



**COMSATS University Islamabad,
Abbottabad Campus**

**SOFTWARE DESIGN
DESCRIPTION
(SDD DOCUMENT)**

for

**< E Book Reader and Downloader >
Version 1.0**

By

Sayed Shah Mohsin

FA2-BSE-035

Baryal Rehman

FA20-BSE-038

Supervisor

Mam Neeli Khan

Bachelor of Science in Computer Science (2020-2024)

Table of Contents

Revision History	3
1. Introduction.....	5
2. Design Methodology and software process model	5
3. System Overview	5
3.1 Architectural Design.....	6
3.2 Process Flow/Representation.....	6
4. Design Models [along with descriptions]	7
5. Data Design.....	9
5.1 Data Dictionary	13
6. Algorithm & Implementation	15
7. Software Requirements Traceability Matrix	15
8. Human Interface Design.....	16
8.1 Screen Images.....	16
8.2 Screen Objects and Actions	Error! Bookmark not defined.
9. Appendix I	Error! Bookmark not defined.

Revision History

Name	Date	Reason for changes	Version

Application Evaluation History

Comments (by committee) *include the ones given at scope time both in doc and presentation	Action Taken

Supervised by
<Neeli Khan>

Signature_____

Introduction

E-Book Reader and Downloader application is designed to offer an exceptional and user-friendly platform for book enthusiasts, using the power of React Native programming language with state-of-the-art tools and techniques. Our primary aim is to create a feature-rich ecosystem for reading and downloading books seamlessly. We will outline the key modules and features that E-Book Central will offer, including user registration and login, a comprehensive book catalog, an immersive book reading experience with EPub Viewer integration, efficient state management, and secure storage for favorites and downloaded content with Firebase-Book Central is driven by the motivation to cater to the practical needs of book lovers, ensuring a convenient and enjoyable reading experience. By using advanced technologies and integrating essential features, we are confident that this app will not only meet but exceed the expectations of book readers, providing them with a versatile and delightful platform for their literary pursuits.

Design methodology and software process model

Design Methodology: Functional Programming

The choice of using a functional programming approach for the design of an E-book Reader and Downloader mobile application is influenced by the characteristics of functional programming that align with the requirements of the project.

Functional programming is a programming paradigm that treats computation as the evaluation of mathematical functions and avoids changing-state and mutable data. Here's how functional programming principles can be applied to the design of the E-book Reader and Downloader:

Software Process Model: Agile

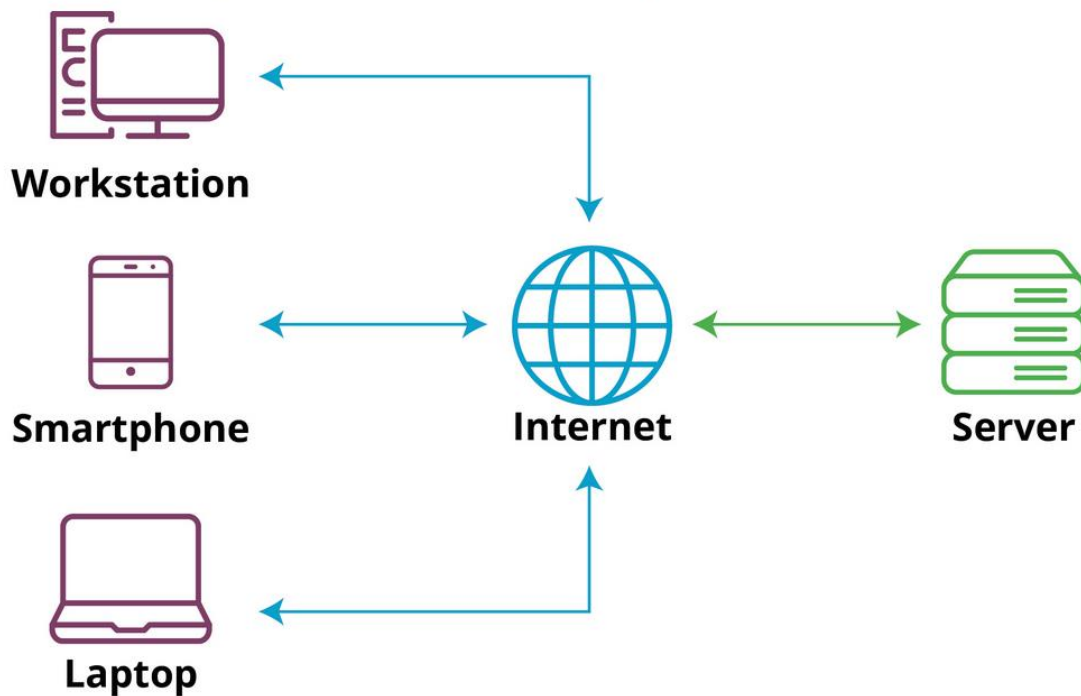
the development of the E-book Reader and Downloader mobile application, the Agile software process model is chosen to prioritize customer collaboration and adaptability. Agile's iterative and incremental approach allows for the rapid delivery of valuable increments, ensuring that user feedback can be promptly incorporated into the evolving product. The Agile methodology's emphasis on continuous testing, quick adaptation to changing requirements, and incremental development aligns well with the dynamic nature of mobile app development for an E-book reader and downloader.

System overview

The E-book Reader and Downloader mobile application is designed to provide users with a seamless and feature-rich platform for discovering, downloading, and reading digital books. The core functionalities include a user-friendly interface for browsing a diverse catalog of e-books, a robust downloading system for offline access, and an immersive e-reading experience. The app supports various file formats, allowing users to enjoy their favourite books in a flexible and personalized manner.

