ReLis
Let the Book Talk!!!

Dharmil Gada
1814076
I.T. , K.J.S.C.E.
gada.d@somaiya.edu

Dev Vora
1814120
I.T. , K.J.S.C.E.
dev.vora@somaiya.edu

Jainam Zobaliya
1814123
I.T. , K.J.S.C.E.
jainam.z@somaiya.edu

Mentor
Prof. Purnima Ahirao
K.J.S.C.E.
purnimaahirao@somaiya.edu

November 5, 2021

Abstract Libraries are used to store books, but we need a system to navigate to a specific book effortlessly. Moreover, it is not possible for all book lovers to visit the library and buy or rent books, so our aim is to design and create a the smart E-Library system helps to solve this issue. ReLis is a smart e-library system which is Eco-friendly as there is no involvement of hard copies and thus, it saves paper. The User can buy/rent books that are available. Keeping in mind the current situation and growth of online learning this system and its features will prove to be very beneficial to the users. Also this system has a very user friendly interface. Thus the users will feel very easy to use it. By using this system admin can add, update, delete and publish a book after reviewing. The system also provides translation and audio-book facilities which are very helpful to the user.

Keywords: ReLis, Flutter, Node JS,Books, Digital Library, E-Library

TABLE Of CONTENTS

List of Figures

Chapter 1

INTRODUCTION

1.1 Problem Statement

Libraries are used to store books, but we need a system to navigate to a specific book effortlessly. Moreover, it is not possible for all book lovers to visit the library and buy or rent books, so the smart E-Library system helps to solve this issue. ReLis is a smart e-library system which is eco-friendly as there is no involvement of hard copies and thus, it saves paper. This platform allows users to search books using specific keywords, titles, author names, genres, etc. Also, this platform has a user-friendly interface, allows users to buy, rent and publish books and also can upload the book. This system has a feature which provides credits to the user on performing certain activities (for example: watch ad, invite friend, daily login rewards) which can be further used to buy or rent books/audiobooks. The platform also provides graphical statistics based on the user's reading history. The user would get a personalized experience by the addition of a recommendation feature , which would suggest books as per the user preferences. The feature of the recommendation system personalizes the user experience by suggesting new books to explore. Based on the current scenario, screen time for most people has increased drastically which causes negative effects on our body and thus the feature of audiobooks is beneficial to users. One can also get translated book where few books are translated and stored in database and if any another book is requested, the translation will take place and the user can enjoy the book later after it gets translated. Similarly process takes place for audio book.

1.2 Scope Of Project

1. Easy Registration.
2. Choose from variety of Genre.
3. Mark your books as favourite, or add it in wish-list.
4. Track your History.
5. Easy payment facility (If time Permits)

1.3 Motivation

The motivation behind having such a Smart E-Library system: ReLis was keeping in mind the constant increase in the technology and people preferring everything online from the

comfort of their homes. Nowadays because of the Novel CoronaVirus everything is going on an online mode and the idea behind implementing such a project is to allow users to read and upload books from the comfort of their homes. Digital libraries promise new societal benefits, especially for e-learning in digital or mobile times, starting with the elimination of the time and space constraints of traditional libraries. Also, recent developments in Internet and WWW have brought significant change in the way the generation, distribution and the accessing of enormous amount of information published through various modes and means. And today digital libraries have overtaken managing the information world using both commercial and open source software. So we thought about the ease of the people by having ReLis and providing them with a wide range of features.

Chapter 2

SOFTWARE PROJECT MANAGEMENT PLAN - SPMP

2.1 Product

2.1.1 Product Overview

ReLis is the system which is user-friendly as well as eco-friendly as there in no involvement of hard copies and thus, it saves paper. The platform allows users to search books using specific keywords, titles, author names, genres, etc. Also this platform allows users to buy, rent, upload a book and even publish their book. The system also provides credits to users on performing certain activities like daily login, reffering etc. These credits later can be used to buy/rent books. There is a Dashboard for users where they can view their details, reading history, graphical statistics, etc.

For users to have a personalised experience, there is a recommendation feature which would suggest books as per the user preferences. If a user is not comfortable reading (like if they are travelling/ in a crowded area) then the Audiobook feature helps users and they can listen to the audio and enjoy their book. This feature is also useful for blind people. Also if someone is not able to understand the language or wants to read book in other languages then the book can be translated so that one can read the book easily in the language they are comfortable. Also, based on the current scenario, screen time for most people has increased drastically which causes negative effects on our body and thus the feature of audio books is beneficial to users.

2.1.2 Project Deliverables

a) Project Approval PPT – 24/09/2021

b) Project Scope, Functional And Non-Functional Requirements – 25/09/2021

c) Software Requirements Specification (SRS) – 24/09/2021

d) Software Project Management Plan (SPMP) – 24/09/2021

e) Software Design Document (SDD) – 08/10/2021

f) Software Test Document (STD) – 15/10/2021

g) Creating the user interface and Backend of the Website which includes the various modules such as:

i) Sign-Up and Sign-In,

ii) HomePage,

iii) User Profile Page,

iv) Admin Profile Page,

v) View, Add, Update and Delete a (all) book,

vi) View and Block a User.

vii) Buy or Rent any Book.

viii) Mark any book as Favourite and Add to Wish-list.

ix) View all Trending Books, Recommended Books, Favourite books, Wish-list, Different Genre.

x) Read any book and Listen to available Audio-book.

xi) Add a book to his Personal Collection by uploading it in PDF format and the system will also provide it in Audio form.

xii) Publish their own books after getting reviewed from admins.

xiii) Publish a User's book after reviewing the book.

xiv) Reading Statistics of a user.

xv) Search for a book using the barcode.

2.2 PROJECT ORGANIZATION

2.2.1 Software Process Model

The prototype model will be used that will help in appropriate satisfaction of user's need. Initially, a prototype of the product with minimal functionality will be delivered to the customer. This prototype will be going through changes as per customer's requirements after taking feedback from the customer after every incremental update of the prototype.

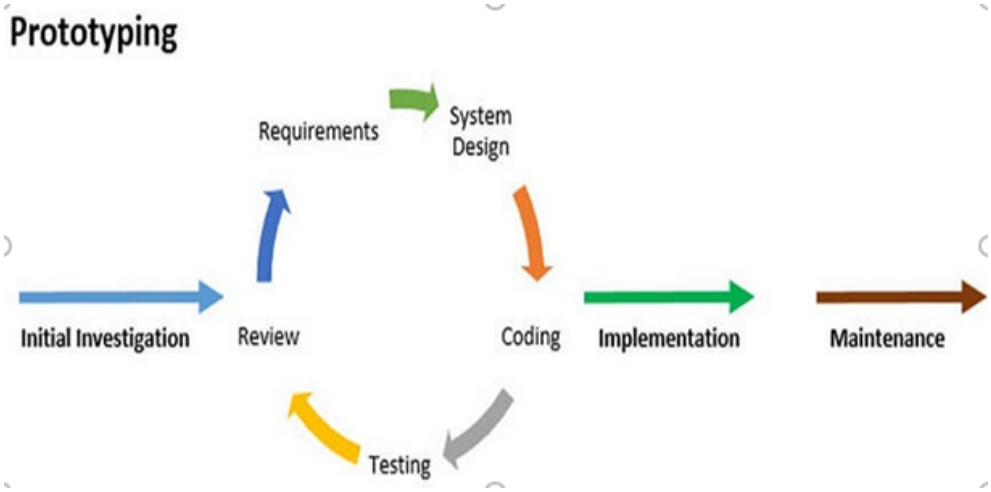


Figure 2.1: Prototyping Model

Initially the requirement analysis will be done by communicating with the stakeholders. After that the SRS, SPMP, and SDD documents will be created. Then the building of prototype will be done. The prototype building phase will go in the cycle of the prototyping model i.e. beginning with communication, quick planning, modelling, construction of prototype and finally deployment, delivery and feedback. The main versions of the product will include:

1. Creating the user interface of the Project
2. Establishing backend of the project

2.2.2 Roles and Responsibilities

Role	Description	Team Member
Project Manager	Overall planning, risk analysis, execution planning, monitoring and closure planning	Dharmil Gada
System Analyst	To develop a design of the required system and ensuring that all requirements are satisfied	Dharmil Gada, Dev Vora
Technical Lead	Leading the development team	Jainam Zobaliya
Development Team	Implementing the functionalities required in the application through coding	Dev Vora, Jainam Zobaliya, Dharmil Gada
Database Administrator	To design and maintain the database and to add, modify, delete, retrieve data from the database efficiently	Jainam Zobaliya, Dev Vora
UI/UX Designer	To prepare a prototype of the website and UI	Dev Vora, Jainam Zobaliya
Testing Team	To check whether the functionalities work as expected and to find out bugs in the system	Dev Vora, Jainam Zobaliya, Dharmil Gada

2.2.3 Tools and Techniques

Activity	**Tools required**	**Technique to be used**
Documentation/User Manual	Microsoft word, PDF converter, GanttProject, LaTeX	The procedure would be given in steps wherever necessary labeled images and diagrams will be used
UI/UX Designing	Marvel App	The prototype will be first prepared and shared with the client. After the design is approved implementation of the same will be performed
Development	Android Studio, Dart, Flutter and Node.js, Firebase, Git and Github, Payment Gateway	The development teams would be working on their prototype for development
Meetings	Microsoft Teams, Google Meet, ZOOM	The agenda of the meeting along with the meeting link would be shared with all the attendees prior the meeting
Testing	Dart, Flutter	Test cases will be prepared for the testing team to conduct the tests

2.3 PROJECT MANAGEMENT PLAN

2.3.1 Tasks

a) **Task 1- Requirements analysis**

i) Description: Analysing the project requirements by communicating with the user. Accordingly preparing the SRS document. It is done to know the exact expectations of the client from the product. By doing sufficient and effective requirement analysis, the functionalities and working of the product would also be clear.

ii) Deliverables and Milestones: This document is made so that the design and development team will be sure of the functioning of the WebApp. This document briefs members of the team as well as the clients about specifications, management plan and functionalities of the software project.

iii) Resources Needed: Conducting multiple meetings with the stakeholders of the WebApp. Organization of brainstorming sessions of joint discussions for effective communication and information gathering.

iv) Dependencies and Constraints: Creation of SRS cannot be completed until the user specifies the requirements in detail. Also the SRS document cannot be finalized before the approval of the user and his expectations of the WebApp.

v) Risks and Contingencies: Failing to communicate with the organisations involved and the users will be the major risk. Miscommunication between the stakeholders and developers/designers can be another problem but it can be tackled by conducting multiple sessions and creating well defined SRS and getting it approved by the user. If the SRS document isn't well defined i.e addressing each and every aspect of the project is not done correctly, then major miscommunication and false information transfer could take place. The expectations and deliverables would have drastic difference between them.

b) **Task 2- Software Project Management Plan**

i) Description: Once the requirements are finalized, the SPMP is created to get a clear picture of the project for the developer. This document defines the management and technical constraints. It also contains project estimation techniques used, Gantt chart representation, team structure. It also comprises of risk management plan that tells the possible risks and if any of which is encountered, how it should be tackled.

ii) Deliverables and Milestones: Effective planning of the project involving work breakdown structure, risk management policies, Gantt chart for showing the structure of the schedule. Project resource, cost and duration estimates are also given. All software and hardware project resources along with some special libraries or resources are also given.

iii) Resources Needed: Communication with the user and meetings with the developers so as to meet the user's requirements.

iv) Dependencies and Constraints: If the requirements document is not approved by the user, we cannot begin to prepare SPMP.

v) Risks and Contingencies

c) **Task 3- Creating the Login Module.**

i) Description: The frontend will be created using Dart and Flutter Framework. The backend of the same will be created to store the information in a database.

ii) Deliverables and Milestones: The deliverables for this particular task will include the source code of the pages and also the databases, one in which the registration information is stored in the database and login for user/admin is provided after entering the correct credentials.

iii) Resources Needed: Developer needs to be provided with the user and admin details so that he can assign the roles and store in database for checking during login.

iv) Dependencies and Constraints: The login page is necessary for the user to access his/her private details and also to use the website.

v) Risks and Contingencies: If there is any error or access is provided despite wrong credentials being entered user data is at risk so appropriate security measures needs to be taken and menu should be kept well updated.

d) **Task 4- Creating the Admin Profile Module.**

i) Description: The module will contain admin rights. The frontend will be created using Dart and Flutter Framework. The backend of the same will be used to get the information from the database.

ii) Deliverables and Milestones: The deliverables for this particular task will include the source code of the pages and also the databases, one in which the details of all users/books is stored in the database and these details are shown to user when they come to appropriate tab at Admin Profile module.

iii) Resources Needed: Developer needs to be provided with the user and book details so that the retrieval of required details from the database is accomplished.

iv) Dependencies and Constraints: The Admin Profile module is for the admin to view admin options (view, update/delete) details of book/users and others if any.

