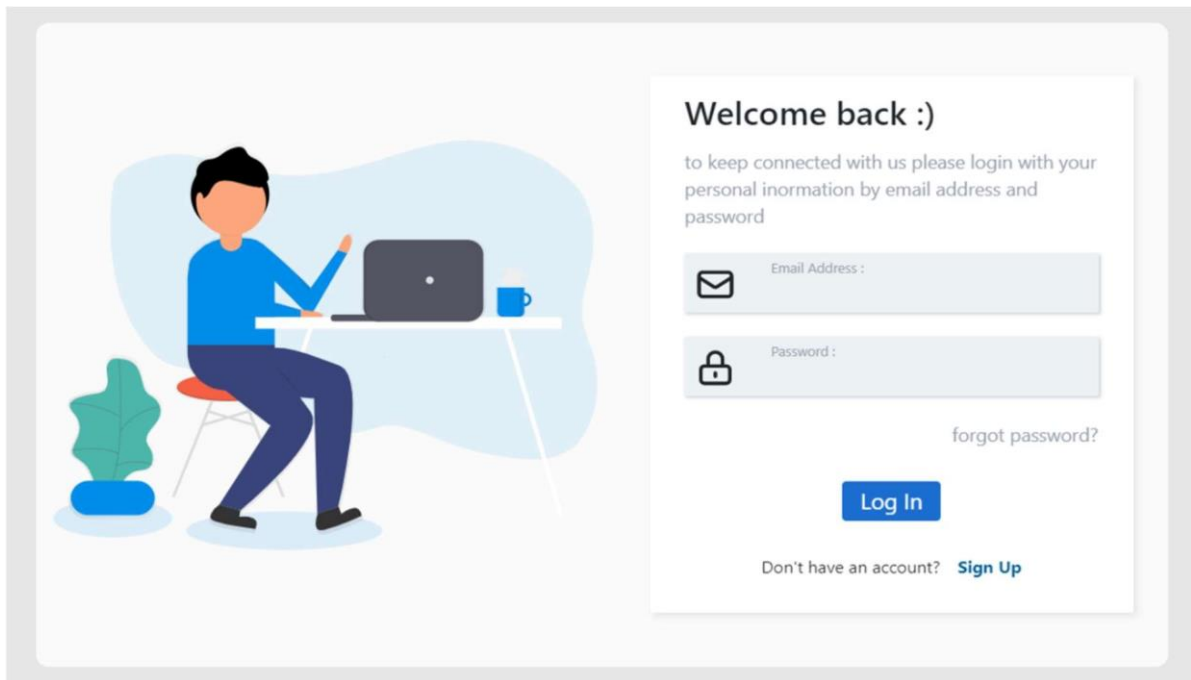


Figure 5.1.2 Login page

- Sign-up page

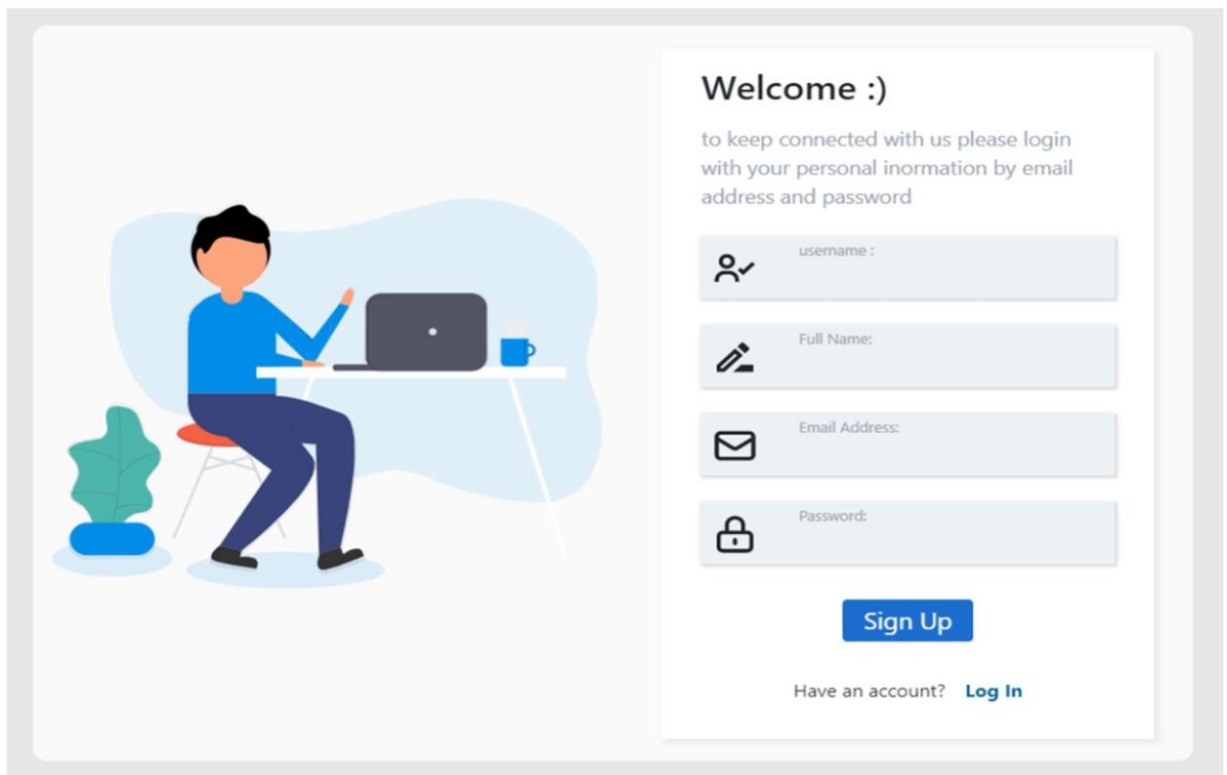


Figure 5.1.3 Sign-up page

5.2 Interface Design

5.2.1 Class Diagram

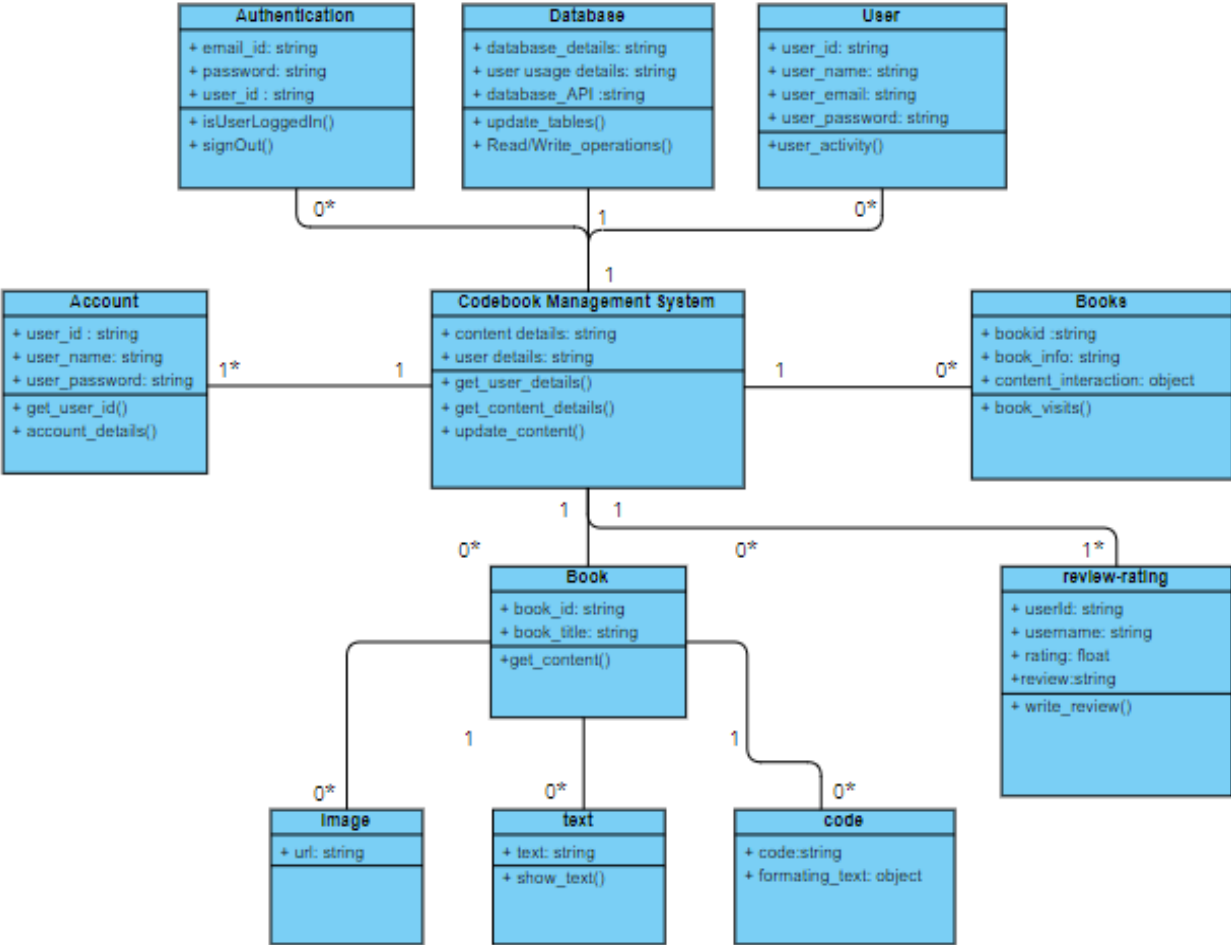


Figure 5.2.1 Class Diagram

5.2.2 Use case Diagram

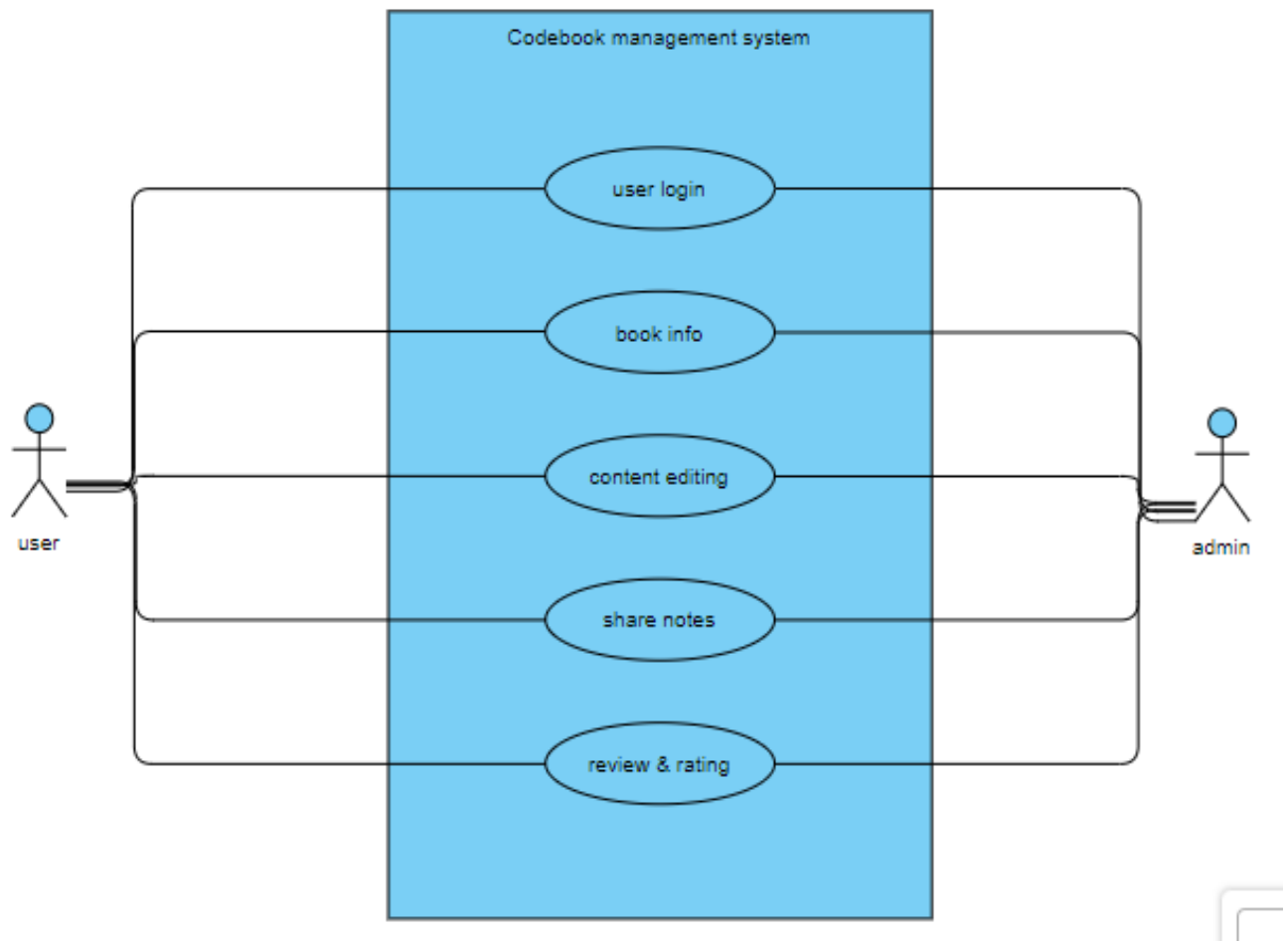


Figure 5.2.2 Use Case Diagram

5.2.3 Activity Diagram

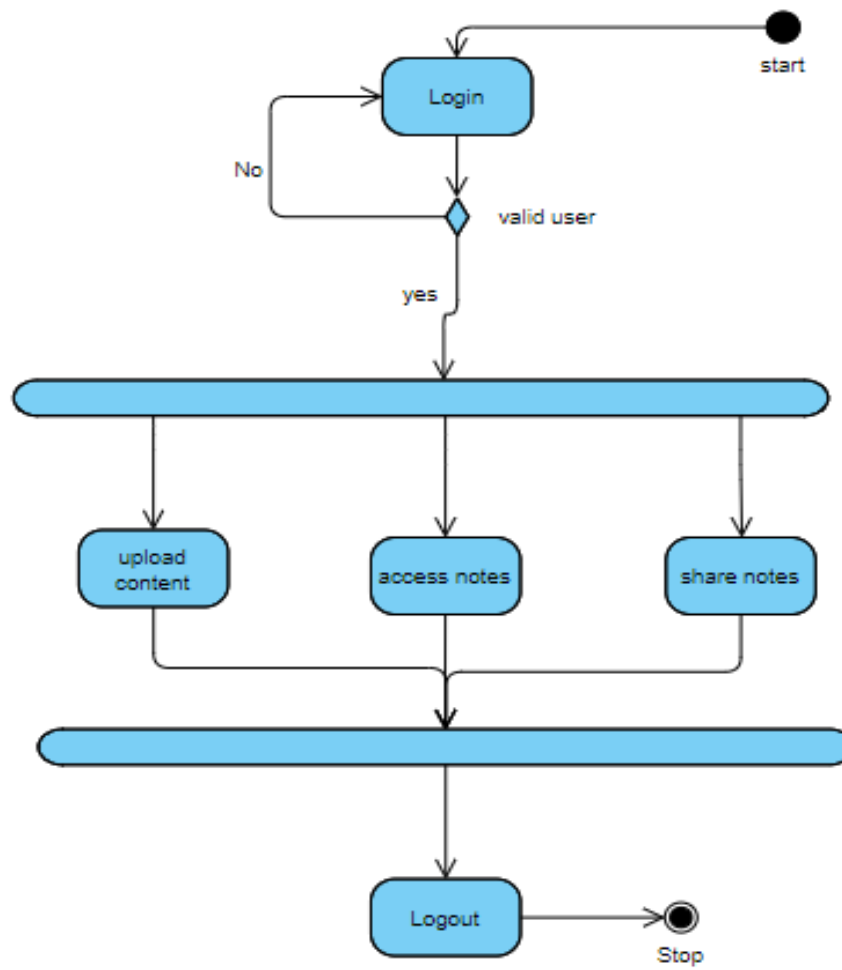


Figure 5.2.3 Activity Diagram

5.2.4 Data Flow Diagram

5.2.4.1 Context Level DFD

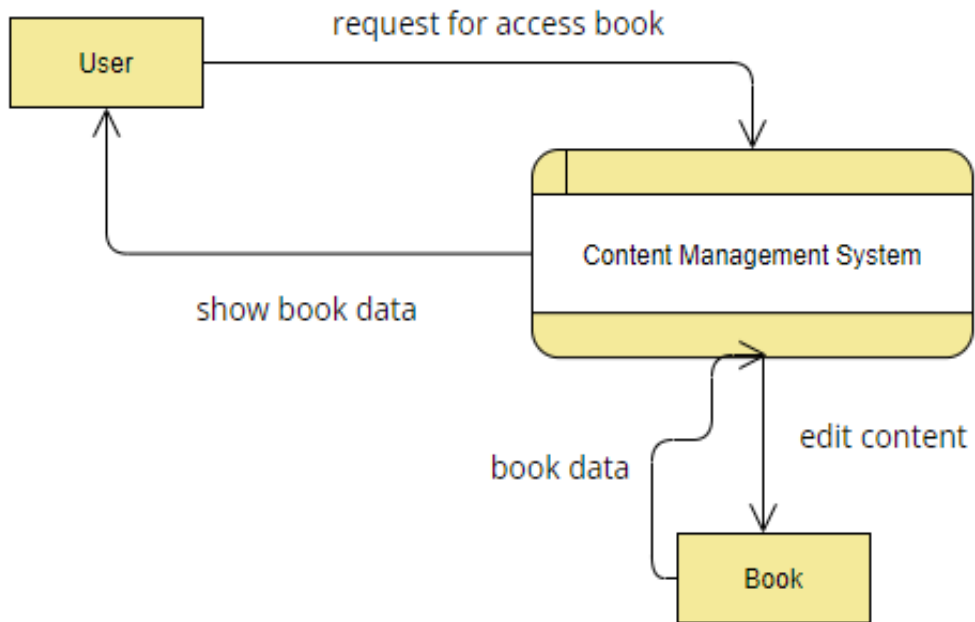


Figure 5.2.4.1 Context level DFD

5.2.4.2 First Level DFD

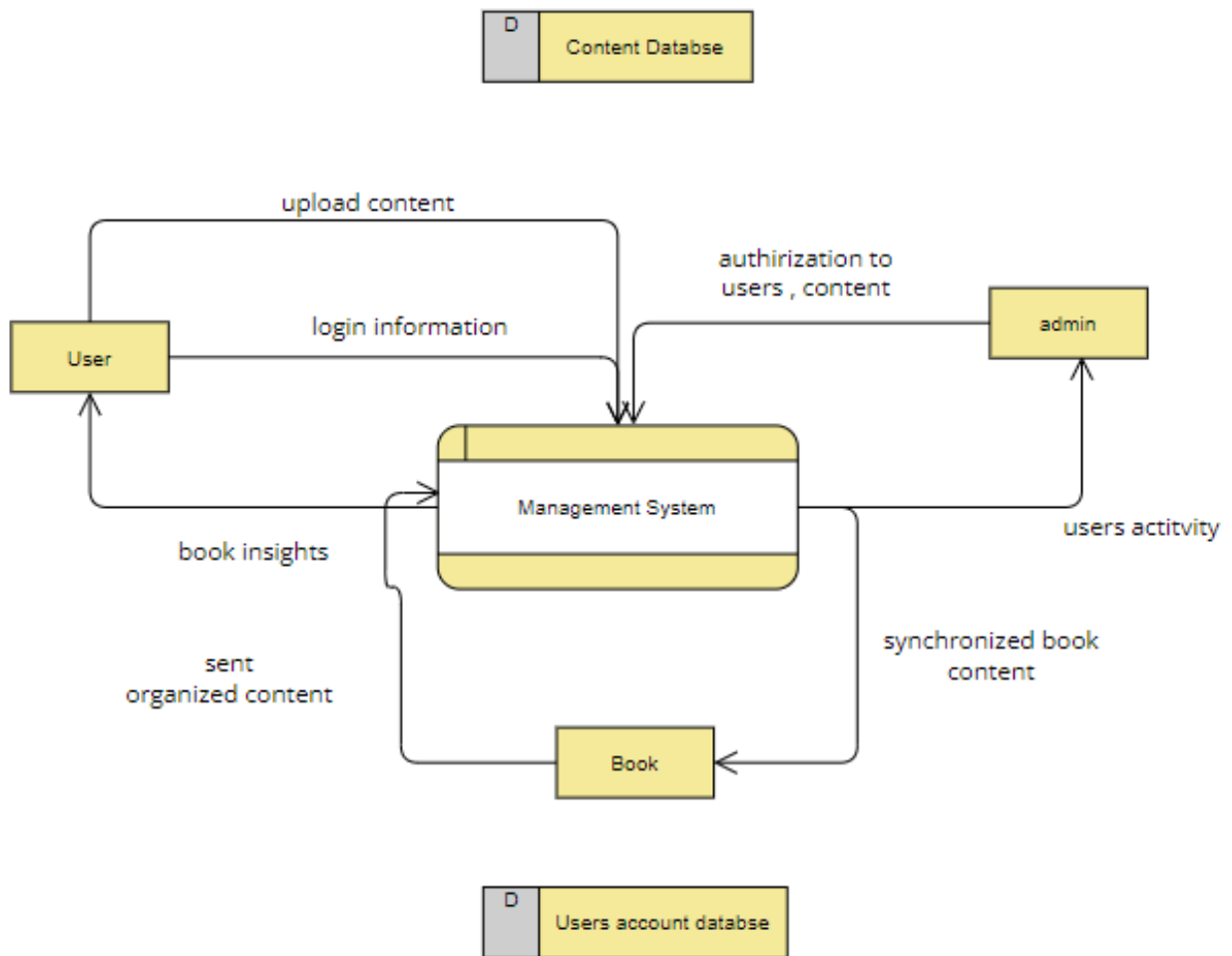


Figure 5.2.4.2 First level DFD

6. Code Implementation

6.1 Implementation Environment