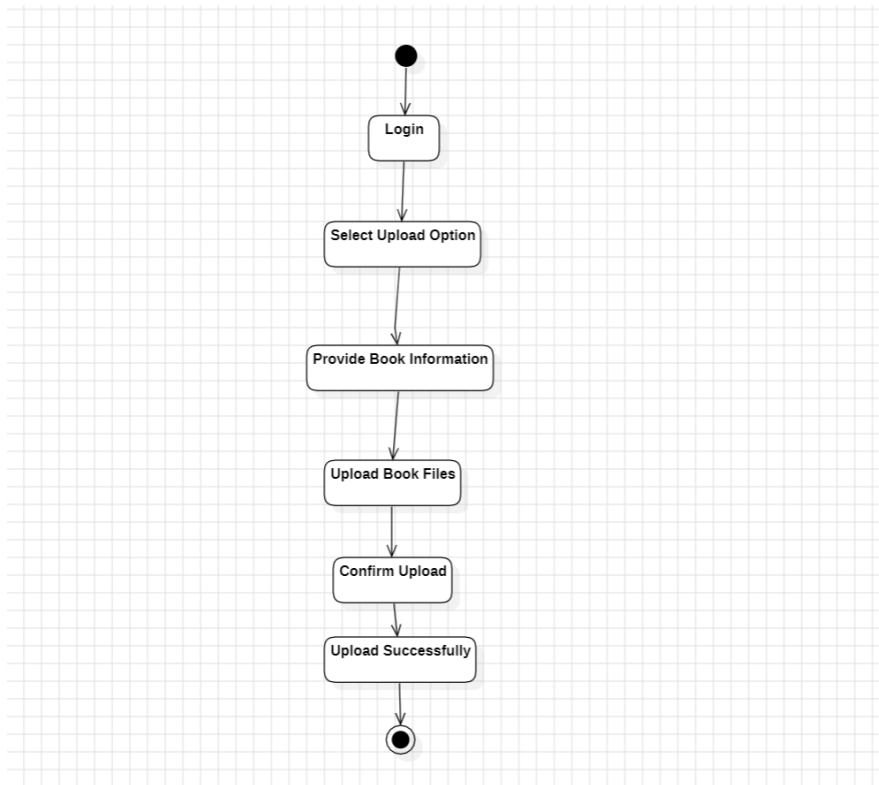
Architectural design

We Use Client Server Architecture

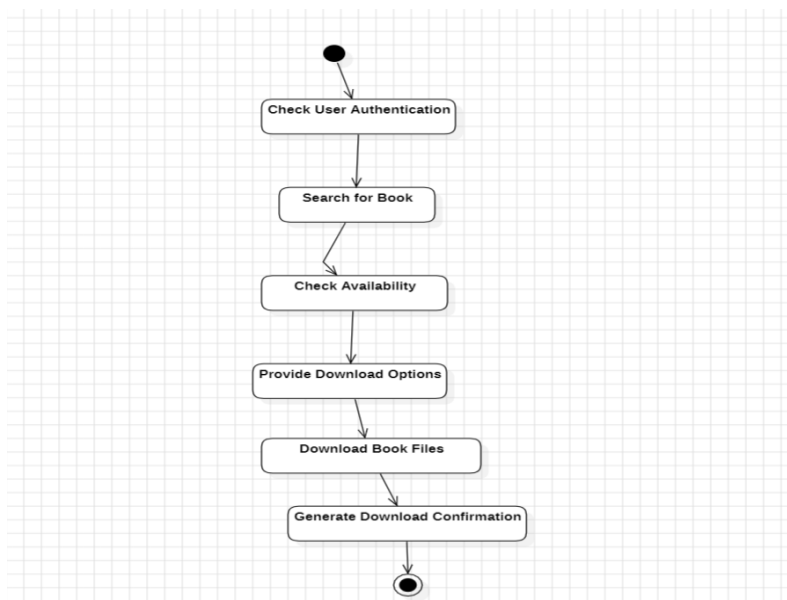


Process flow/Representation

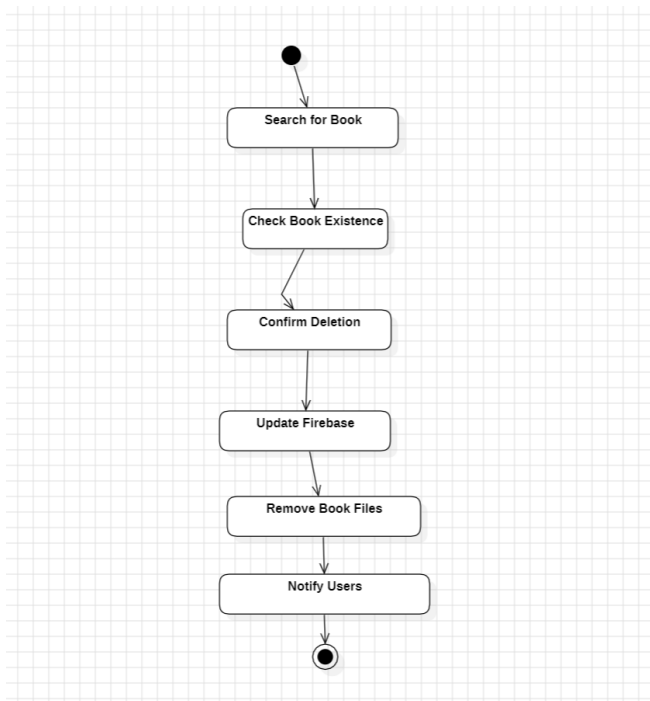
Upload Book Activity Diagrams



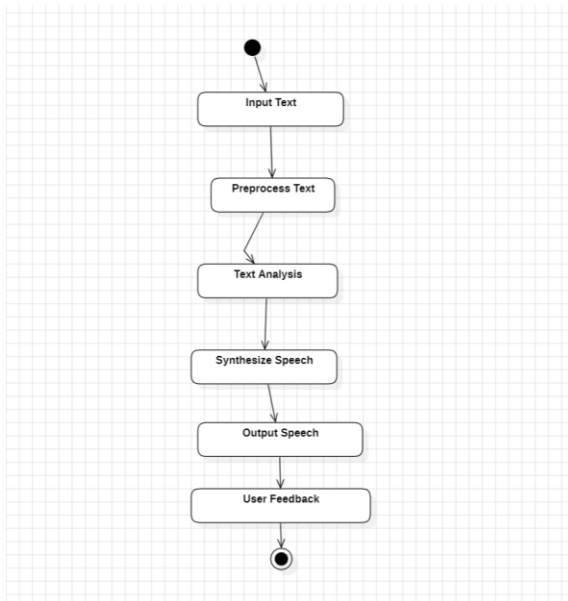
Download Book Activity Diagram



Delete Book Activity Diagram

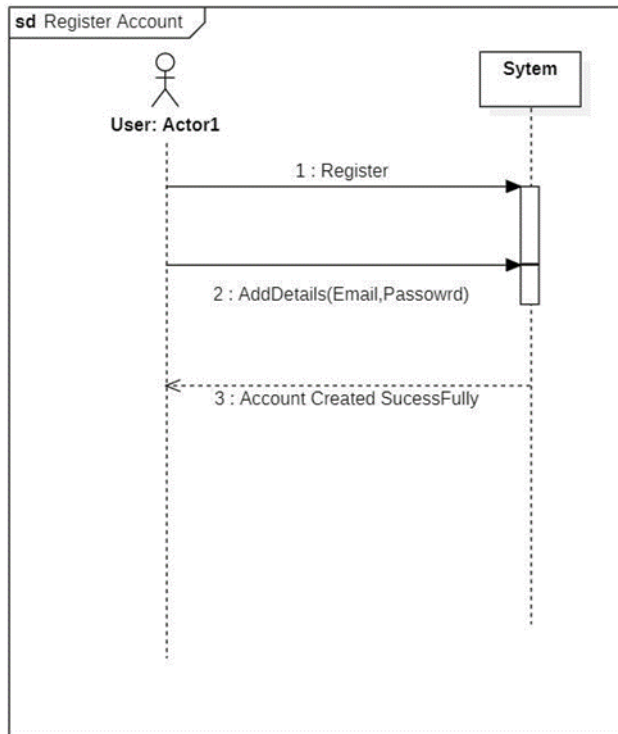


Text To Speech Recognition Activity Diagram

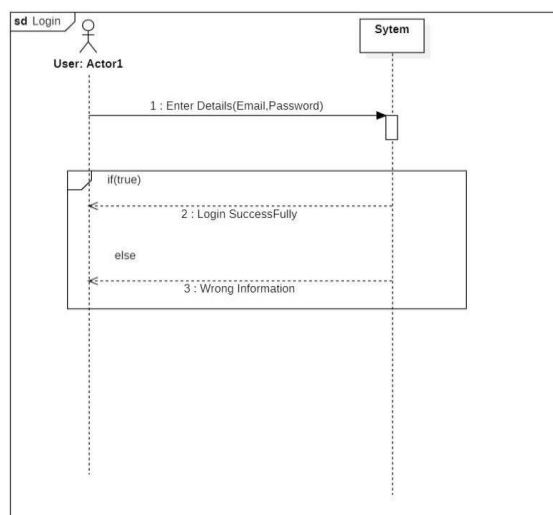


Design models

System Sequence Diagram Register Account

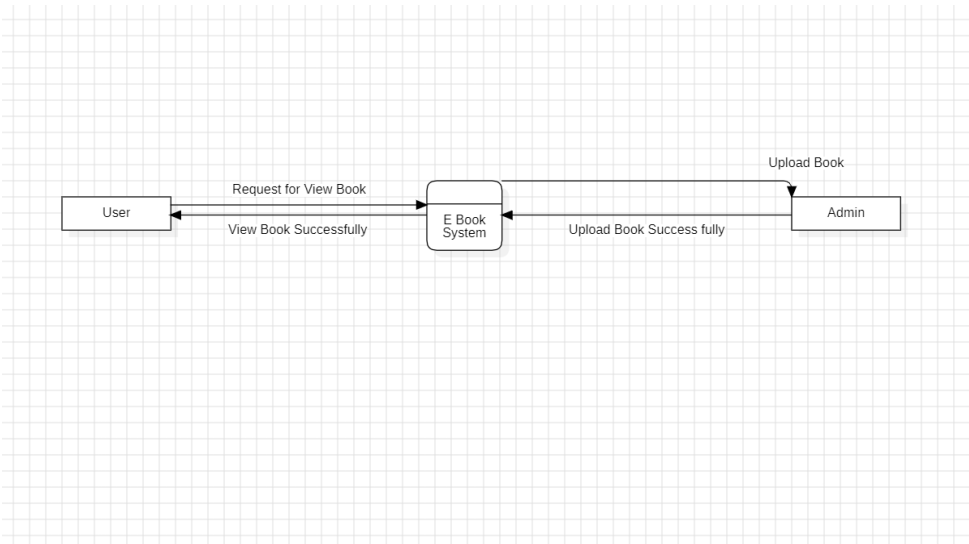


Login

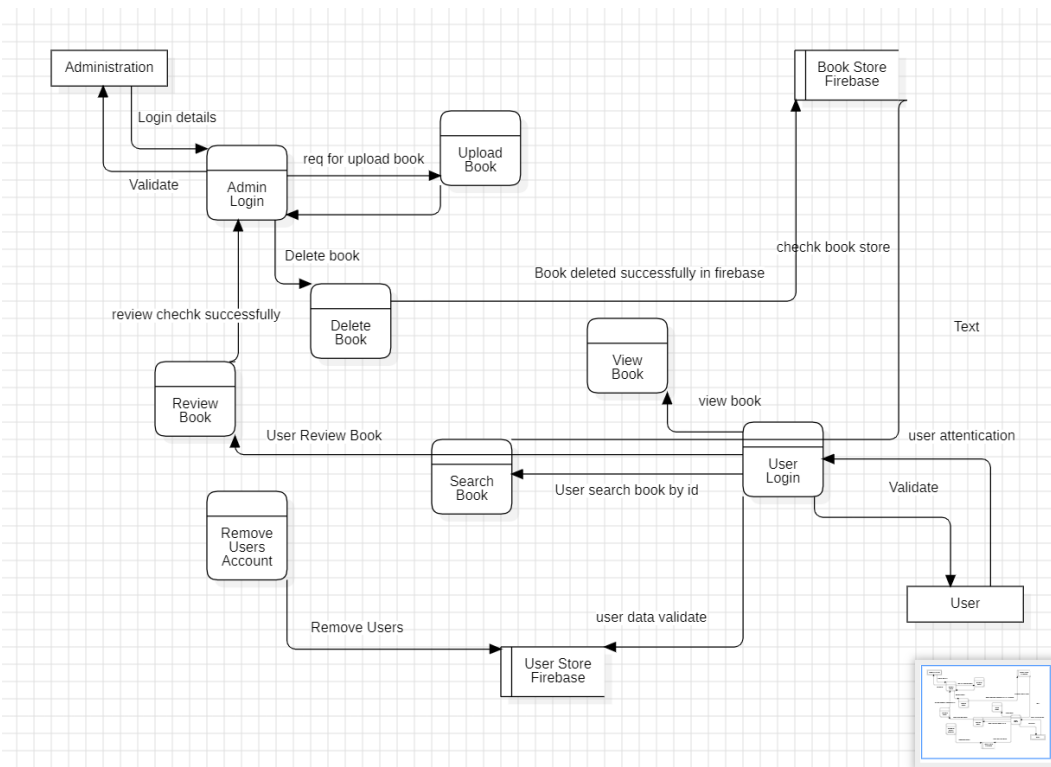


DFD

Level 0

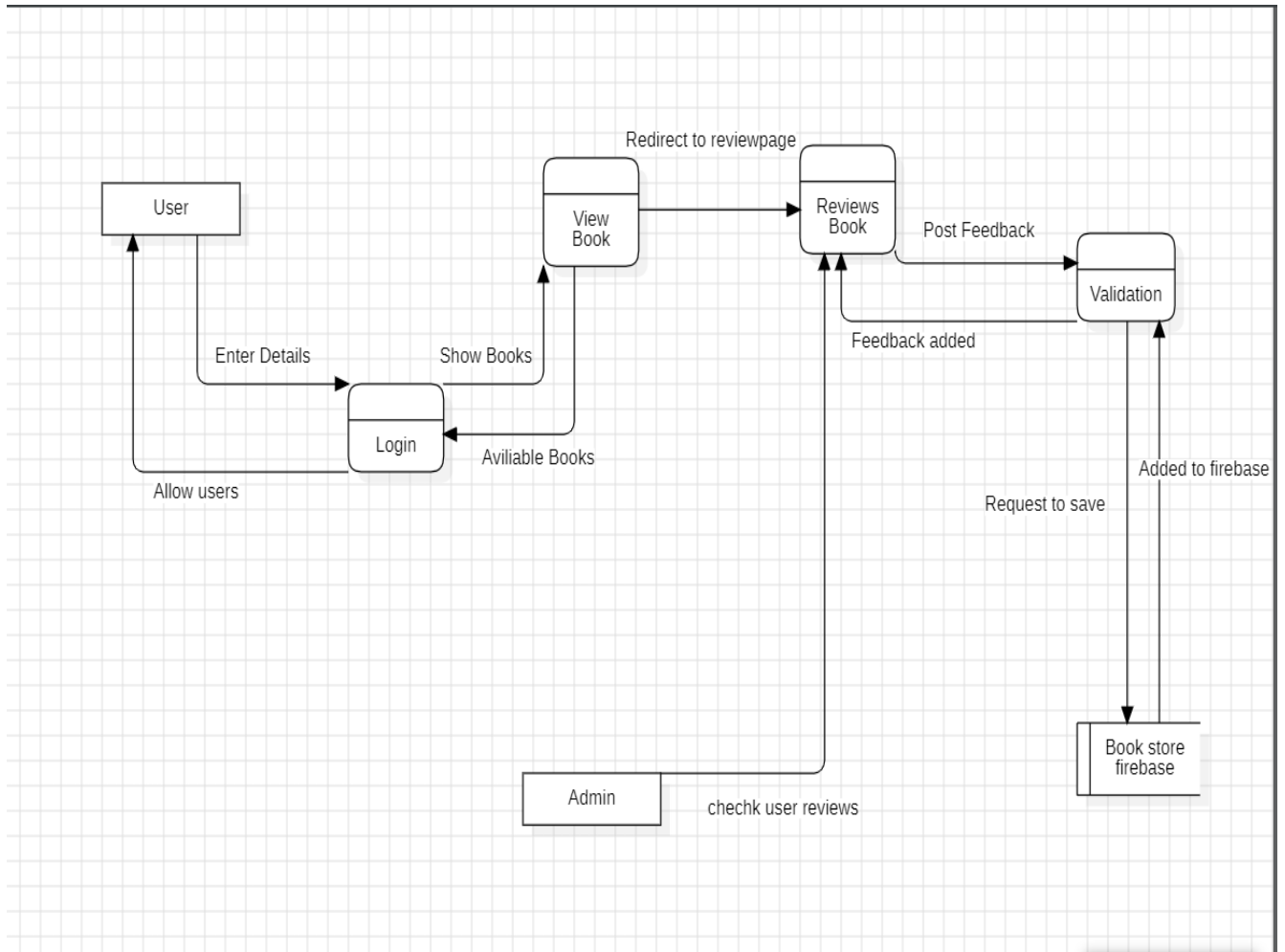


Level 01



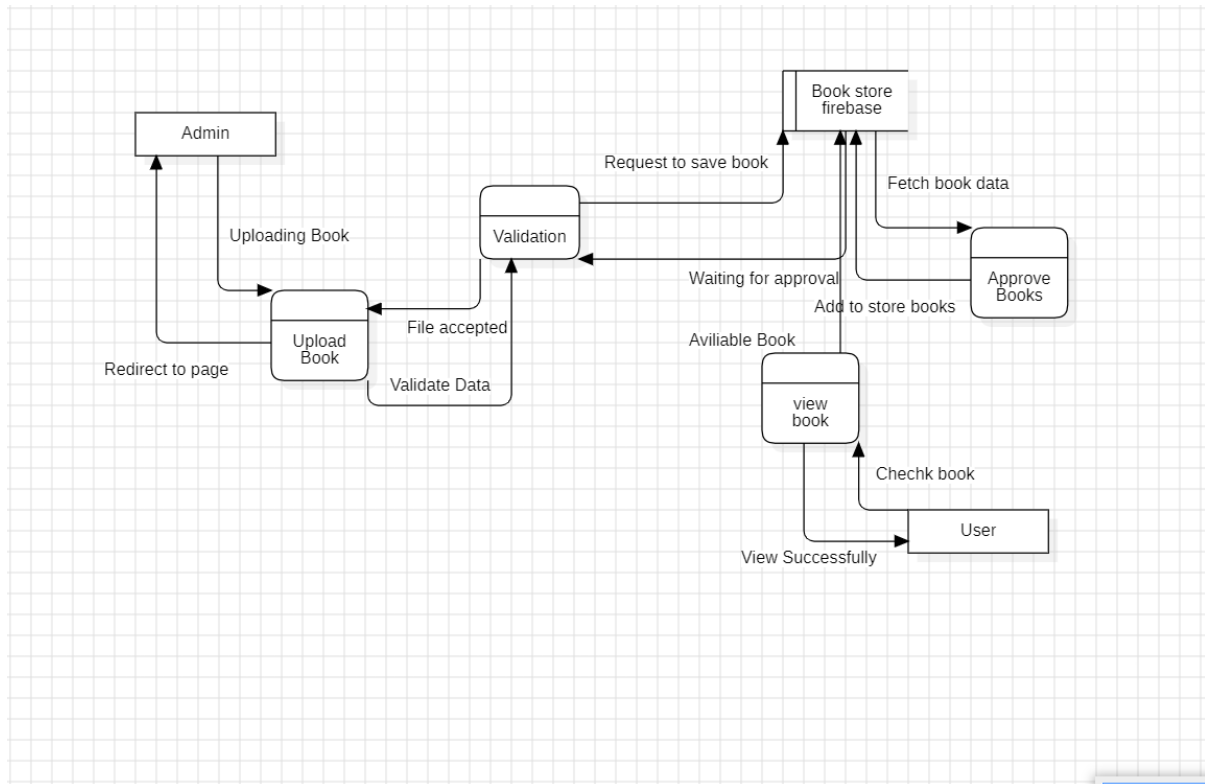
Level 02

Review Books



Level 02

Upload Book and View Book



Data design

1. **Users:**
 - Firebase Authentication can handle user registration and login securely.
 - Firebase Realtime Database or Fire store can store additional user data such as preferences, download history, and user-specific settings.
2. **Books:**
 - Each book can be represented as a document or a node in the database.
 - Attributes like title, author, genre, description, and file location can be stored within each book document.
3. **Downloads:**
 - Use Firebase Realtime Database or Fire store to track downloads.
 - Create a node for downloads with user IDs as keys and book IDs as subkeys, recording timestamps and download status.