v) Risks and Contingencies: If there is any error or details are wrong or not updated than data is at risk so appropriate security measures needs to be taken and data should be kept well updated.

e) **Task 5- Creating the Review New Book Module.**

i) Description: The module will contain admin rights to publish a book after the admin reviews it. The frontend will be created using Dart, Flutter Framework. The backend of the same will be used to get the information from the database.

ii) Deliverables and Milestones: The deliverables for this particular task will include the source code of the pages and also the databases, one in which the details of users and the required book to be published is stored in the database and these details are shown to admin when they come to appropriate tab at Admin Profile module.

iii) Resources Needed: Developer needs to be provided with the user and book details so that the retrieval of required details from the database is accomplished.

iv) Dependencies and Constraints: This module is for the admin to publish books if any user posts it and publish it after reviewing it.

v) Risks and Contingencies: If there is any error or details are wrong or not updated than data is at risk so appropriate security measures needs to be taken and data should be kept well updated.

f) **Task 6- Creating the View/Remove/Update User Module.**

i) Description: The module will contain admin rights to Add / Read / Update / Delete a user from the database. The frontend will be created using Dart, Flutter Framework. The backend of the same will be used to get the information from the database.

ii) Deliverables and Milestones: The deliverables for this particular task will include the source code of the pages and also the databases, one in which the details of users is stored in the database and these details are shown to admin when they come to appropriate tab at Admin Profile module.

iii) Resources Needed: Developer needs to be provided with the user details so that the retrieval of required details from the database is accomplished.

iv) Dependencies and Constraints: This module is for the admin to Add / Read / Update / Delete a user from the database.

v) Risks and Contingencies: If there is any error or details are wrong or not updated than data is at risk so appropriate security measures needs to be taken and data should be kept well updated.

g) **Task 7- Creating the View/Add/Remove/Update Book Module.**

i) Description: The module will contain admin rights to Add / Read / Update / Delete a book from the database. The frontend will be created using Dart, Flutter Framework. The backend of the same will be used to get the information from the database.

ii) Deliverables and Milestones: The deliverables for this particular task will include the source code of the pages and also the databases, one in which the details of books is stored in the database and these details are shown to admin when they come to appropriate tab at Admin Profile module.

iii) Resources Needed: Developer needs to be provided with the book details so that the retrieval of required details from the database is accomplished.

iv) Dependencies and Constraints: This module is for the admin to Add / Read / Update / Delete a book from the database.

v) Risks and Contingencies: If there is any error or details are wrong or not updated than data is at risk so appropriate security measures needs to be taken and data should be kept well updated.

h) **Task 8- Creating the User Profile Module.**

i) Description: The module will contain user details. The frontend will be created using Dart and Flutter Framework. The backend of the same will be used to get the information from the database.

ii) Deliverables and Milestones: The deliverables for this particular task will include the source code of the pages and also the databases, one in which the details of user is stored in the database and these details are shown to user when they come to User Profile module.

iii) Resources Needed: Developer needs to be provided with the user and admin details so that he can assign the roles and retrieve the details of user from the database.

iv) Dependencies and Constraints: The User Profile module is for the user to view their details and modify if needed.

v) Risks and Contingencies: If there is any error or user details are wrong or not updated than user data is at risk so appropriate security measures needs to be taken and data should be kept well updated.

i) **Task 9- Creating the Home Page.**

i) Description: This will be the main page which user will see just after login. The frontend will be created using Dart and Flutter Framework. The backend of the same will be used to get the information from the database.

ii) Deliverables and Milestones: The deliverables for this particular task will include the source code of the pages and also the databases, one in which all the information of the books is mentioned.

iii) Resources Needed: Developer needs to be provided with the book details so that the book can be stored in the database and can be shown on the Home page.

iv) Dependencies and Constraints: The Home page is for the user to view current trending books, books recommended for user, top picks etc. The user can pick his/her favourite book and start reading it.

v) Risks and Contingencies: If there is any error or wrong details are entered user data is at risk so appropriate security measures need to be taken and categories should be kept well updated.

j) **Task 10- Creating the Book Preview Module.**

i) Description: The frontend will be created using Dart and Flutter Framework. The backend of the same will be used to get the information from the database.

ii) Deliverables and Milestones: The deliverables for this particular task will include the source code of the pages and also the databases, one in which all the information of the book picked by user is mentioned so as to provide a book preview to the user.

iii) Resources Needed: Developer needs to be provided with the book details so that the book can be stored in the database and the preview can be shown to the user.

iv) Dependencies and Constraints: The Book Preview module is for the user to give some idea about the book like category, book name, author name, description of book etc.

v) Risks and Contingencies: If there is any error or wrong details are entered so appropriate security measures need to be taken and data should be kept well updated.

k) **Task 11- Book History Module**

i) Description: The frontend will be created using Flutter. The backend of the same will be created to store the information in a database.

ii) Deliverables and Milestones: The deliverables for this particular task will include the source code of the pages and also the databases, one in which the information regarding all the previous books rented or bought and read along with the user details, payment and feedback.

iii) Resources Needed Developer needs to be provided with the previous books information of all customers date wise

iv) Dependencies and Constraints: The History module is for the user to view the previous book history and information to check and maintain history of books bought or rented.

v) Risks and Contingencies: If there is any error or wrong details are entered user data is at risk so appropriate security measures need to be taken and categories should be kept well updated.

l) **Task 12- Search Book**

i) Description: The frontend of the search module will be done using Dart and Flutter framework. Backend will have a database with stored information and an API to search a book through barcode

ii) Deliverables and Milestones: The deliverables for this particular task will include the source code of the pages and also the databases, one in which the book information is stored and other of the API where a book and its unique barcode is provided.

iii) Resources Needed: A database used for storing information and Search by Barcode API.

iv) Dependencies and Constraints: The user will have to provide the name of the author or the book or else scan a barcode of that book after which he will be redirected to the page which has that same book's details.

v) Risks and Contingencies: If there is any error or wrong details are entered so appropriate security measures need to be taken and data should be kept well updated.

m) **Task 13- Wish List and Favourites module**

i) Description: The frontend of the pages will be done using Dart and Flutter framework. Backend will have a database to store information based on user choices.

ii) Deliverables and Milestones: The deliverables for this particular task will include the source code of both the pages and also the two databases, one in which the books which are selected as favourites are stored and another with the wish listed preferences..

iii) Resources Needed: A database used for storing information.

iv) Dependencies and Constraints: The user will have to select the books which he wants to keep as wish listed or favourites for easy access from the home page so that every time he does not have to remember and search for the book again and again.

v) Risks and Contingencies: If there is any error or wrong books are entered so appropriate security measures need to be taken and data should be kept well updated.

n) **Task 14- Adding online payment gateway. Create feedback page and store responses in database.**

i) Description: The frontend of the payment gateway will be done using Dart and Flutter framework. Backend will have a database to store information and payment gateway API will be used for transactions.

ii) Deliverables and Milestones: The deliverables for this particular task will include the source code of both the pages and also the two databases, one in which the transaction information is stored and other of the feedback.

iii) Resources Needed: A database used for storing information and payment API.

iv) Dependencies and Constraints: The user will have to pay the bill and a message will be given to the user that bill has been paid after which he will be told to give feedback. Once feedback is given only then can the user go back to the home page.

v) Risks and Contingencies: Safe payment gateway and apt methods for payment needs to be maintained otherwise user privacy and data are at risk.

2.3.2 Assignment

Task	Role	Name
Task -1,4,9,10	Manager,Developer	Dharmil Gada
Task -2,5,8,11,13	Designer,Developer	Jainam Zobaliya
Task -3,6,7,12,14	Tester,Developer	Dev Vora

2.3.3 Time Table

Time Table of the given project

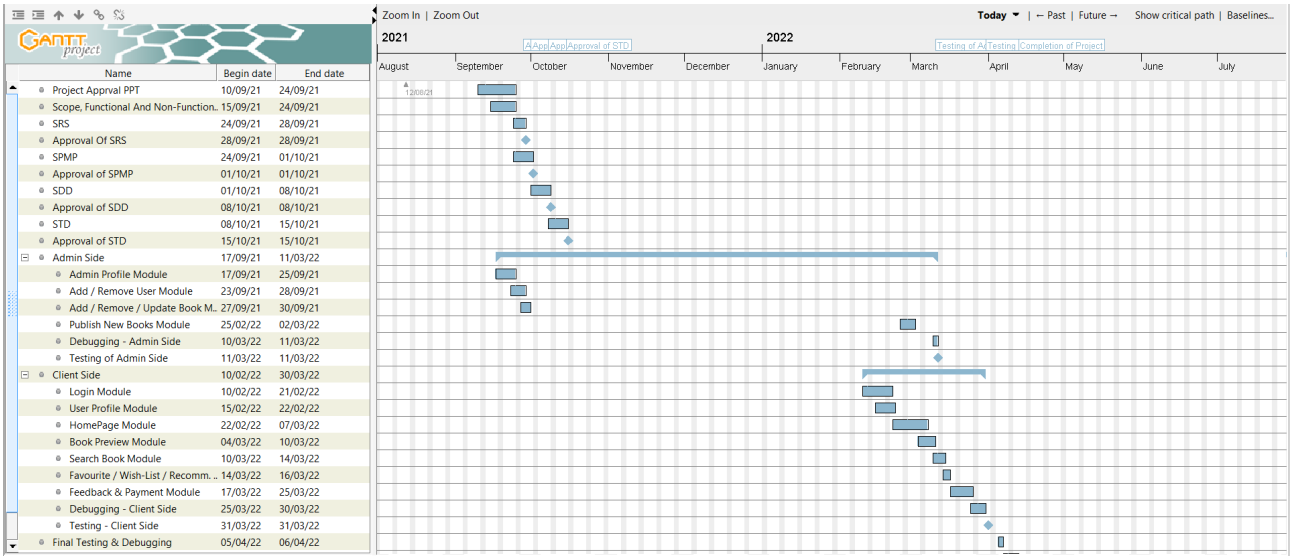


Figure 2.2: Gantt chart

Chapter 3

SOFTWARE REQUIREMENT SPECIFICATION - SRS

3.1 Product Overview

Libraries are used to store books, but we need a system to navigate to a specific book effortlessly. Moreover, it is not possible for all book lovers to visit the library and buy or rent books, so the smart E-Library system helps to solve this issue. ReLis is a smart e-library system which is Eco-friendly as there is no involvement of hard copies and thus, it saves paper. Also, based on the current scenario, screen time for most people has increased drastically which causes negative effects on our body and thus the feature of audio books is beneficial to users.

3.2 Specific Requirements

3.2.1 External Interface Requirements:

User Interfaces

i. **Splash Screen Page:**
In this page (Fig 1),is the loading screen of the system. It shows the logo and the name of our application along with a loading icon.

ii. **SignUp Page:**
In this page (Fig 2), there will be fields to enter the full name, email-id and password followed by a Sign up button to register. Also, there would be option, that if anyone already has an account he/she can be redirected to the Sign-in page.

iii. **SignIn Page:**
In this page (Fig 3), there will be email-id and password followed by a sign-in button to login. Also, there would be option of forgot password or a button for a new user to be redirected to the Sign Up page.

iv. **OTP Page:**
In this page (Fig 4), the user has to enter a 6-digit one time password to verify his credentials. There are buttons to verify the OTP or to request the admin to resend the OTP.

v. **Home Page:**
In this page (Fig 5), the user is able to see the different types of books available to buy displayed in a categorized fashion. Further, tabs are proved for the different genres for easy search of preferences to the user. And lastly, it displays the history of books purchased/rented or read by the user in a suitable manner.

vi. **User Profile Page :**
In this page (Fig 6), the user can view his details which he provided during registration. This page is crucial, since this gives the customer an option to view his data as well as upload any status and also check details such as books bought, rented, books added to the wish-list, cart or his favourites.

vii. **Admin Profile Page:**
In this page (Fig 7), the admins can view the details of all his customers and also has the options to remove or the customer for inappropriate behaviour.Further, he has the extra features to add remove or update any details regarding the books available in the system.

viii. **Favourites Page:**
In this page (Fig 8), the users can view the books which they have added as favourites from the home page.

ix. **Wish List Page:**
In this page (Fig 9), the users can have a look at the books which they are interested in reading at a later time for the ease of the user.

x. **Category Page:**
In this page (Fig 10), there are tabs provided based on the genre of books and when a user clicks on any one tab, the books relevant to that category are displayed.

xi. **Trending Page:**
In this page (Fig 11), the most read or favourite books of the users are displayed on a single page so that the user can easy select the trending boos with out any delay.

xii. **Books Bought Page:**
In this page (Fig 12), books bought by user is shown.

xiii. **Books Rented Page:**
In this page (Fig 13), books rented by user is shown.

xiv. **Personal Books Page:**
In this page, the user can see his personal book collection (Fig 14). And can add a new book by filling the new book details (Fig 15) on clicking the add icon.

xv. **Search Page:**
In this page (Fig 16.), user can search for books by different search commands like by book name, book id, author name, keywords, etc.

xvi. **Book Preview Page:**
In this page (Fig. 17), information about books like author name, book name, price, genre, description, cover image is shown. Along with options to add book to the users favourite book collection or to wish-list, or to read/listen book if book is bought or rented or else the option to buy or rent the book.

xvii. **PDF reader Page:**
In this page, the book PDF is displayed.

xviii. **Audio book Page:**
In this page, the audio playback of the book is shown.

xix. **Bill/Feedback Page:**
In this page, the user can see the bill for the books bought/rented and can also rate the book or can add comments.