Challenges identified for successful design and implementation of this project are dominated by: Complexity, reliability/availability, weather data access.

6.2 Program/Module Specification

- After discussion with our team, we decided to develop the one main module which areas following:

1. CodeBook User Interface

- For developing this interface, we decided to use ReactJs, Chakra UI and Visual Studio codeeditor for developing code.
- Our project needs to have a considerable size so we decide to use NoSQL database which is Firebase.
- Proper validation is placed as and when it is required.

6.3 Coding Standards

The standards we have followed are:

- Comments are used for documentation of what is doing that piece of code and how done.
- Comment may also be used in the body of the Cascading style sheets to explain individual sections or lines of codes to easily get access and easily review or manage the classes or properties for the pages.
- Inline comments should be made with the // and for multiple line comments we write between /* */.

7. Testing

7.1 Testing Strategy

A strategy for software testing integrates software test case design method into a well-planned series of steps that result in the successful construction of the software. The strategy provides the roadmap that describes the steps to be conducted as a part of testing, then these steps are planned and then undertaken, and how much effort, time and resource will be required.

7.2 Testing Method

7.2.1 Unit Testing

- The unit testing is meant for testing smallest unit of software.
- So, we use the bottom-up approach testing method for this project.
- In this testing method we test the code of individual modules by logically.

7.2.2 Validation Testing

- Validation can be defined as when software functions in a manner that can be reasonably accepted by the user.
- In the validation testing we check the validation for user email, password as well as format of input given by the user.

7.2.3 Integration Testing

- The integration testing is meant to test all the modules simultaneously because it is possible that all the modules may function correctly when tested individually. But they may not work altogether and may lead to unexpected outcome.
- So in the integration testing we test the whole project by register the user and then check the weather details of different locations.
- We doing this testing by create multiple users.

8. Limitations and Future Enhancement

8.1 LIMITATIONS:

Though we tried our best in developing this system but as limitations are mere parts of any System so are of our system. Some limitations of CodeBook are as under:

- Limited storage for notes.
- Not all notes are available offline.

8.2 FUTURE ENHANCEMENT:

There is always a scope for enhancements in any developed system, especially when our nature of the project is iterative waterfall which allows us to rethink on the method of development to adopt changes in the project. Below mentioned are some of the changes possible in the future to increase the adaptability, and efficiency of the system.

- Custom styling, settings for books structure
- More user-friendly user interface.

7. Conclusion

The CodeBook WebApp is a web application that can be downloaded [memory required less than 1 MB] and used in android device as well as in computers to access and prepare notes. CodeBook is a user friendly which is very easy and convenient to use. The system is complete in the sense that it is operational and it is tested by creating book and getting the book content in proper format. But there is always a scope for improvement and enhancement.

8. References

- 1) <https://reactjs.org/> ReactJs framework
- 2) <https://firebase.google.com/> Firebase noSQL database
- 3) <https://chakra-ui.com/> For Chakra UI Documentation
- 4) <https://web.dev/progressive-web-apps/> For PWA Documentation