4. **User Preferences:**
 - Store user preferences within each user's document in the Firebase Realtime Database or Fire store.
5. **Reviews and Ratings:**
 - Create nodes for reviews and ratings, associating them with the respective books.
 - Each review and rating can be a document with attributes like user ID, book ID, rating, review text, and timestamp.
6. **Bookmarks:**
 - Similar to downloads, use Firebase Realtime Database or Fire store to store bookmark information.
 - Create a node for bookmarks with user IDs as keys, book IDs as subkeys, and bookmark locations.
7. **Analytics Data:**
 - Firebase Analytics can be integrated to collect usage statistics, user engagement metrics, and other relevant analytics data.

Data dictionary

1. **User**
 - Type: Entity
 - Description: E Book Reader and Downloader Mobile App.
 - Attributes:
 - UserID: Unique identifier for the user.
 - Username: User's username for authentication.
 - Email: User's email address.
 - Password: User's password (encrypted).
 - Preferences: User's dietary preferences or Downloading interest.
2. **Upload Book**
 - Type: Process

- Description: Represents the process of uploading a book to the E-Book Reader and Downloader mobile app
- Attributes:
 - BookID: Type: Unique identifier (Integer or String).
 - Title: String.
 - File Location: String.
 - Genre: String.

3. Download Book

- Type: Process
- Description: Represents the process of downloading a book from the E-Book Reader and Downloader mobile app.