Hardware Interfaces:

The web app will work in the supported OS on any smart device. Following are minimum requirements of hardware required:

i RAM: 1GB

ii Screen pixel density: 350px

iii Connectivity: Wi-Fi connectivity or mobile data network.

Software Interfaces:

i Any Electronic Device- above v5.0

ii Android Studio - v4.2.2

iii Visual Studio Code - v1.60

iv Flutter (To make mobile application) - v1.25.0-8.2

v Dart SDK version: 2.12.0-133.2

vi GIT Version Control System- v3.2.0

vii Node JS: v14.17.6

viii Browser - We have chosen Chrome Browser for its best support and user-friendliness. (v93.0.4577.82)

Communication Protocols:

Required protocol for connectivity:

i Hyper-Text Transfer Protocol (HTTP),

ii Hypertext Transfer Protocol Secure (HTTPS),

iii Domain Name System (DNS),

iv Dynamic Host Configuration Protocol (DHCP),

v Simple Mail Transfer Protocol (SMTP),

vi Transmission Control Protocol (TCP)

3.2.2 Software Product Features:

New User Registration:

i This is an essential feature.

ii Each user who is new to the application has to go through the process of registration.

iii If the user clicks on Register button, a registration form will be displayed to the user wherein the user has to provide the required details.

iv These details include the full name, the email address, and a password necessary to create the account.

v The validation of all these fields take places and a message is displayed if any invalid input is provided.

vi Once a user is registered, the credentials are veriffied through OTP and then he can directly sign in the next time he wishes to use the application.

Accessibility:

i This is an essential feature as no illegitimate user can access without authority.

ii The user's side will be permitted limited access like viewing the history of the books read/ reading progress, wish lists, favourites and cart. The user will not be permitted to modify any system data of the Project.

iii The Admin's side will be permitted to view and modify the system data of the Project like changes in the website adding extra features, increasing the books, etc.

Book Renting:

i This is an essential feature of the project.

ii The user has to choose the books he wants to buy or rent by adding them to the cart.

iii After this the total cost of the entirety is calculated and the user has to proceed to payment after checking the bill.

iv After the payment is completed the books are made available to the user for reading.

Payment:

i This is an essential feature.

ii The software system is integrated with the banking API for payments.

iii Multiple payment options are provided to the user.

iv The user can choose the appropriate payment mode and make the payment

v Options offered in the payment gateway are namely credit cards, debit cards, net-banking, mobile wallets.

Feedback:

i This is a desirable feature.

ii After the user makes the payment successfully, the invoice is received and then feedback is prompted to the user about the Ride.

iii The user's feedback's are valuable as they help to fix bugs, correct problems, and help in the improvement of services.

iv The user should provide additional requirements for improving the service.

Security:

i This is an essential feature.

ii The login credentials of the users are securely stored in a database.

iii The payment is done using the banking API through the payment gateways securely.

History of Books :

i This is an desirable feature.

ii The user is able to check his/her previous readings and also track his progress.

Profile:

i This is an important feature.

ii Profile Page contains the personal details of the customer for an individual account creation and identification through the database along with the favourites, wish listed books and the books in the cart or already bought.

3.2.3 Software System Attributes:

Reliability:

The user should have a strong internet connection whenever they want to use the application as the application is completely reliable on data provided by the server.

Availability:

The application shall be available at all times, with the exception being software updates (if any). In case of any system breakout the system will restart from the Home state and there will be no loss of data or data breakout from the system.

Security:

All the user data should be encrypted and then stored in the databases. In case of a security breach in the application, the user data will then have an additional layer of security. All the login activity of the users will be stored in a log, along with, the time of log in, the session time and other important details.

Maintainability:

The data from the user will be updated in database whenever there is any changes. The user interface of the web app will be aimed at making it more user friendly as per customer requirements. The approach will be aimed to maintain and make the web app interface user friendly and intuitive. As per user feedback the changes will be made accordingly in the web app interface. The prices of books will be updated according to the demand and availability.

Portability:

The application is easily portable on most machines. Most of the application code is written in dart programming language (Framework Flutter) which will be using object-oriented concepts which is a proven portable language.

Performance:

The application should be able to handle at least 5 simultaneous users, without any lag in performance. The application should provide data storage of at least 1 GB to each user.

3.2.4 Database Requirements:

All the data will be stored in Firebase Real time Database and Firebase Cloud Firestore. It is NoSQL database which stores data in the form of JSON, i.e., in key-value pair.
Since, Data in Real time Database is structured as a single tree, there are no collections more than 1.
This may include the following:

Types of information used by various functions:

i The Registration function will take user information such as full name, email-id through which an OTP is shared to authenticate required details.

ii The payment module will use the bank account credentials of the user and generate the bill.

Accessing capabilities:

i Only the Database administrators can access the database and make changes if required.

ii Other users can access the data from the application as per their authorization.

Integrity constraints:

i User passwords will be stored using hash functions.

ii Email id of each user is unique.

iii Database is accessed only by authorized users.