- BookID: Type: Unique identifier (Integer or String).
- Title: String.
- File Location: String

4. Review Book

- Type: Process
- Represents the process of reviewing a book within the E-Book Reader and Downloader mobile app.
- ReviewID: Type: Unique identifier (Integer or String).
- BookID: Type: Identifier of the book being reviewed (Integer or String).
- UserID: Type: Identifier of the user writing the review (Integer or String).
- Rating: Type: Numeric rating score given by the user (e.g., 1-5).
- Comment: Type: Textual comment or feedback provided by the user.

Algorithm & Implementation

User Authentication:

Utilize Firebase Authentication for secure user registration and login. Implement functions to register users with email and password and to authenticate user login.

Books Management:

Use Firebase Realtime Database or Firestore to store book information. Implement functions to add books with attributes like title, author, genre, and file location. Retrieve books from the database to display in the application.

Downloads Tracking:

Employ Firebase Realtime Database or Firestore to track user downloads. Create a node for downloads with user IDs and book IDs, recording timestamps and download status. Provide functions to track downloads and retrieve download history for a user.

User Preferences:

Store user preferences within each user's document in Firebase Realtime Database or Firestore. Implement a function to update user preferences based on user input.

Reviews and Ratings:

Create nodes for reviews and ratings, associating them with respective books. Implement functions to add reviews and ratings with user ID, book ID, rating, and review text. Retrieve reviews and ratings for a specific book to display in the application.

.

Software requirements traceability matrix

Table 1: Requirements Traceability Matrix

Req. Number	Ref. Item	Design Component	Component Items
FRO1	Dfd Diagram	Dfd Name	Function Name(s)
FR02	Use Case Diagram	Use Case Name	Component Name(s)
FR03	Sequence Diagram	Diagram Number	Function Name(s)
FRO4	Data Dictionary	Data Item Name	Component Name(s)
FR05	Algorithm	Algorithm Name	Function Name(s)
FR06	Performance	Performance Metrix	Component Name(s)
FR07	Usability	Usability Metrix	Component Name(s)
FRO8	Security	Security Measure	Component Name(s)
FR09	Error Handling	Error Handling Step	Component Name(s)

Human interface design

1. User Registration and Login:

- **Functionality:**
 - Users can register for a new account using their email and password.
 - Existing users can log in securely.
- **User Interaction:**
 - Access the registration/login interface in the app.
 - Provide email and password for registration/login.
- **Feedback:**
 - Receive confirmation messages for successful registration and login.
 - Get error messages for unsuccessful attempts with appropriate guidance.