3.3 User Interfaces Images:

Figure 3.1: Splash Screen

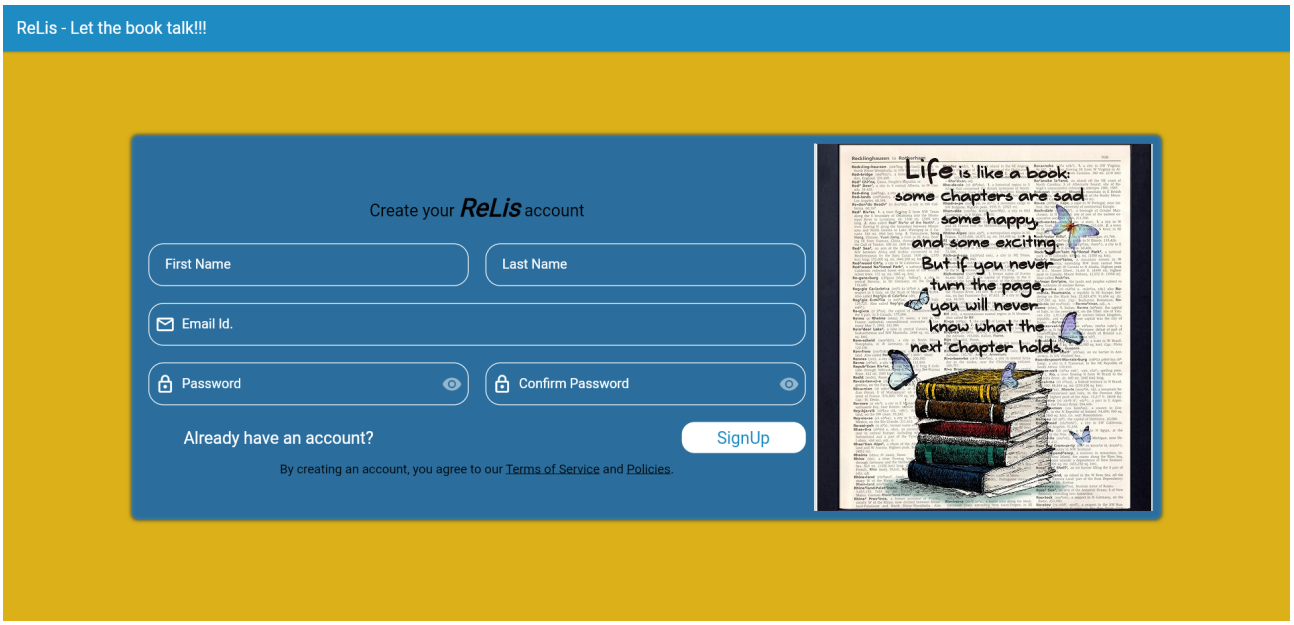


Figure 3.2: Sign-Up Page

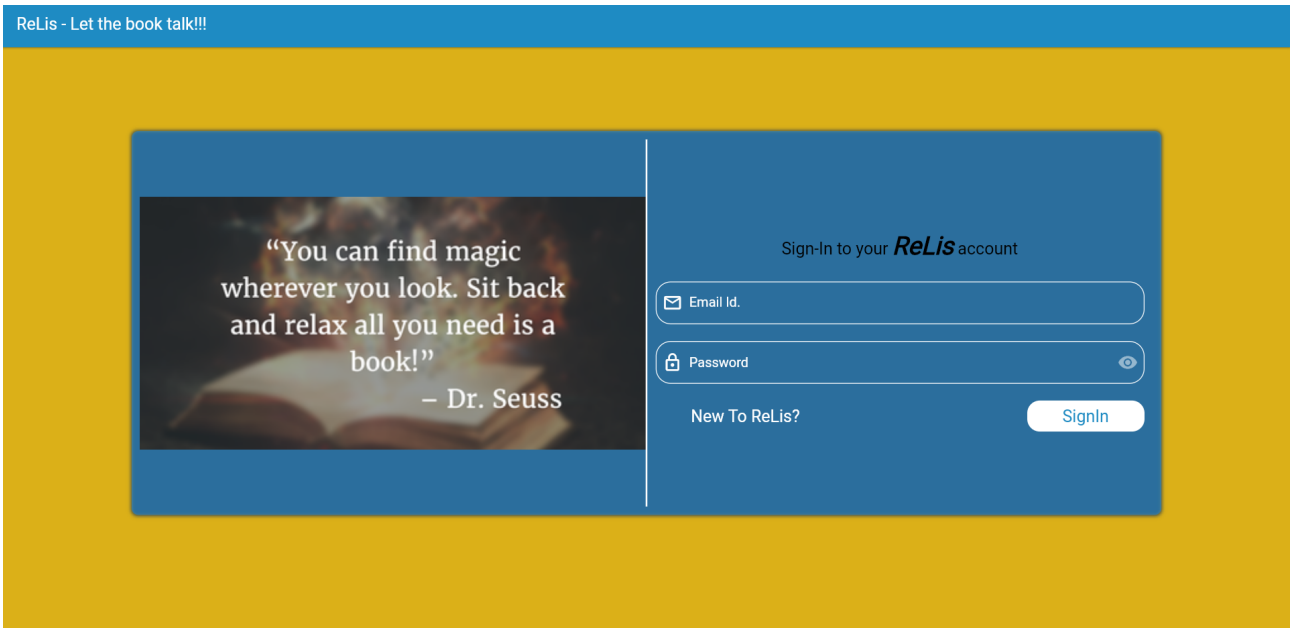


Figure 3.3: Sign-In Page

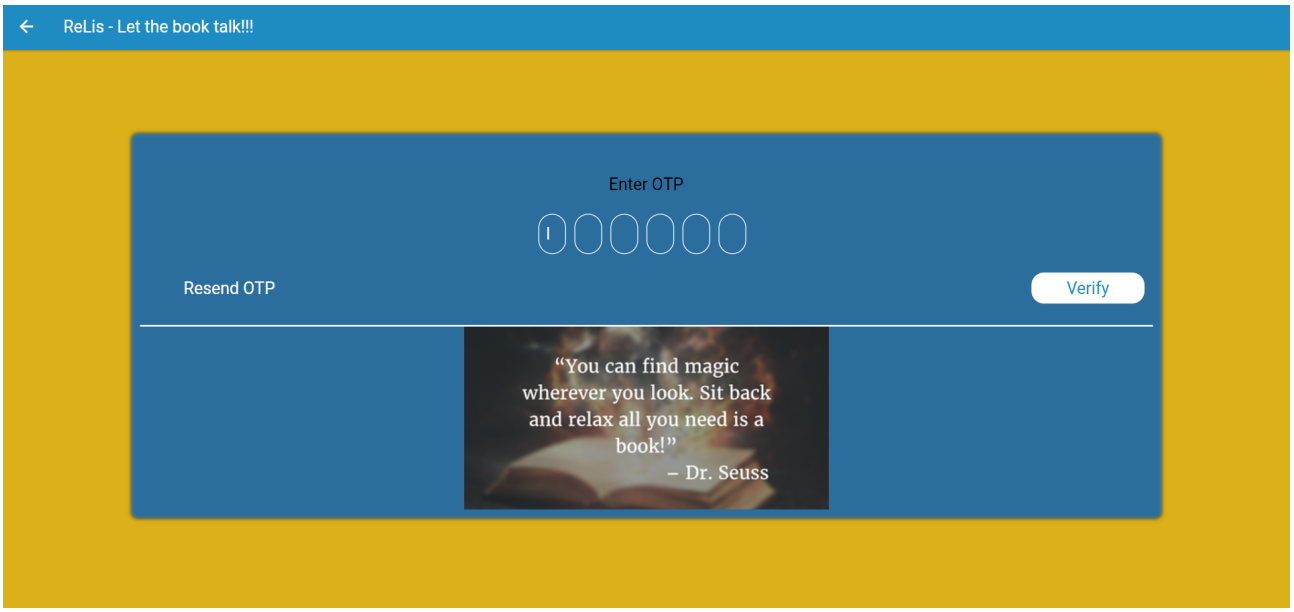


Figure 3.4: OTP Page

Figure 3.5: Home Page

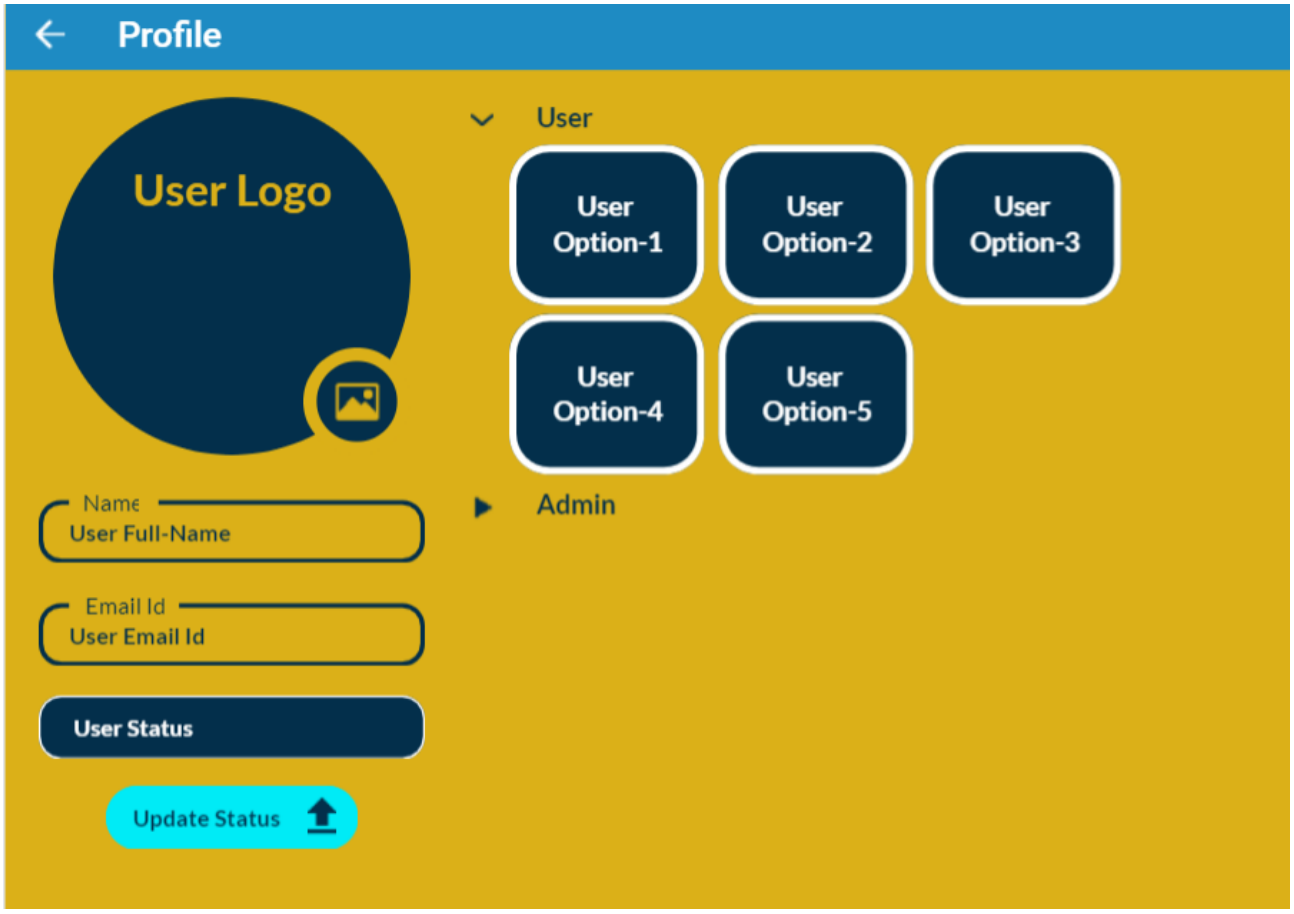


Figure 3.6: User Profile Page

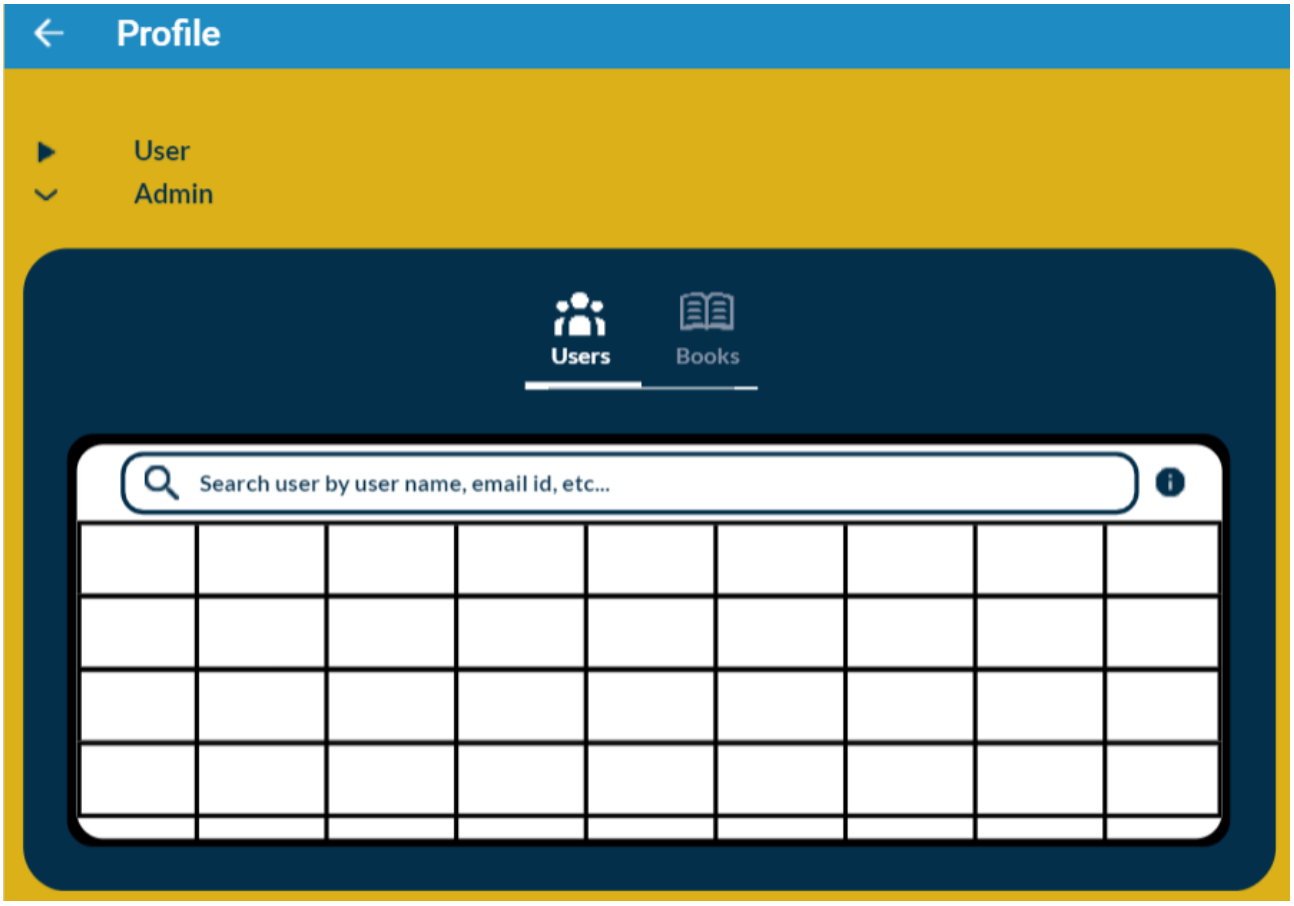


Figure 3.7: Admin Profile Page

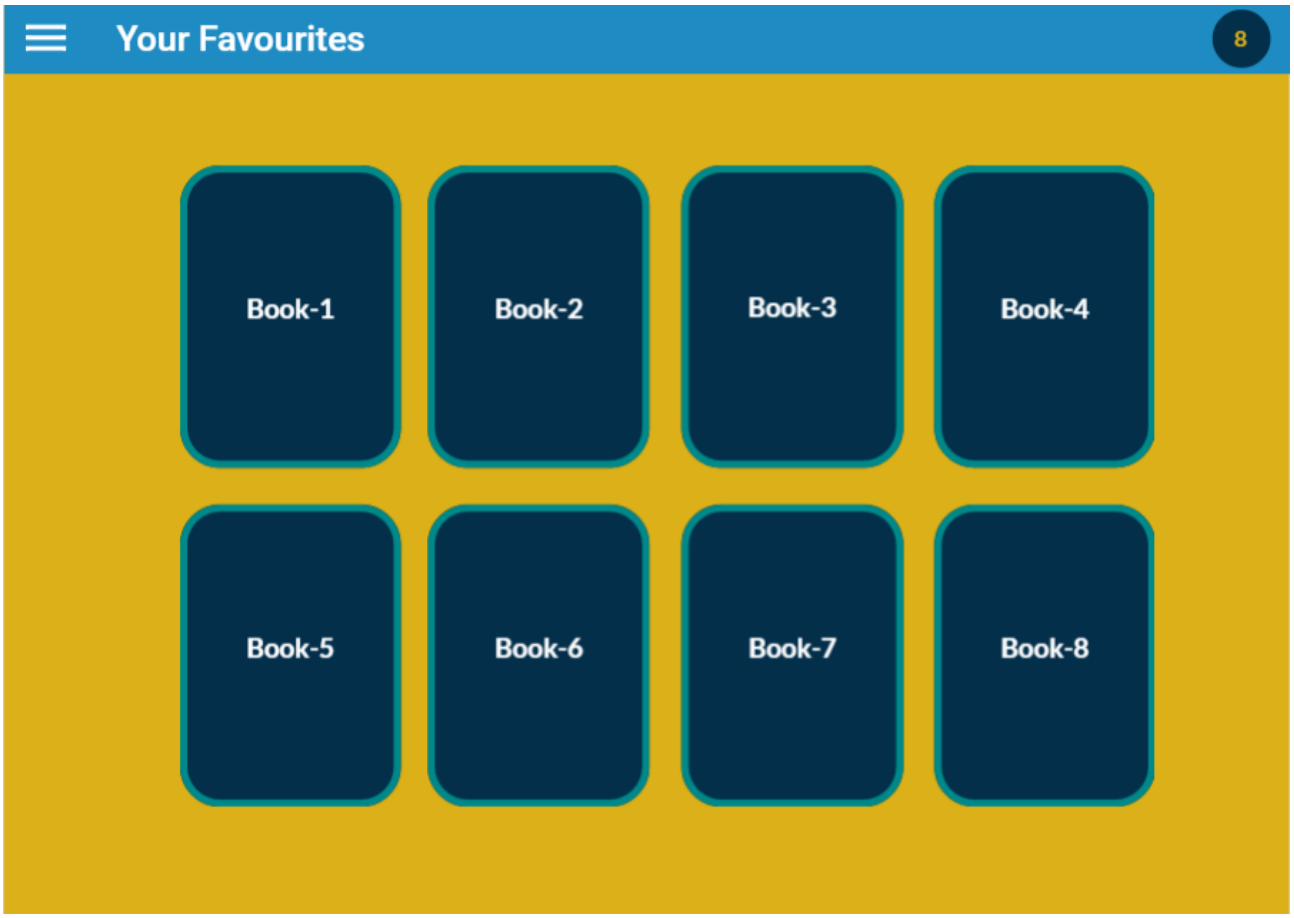


Figure 3.8: Favourites Page

Figure 3.9: Wish-List Page

Figure 3.10: Genre Page

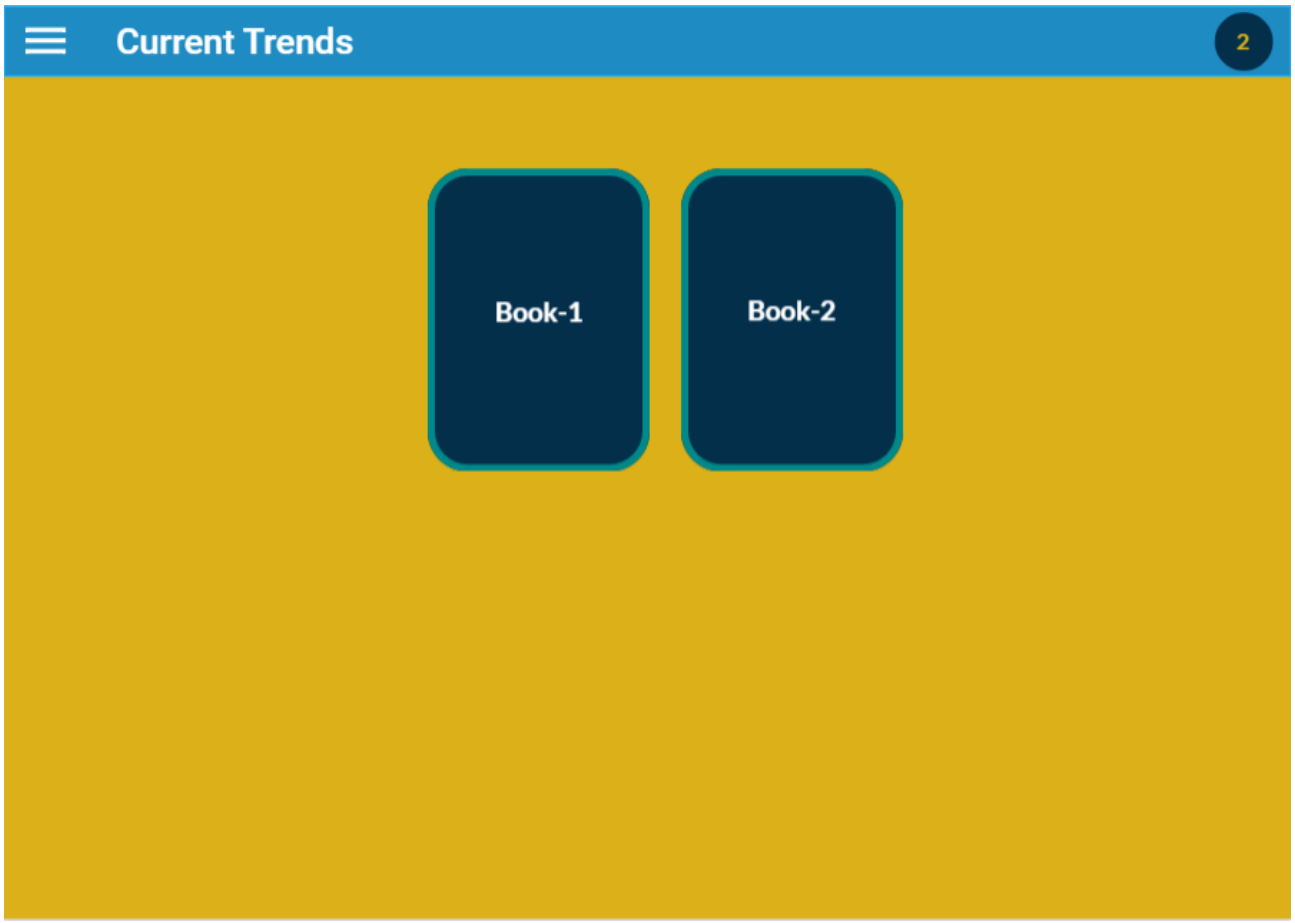