2. Browse and Discover Books:

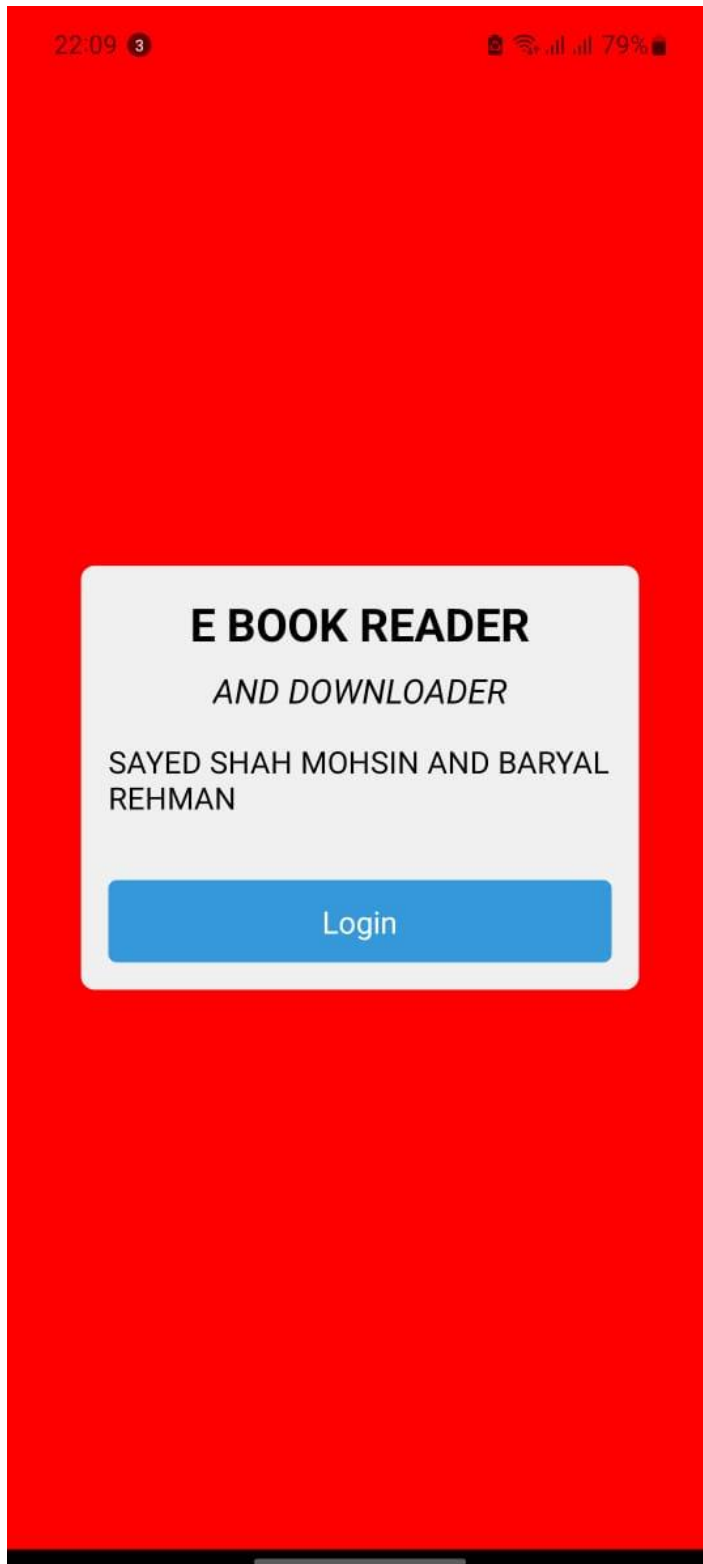
- **Functionality:**
 - Users can explore a collection of books with details like title, author, and genre.
- **User Interaction:**
 - Navigate through the app to view a list of available books.
 - Click on a book to view detailed information.
- **Feedback:**
 - Visual cues (thumbnails, titles) for an engaging browsing experience.
 - Information-rich book details for informed decision-making.

3. Download and Read Books:

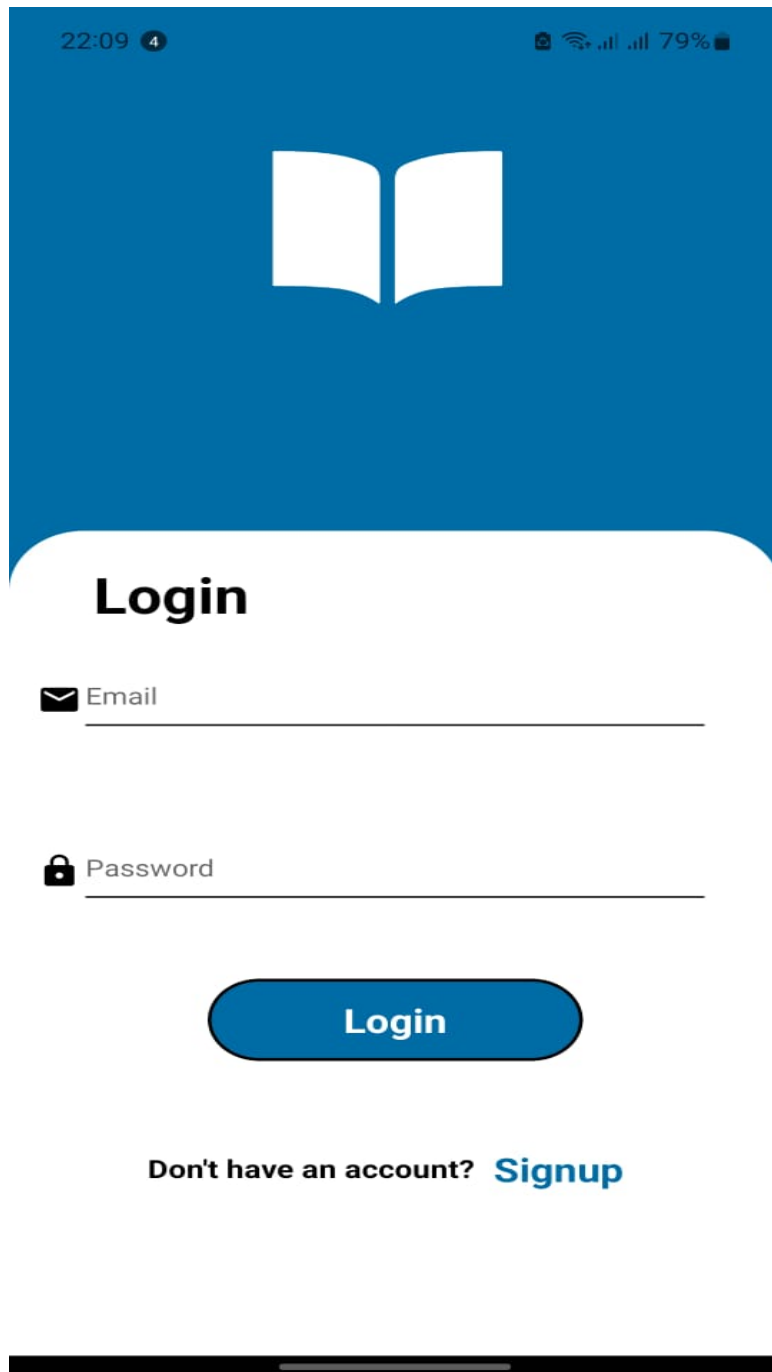
- **Functionality:**
 - Users can download books for offline reading.
 - Read books within the app.
- **User Interaction:**
 - Click on the download button next to a book.
 - Access downloaded books from a personal library.
- **Feedback:**
 - Progress indicators during the download process.
 - Clear indications of downloaded books in the library.
- 4. **Review and Rate Books:**
 - **Functionality:**
 - Users can provide reviews and ratings for books.
 - **User Interaction:**
 - Navigate to the book's detail page.
 - Submit a review and rating with textual comments.
 - **Feedback:**
 - Confirmation of successful submission.
 - Display of aggregated ratings and individual reviews.
- 5. **Manage User Preferences:**
 - **Functionality:**
 - Users can set preferences such as preferred genres.
 - **User Interaction:**
 - Access the user profile or settings section.
 - Choose and update preferences.
 - **Feedback:**
 - Acknowledgment of successful updates.
 - Guidance for modifying preferences.
- 6. **Track Downloads and Bookmarks:**
 - **Functionality:**
 - Users can view their download history and manage bookmarks.
 - **User Interaction:**
 - Access a user-specific dashboard or section.
 - View download history and bookmarks.
 - **Feedback:**
 - Clear display of download history and bookmarked books.
 - Option to remove or organize bookmarks.
- 7. **View Analytics Data:**
 - **Functionality:**
 - Users can access analytics data related to their usage and engagement.
 - **User Interaction:**
 - Navigate to an analytics or insights section.
 - Explore visualizations and data summaries.
 - **Feedback:**
 - Informative charts and insights into reading habits.
 - Guidance on improving the reading experience based on analytics.

Screen images

Home Screen




Login Screen





A mobile app login screen with a blue header and a white login form. The header features a white book icon. The form includes fields for Email and Password, a Login button, and a link to Signup. The status bar at the top shows the time 22:09, a notification icon, and a 79% battery level.

22:09 4 79%



Login

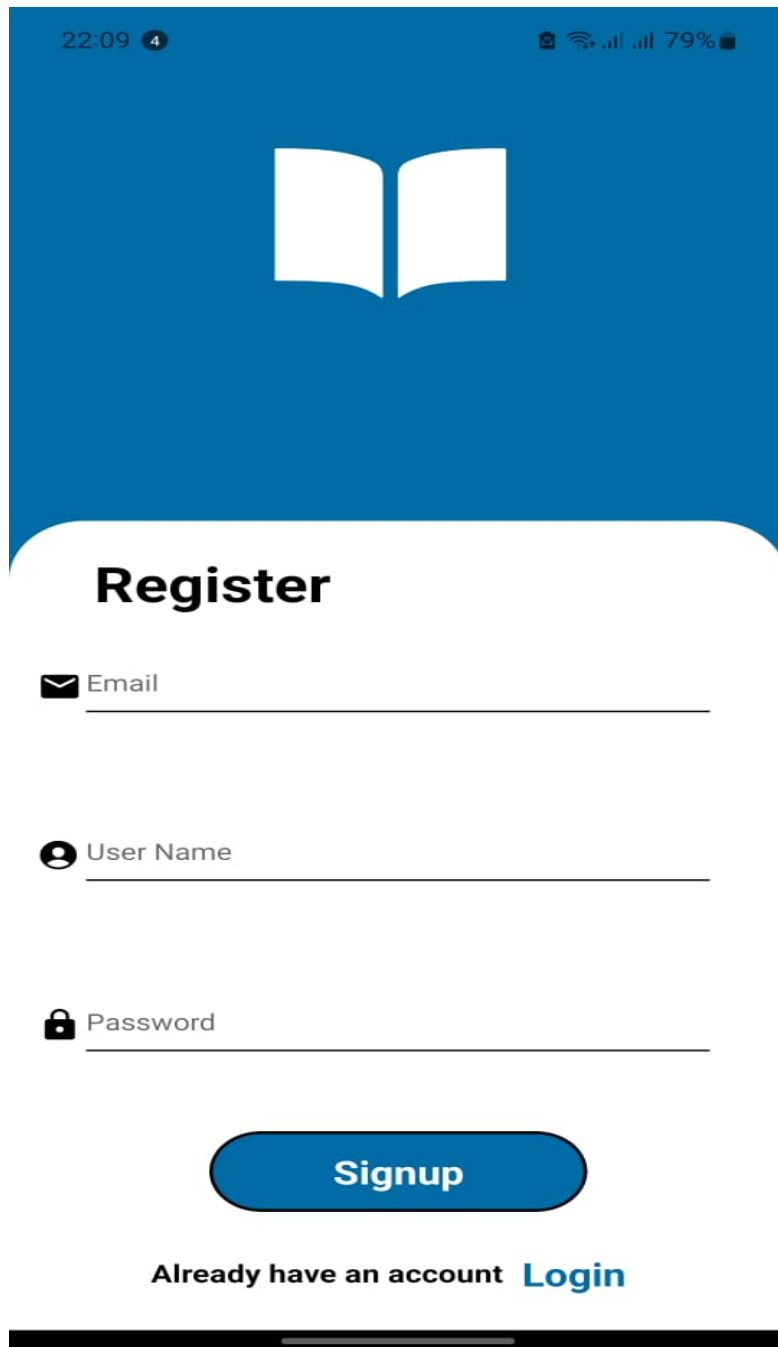
 Email

 Password

Login

Don't have an account? [Signup](#)