Figure 3.11: Trending Books Page

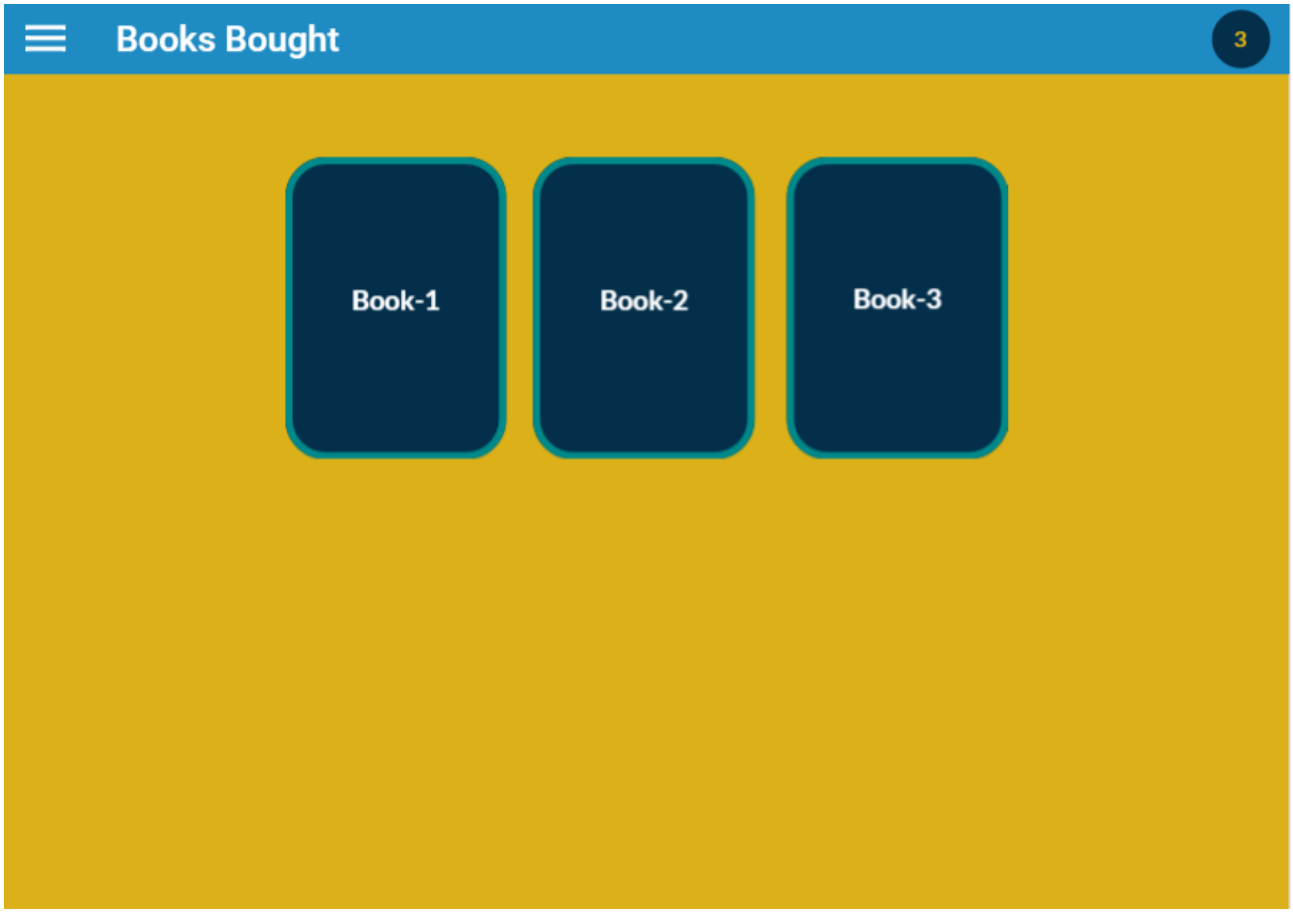


Figure 3.12: Books Bought Page

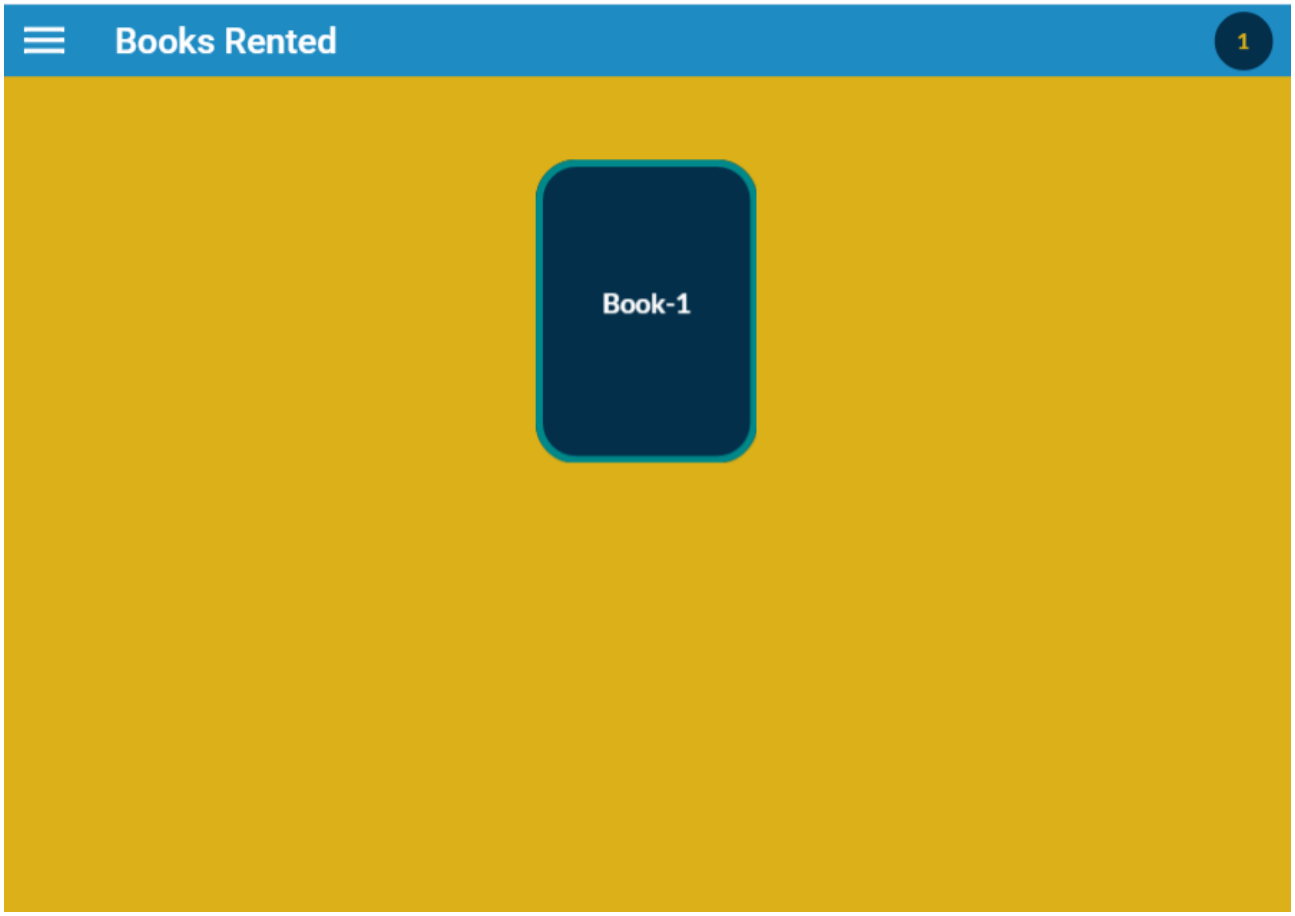


Figure 3.13: Books Rented Page

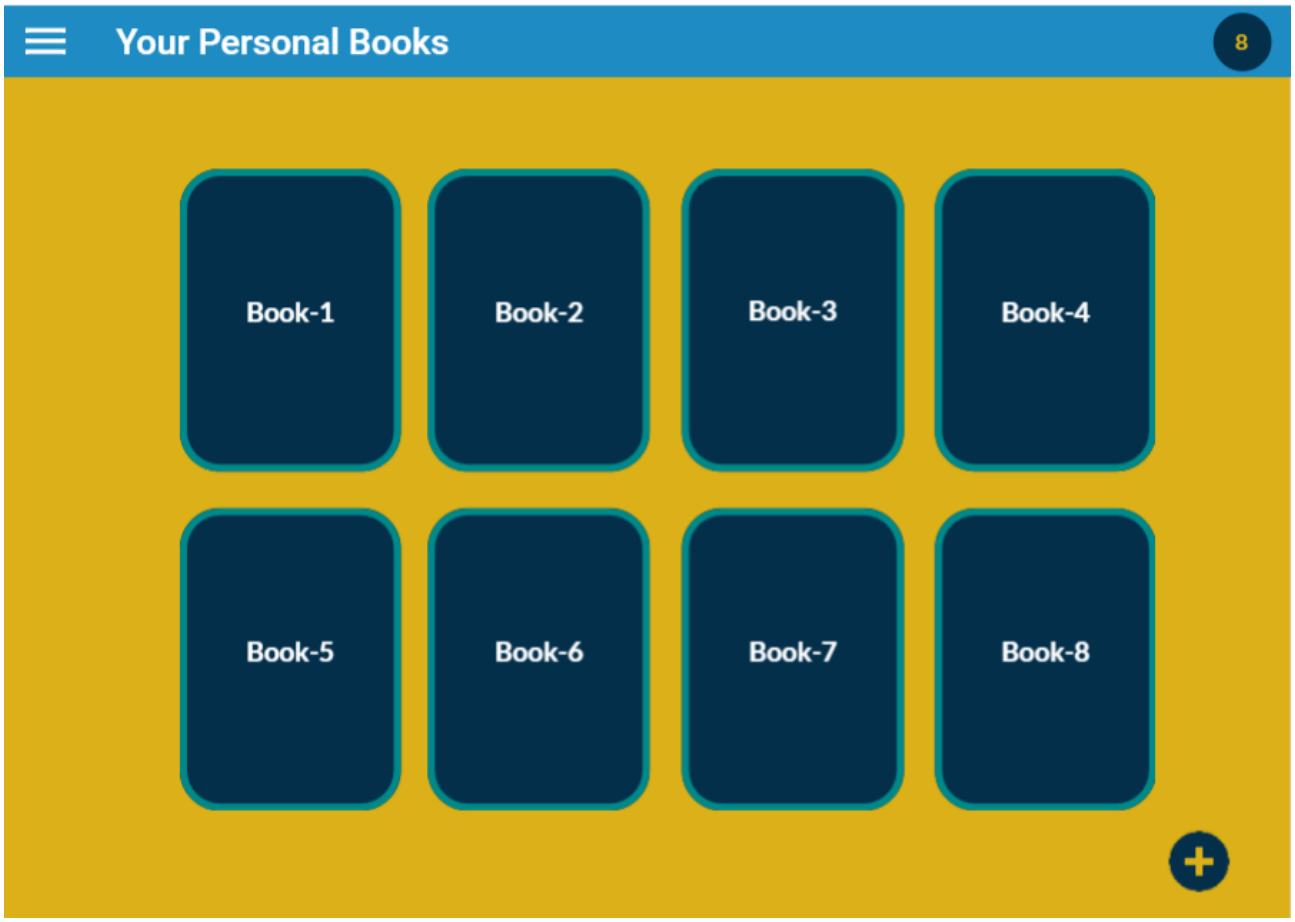


Figure 3.14: Personal Books Page

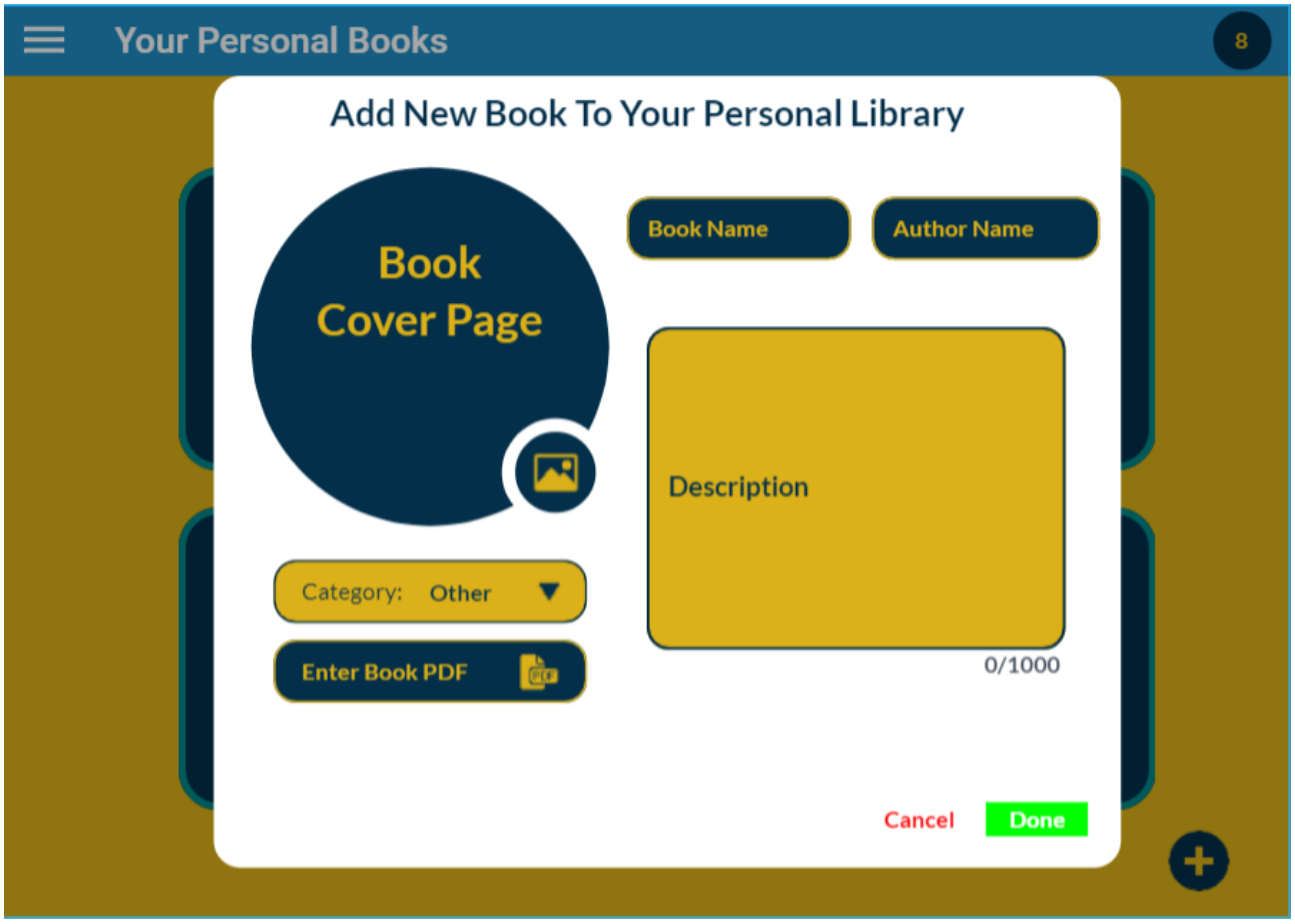


Figure 3.15: Add Book in Personal Books

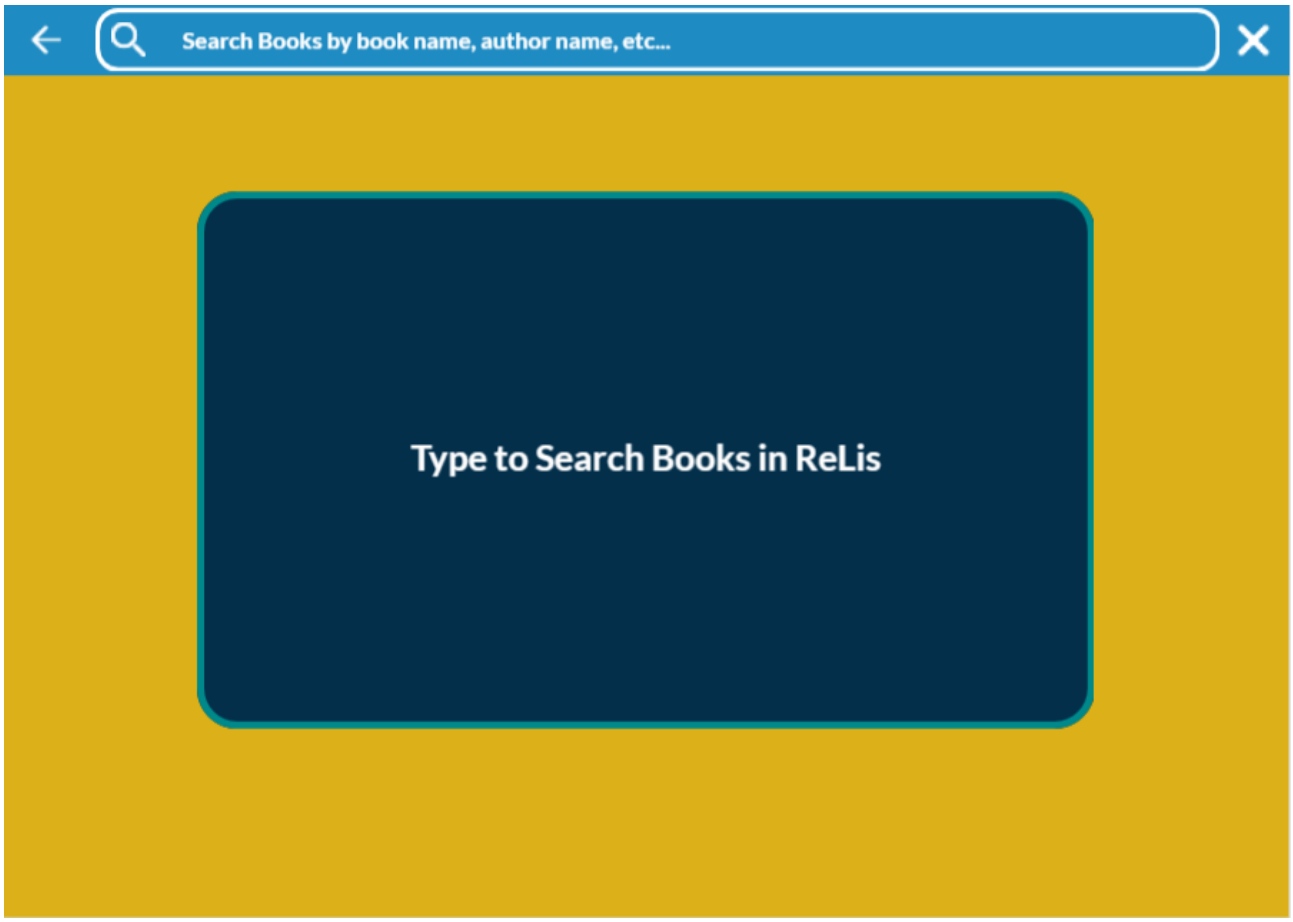


Figure 3.16: Search page

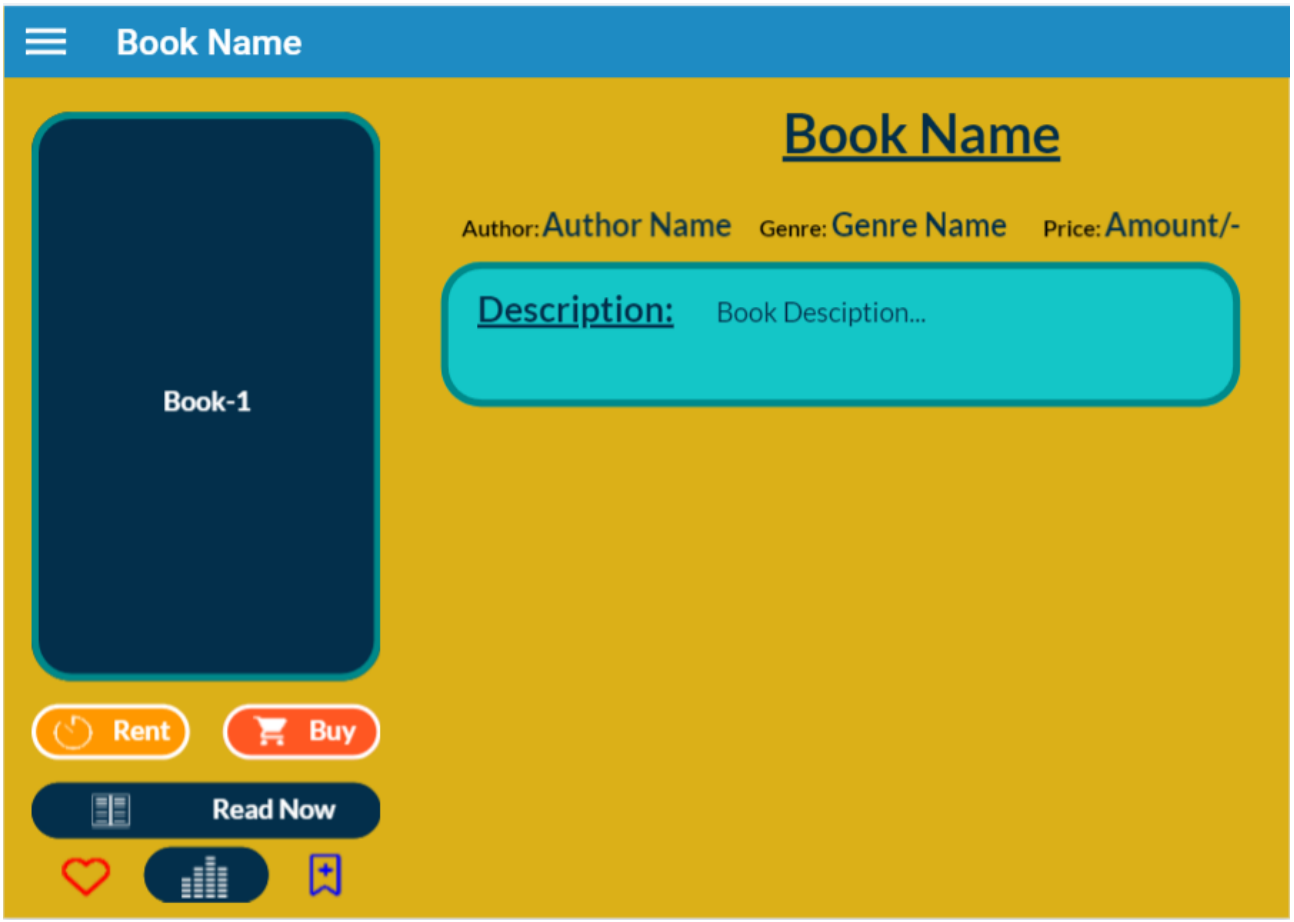


Figure 3.17: Book Preview Page

Chapter 4

SOFTWARE DESIGN DOCUMENT - SDD

4.1 INTRODUCTION

4.1.1 Design Overview

Libraries are used to store books, but we need a system to navigate to a specific book effortlessly. Moreover, it is not possible for all book lovers to visit the library and buy or rent books, so the smart E-Library system helps to solve this issue. ReLis is a smart e-library system which is Eco-friendly as there is no involvement of hard copies and thus, it saves paper. Also, based on the current scenario, screen time for most people has increased drastically which causes negative effects on our body and thus the feature of audio books is beneficial to users.

This web app is designed to be run on any device by a user for buying or renting an book or audio book after choosing from the available book on the Home page or searching the book through book id, name , author's name or barcode from the web app. The web app will require a working web browser in the device to work. The interfaces will be user friendly and everything will be done digitally. This app will be made user friendly as it will be used by people in age group 14-65.

4.1.2 Requirements Traceability Matrix

	User	Admin
New User Registration	✔	
Login and Logout	✔	✔
Read a Book	✔	✔
Profile Management	✔	✔
Add a Book		✔
Publish a Book	✔	✔
Book Preview	✔	✔
PDF Reader	✔	✔
Audio Book	✔	✔
Payment	✔	
Feedback	✔	✔
History of Book	✔	✔

4.2 SYSTEM ARCHITECTURAL DESIGN

4.2.1 Chosen System Architecture

We have selected MVC architecture for my project. MVC Stands for " Model-View-Controller" . We selected this because it matches with the characteristics of our project. i.e. one user can work on the view which is the front end while the other users can work on the backend to create the main functionality and compartmentalized the app. MVC mode is used to create rapid and parallel development. MVC is an application design model comprised of three interconnected parts. They include the model (data), the view (user interface), and the controller (processes that handle input).

Model
A model is data used by a program. This may be a database, file, or a simple object, such as an icon or a character. Here model files:
HomePage.dart
SignInPage.dart
SignUpPage.dart
OTPPage.dart
Dashboard.dart
Bookpreview.dart
BillPage.dart
ProfilePage.dart SearchBook.dart
Retrieves/Updates/Stores data from firebase file.

View
A view is the means of displaying objects within an application. Examples include display-

ing a window or buttons or text within a window. It includes anything that the user can see. Here view Files:
ViewCustomer.dart
ViewBook.dart
Favourites.dart
WishList.dart
Cart.dart
BookHistory.dart
Statistics.dart

Controller
A controller updates both models and views. It accepts input and performs the corresponding update. The three parts of MVC are interconnected (see diagram). The view displays the model for the user. The controller accepts user input and updates the model and view accordingly. Here Controller files:
addCustomer.dart
addBook.dart
RemoveCustomer.dart
RemoveBook.dart
PublishBook.dart
Contains all input and logic files.

4.2.2 Discussion of Alternative Designs

The alternative options were Layered Architecture, Data-flow architecture, Client Server Architecture and Data-centered architecture. Layered architecture is used at system level. It's for any type of editors like Notepad, etc. Data-flow architecture is for android based projects. We are making use of data-flow architecture as our project will be mainly deployed on android level.
Here, Our System is using Client-Server architecture but inside MVC Architecture, using individually client-server architecture will not accomplish required features of system. It will efficiently work with backend of the system, so it would be better if we merge two architectures to work efficiently with frontend and backend of the system.
MVC is dividing our System into three parts and it keeps data (from database), User interactivity (UI), Processing input(controller) apart from each other therefore it is triangular architectural whereas in client-server architecture we can't differentiate between these 3 actions. It deals with request and response, client sends request to server and server fulfil request by providing dynamic/static pages, and it is linear architecture.
ReLis is Customer centric application that is, it is all about user's interactivity with the app and that allows users to rent or buy a book or audio book from anywhere in the world so that he/she can read the book with any burden of carrying the book and also saving paper at their desired location.

4.2.3 System Interface Description

- The user interface for the system will allow the user to buy or rent books and audio books from their homes easily for a fair price through a payment option provided by the developer successfully.
- The user should be presented with all main functions on the first user interface page to allow for the user to select the function to use without the need to navigate inward to find it.
- The interface will need to use API to figure out the searching through barcode functionality and the payment gateway.And complete the shopping of books by performing the payment and providing the feedback for their experience.
- It will be accessible through a web app interface to allow for centralized hosting and use by various types of customers of different age groups.
- The admin side is merged in the same website with additional features provided to the admin.
- The admin side is responsible for the maintenance of records which includes adding, editing, or deleting any user or book information as well as the approval of any new books which are meant to be published on the interface.
- Another function is the adding, deleting of books and its categories by providing and managing the necessary details.
- Finally checking the history of all the bought ebooks or audio books and payment verification and look into the rating feedback is the most important function performed on the admin side.

4.3 DETAILED DESCRIPTION OF COMPONENTS

Figure 4.1: ER Diagram

The description of components is given below:

1. **Authentication System**
   This entity has attributes email id / mobile number, password. This entity is related to all the other entities as shown in the diagram. The attribute email-id / phone-no is a primary key.

2. **Person**
   The user entity has several attributes such as name, email id / mobile number, user status, favourite book list, wished book list. The user entity is related to all the other entities as shown in the diagram. The attribute email-id / phone-no is a primary key.

3. **Categories**
   The category entity has attributes such as categoryName, categoryId. The category entity is related to all the other entities as shown in the diagram. The attribute category Id is a primary key.

4. **Books**
   The book entity has several attributes such as bookId, bookName, authorName, categoryId, authorName, price, BookData, BookCoverImage, fav-Book-User-List, Wished-Book-User-List. The book entity is related to all the other entities as shown in the diagram. The attribute bookId is a primary key.

4.3.1 Component-1: Authentication Module

The module requires a user to log in with valid credentials. The user has to enter the valid email-id and password. It has to be reliable and secured.

4.3.2 Component-2: Admin

This component requires the user to login with his valid admin credentials. The profile can also be seen from the personal and professional details of the admin. Their responsibility is to view and maintain the user (reader and admin) lists, add remove or edit the book and their categories, add or remove or edit user, and maintain the history for the books purchased / rented.

4.3.3 Component-3: Reader

A new reader always has to register himself by providing the required details via a form and later login to access the service provided. The user can read a book, provide feedback for the same. Also the reader can mark a book as favourite and also can add a book in wish-list. A user profile is also created. It should always be a secure usage for the reader.

4.3.4 Component-4: Purchase Module

The module requires a user to be logged in at all times. He/She has to choose the book and click on the desirable option - Buy or Rent. And Later has to complete the transaction process. It has to be reliable and secured.