Registration Screen



The image shows a mobile app registration screen. At the top, there is a blue header with a white icon of an open book. Below the header, the word "Register" is displayed in a large, bold, black font. Underneath, there are three input fields: "Email" with an envelope icon, "User Name" with a person icon, and "Password" with a lock icon. Each field has a horizontal line for text entry. Below the input fields is a blue rounded rectangular button with the text "Signup" in white. At the bottom, there is a link that says "Already have an account Login", where "Login" is in blue. The entire screen is set against a white background with a blue header and a black home indicator bar at the very bottom.

22:09 4 79%

Register

Email

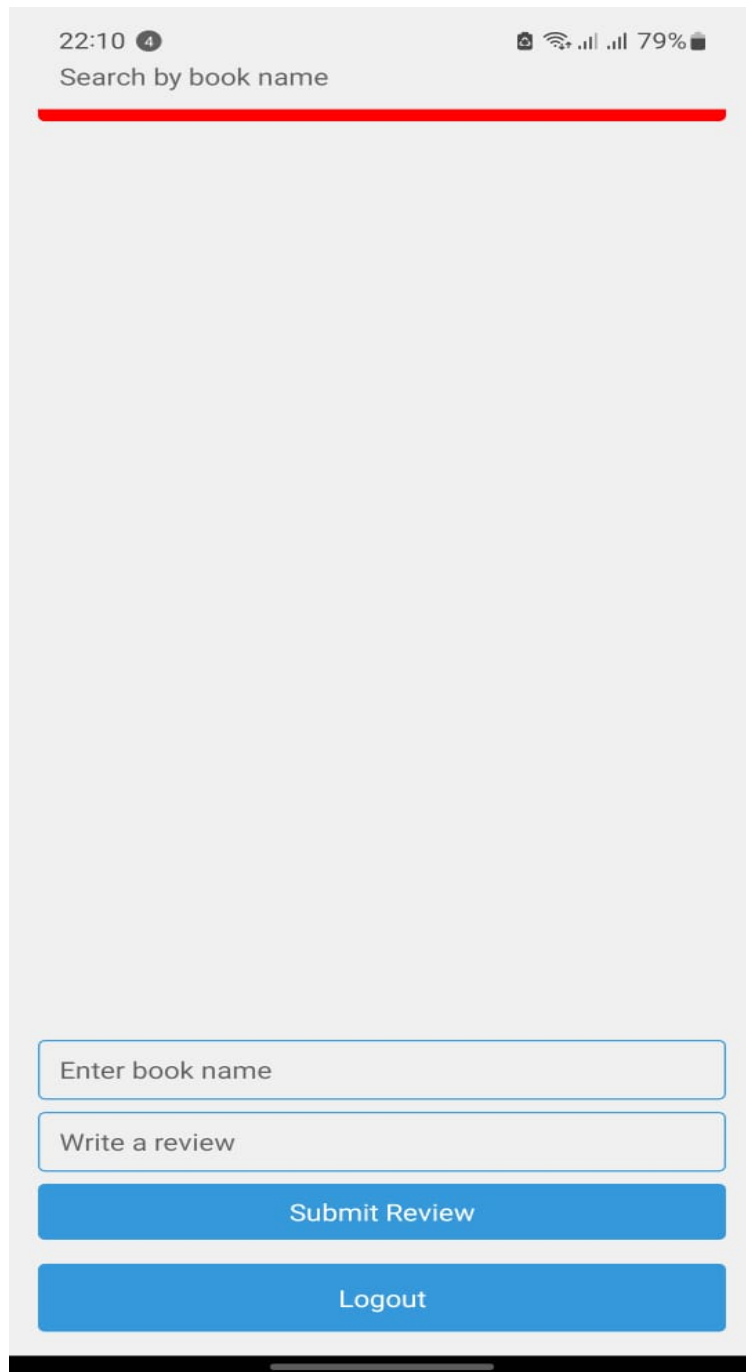
User Name

Password

Signup

Already have an account [Login](#)

Dashboard Screen

A mobile application dashboard screen with a light gray background. At the top, a status bar shows the time 22:10, a notification icon with the number 4, and connectivity icons for Wi-Fi, cellular signal, and 79% battery. Below the status bar is a search bar with the placeholder text "Search by book name" and a red horizontal line underneath. The main area is mostly empty. At the bottom, there are four input fields and two buttons. The first input field is labeled "Enter book name". The second input field is labeled "Write a review". Below these are two blue buttons: "Submit Review" and "Logout".

22:10 4

Search by book name

Enter book name

Write a review

Submit Review

Logout

22:10 4

79%

Book



Maximum
PC

Download



Popular
Science

Download



Popular
Mechanics

Download

L. Watson

The World Book
Encyclopedia

The World
Book Encycl
opedia

Download



Book Lovers

Download



Ebony

Download



Enter book name

Write a review

Submit Review

Logout

Appendix I

How to design using UML (OOP): For guidance please follow the instructions mentioned in the link: <http://agilemodeling.com/artifacts/>

Data flow diagrams: For guidance please follow the instructions mentioned in the link and book:

<http://www.agilemodeling.com/artifacts/dataFlowDiagram.htm>

Software Engineering –A Practitioner’s approach by Roger Pressman

Architecture diagram: For guidance please follow the instructions mentioned in the link and book: Ian Sommerville – Software Engineering 9th Edition– Chapter 6