4.4 USER INTERFACE DESIGN

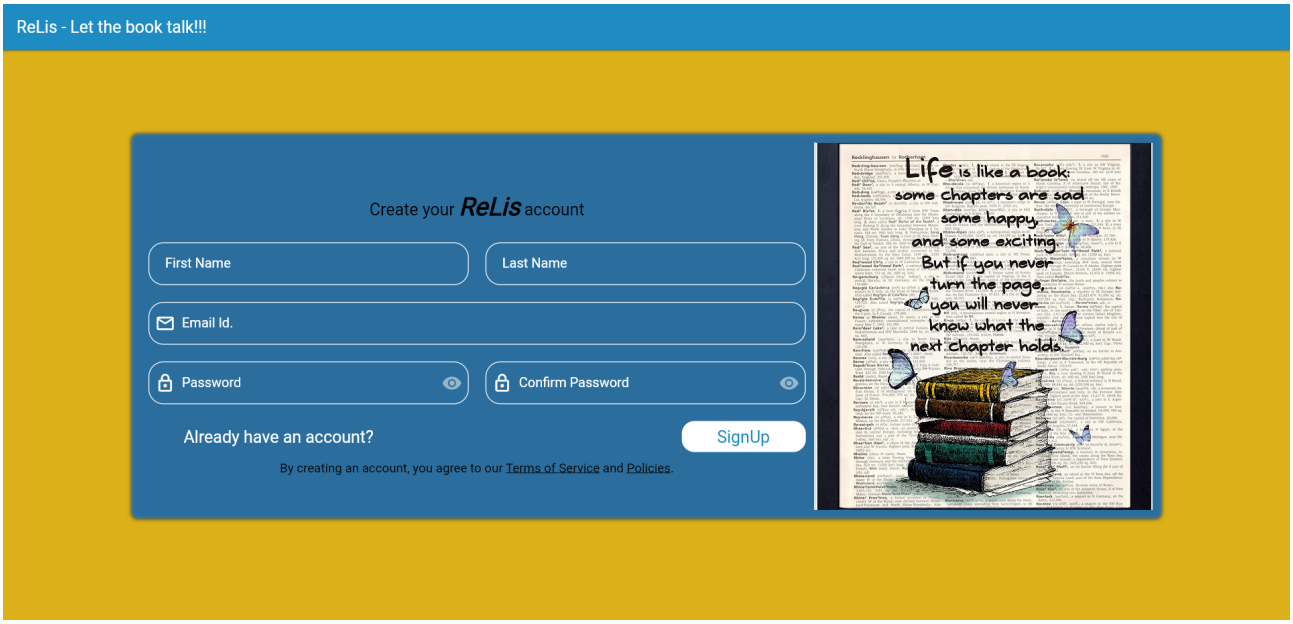
4.4.1 Description of the User Interface

Screen Images

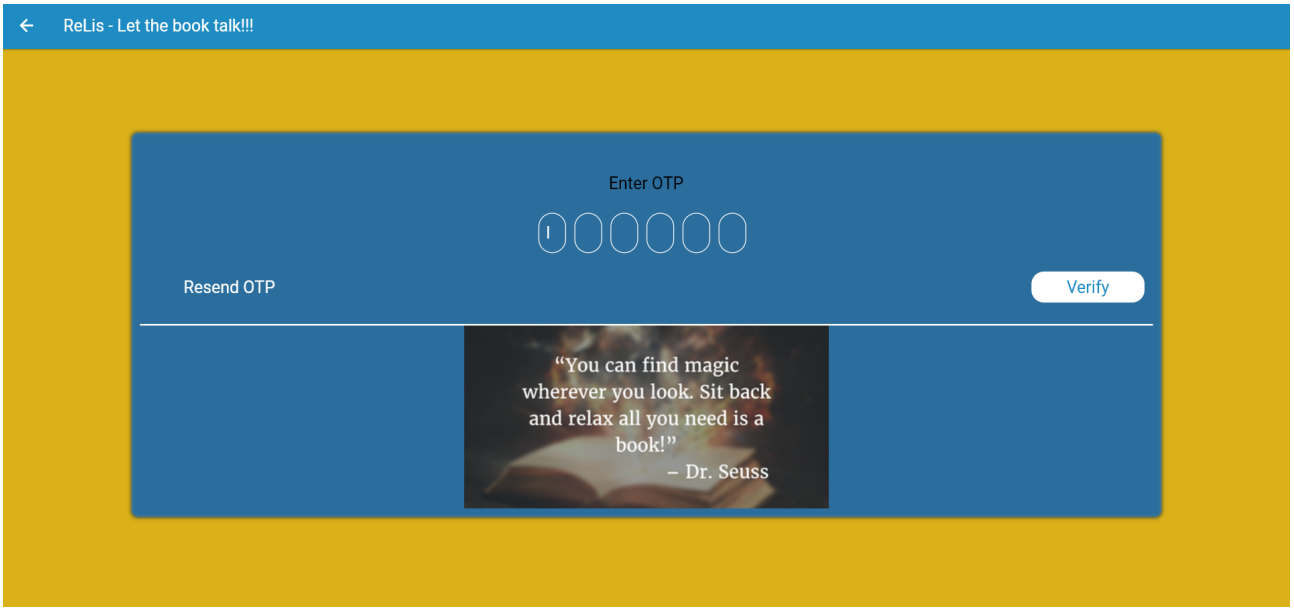
1. **Splash Screen:**

2. **Login Module:**

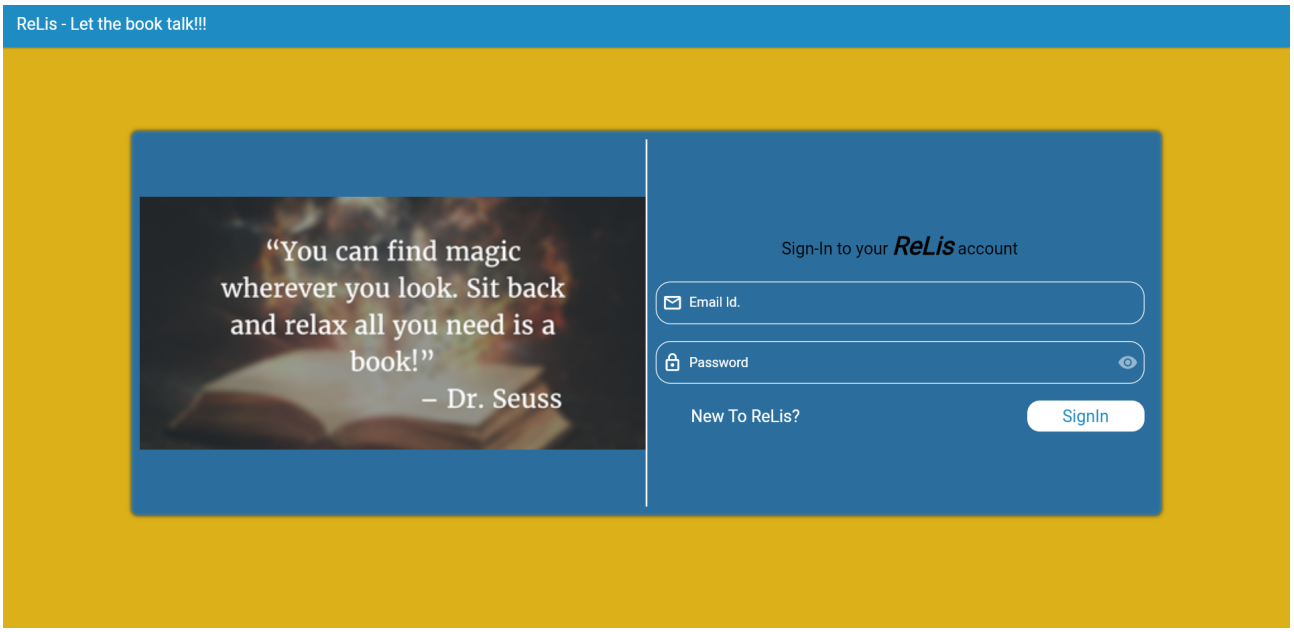
   (a) **Sign Up Page:**



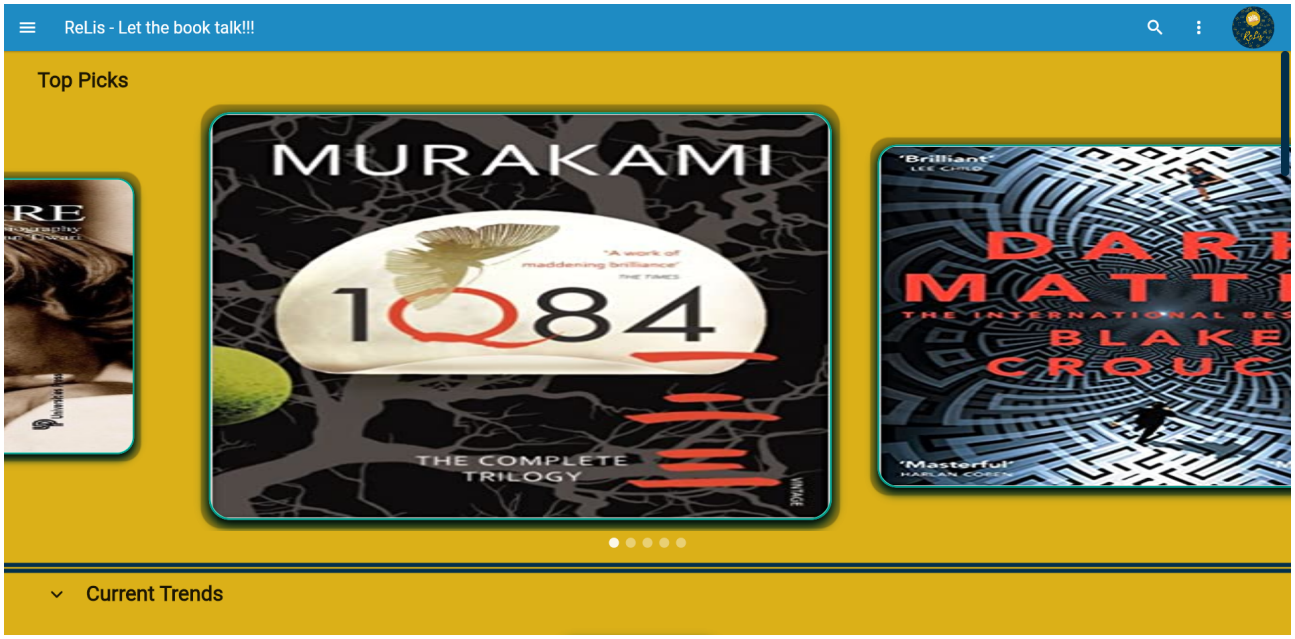
(b) **OTP Page:**



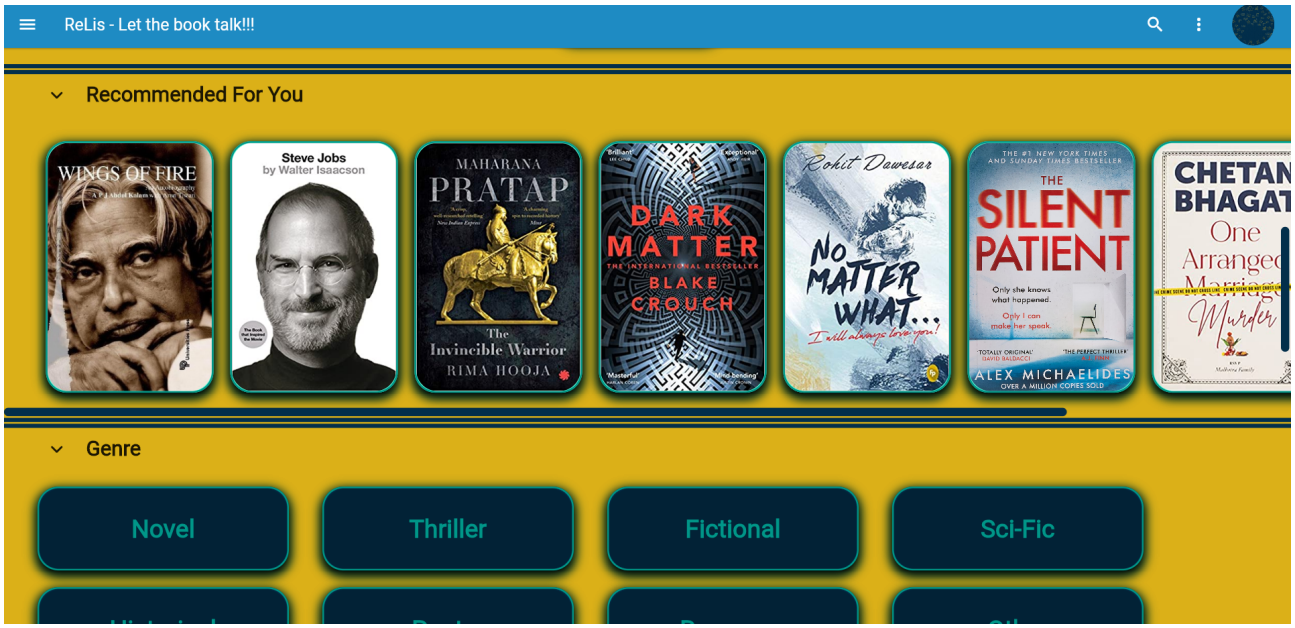
(c) **Sign In Page:**



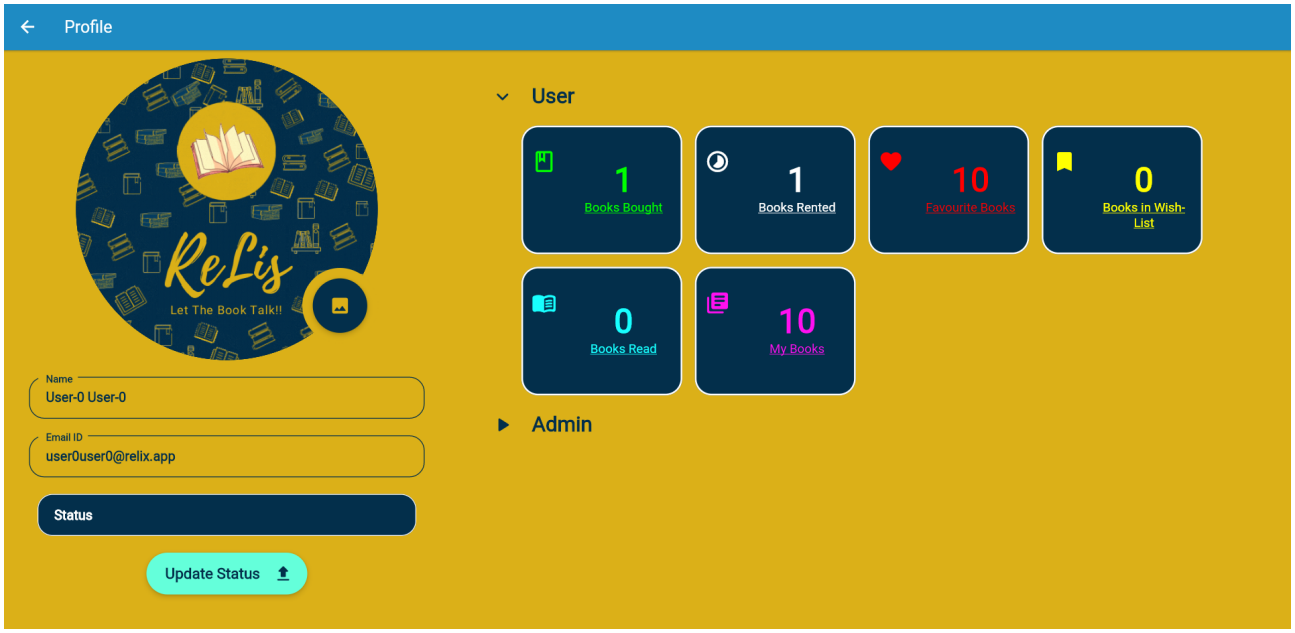
3. **Home Page:**


ReLis - Let the book talk!!!
Top Picks
MURAKAMI
1Q84
THE COMPLETE TRILOGY
'Brilliant'
'Masterful'
Current Trends

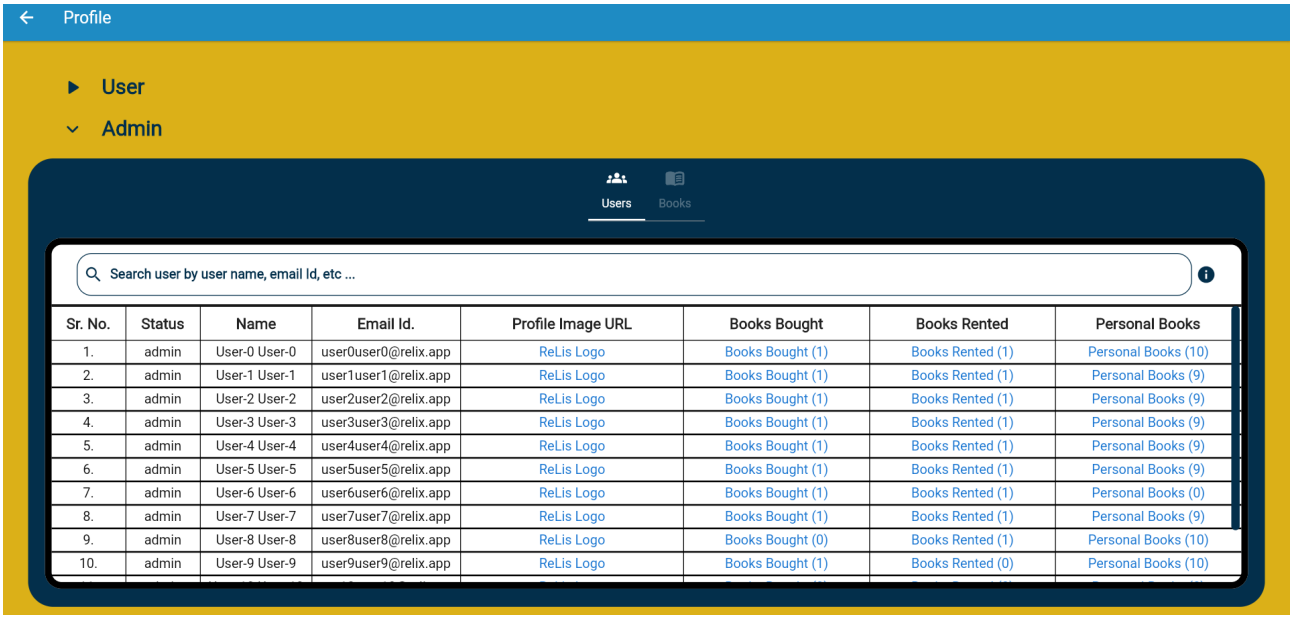

ReLis - Let the book talk!!!
Current Trends
MAHARANA
PRATAP
The Invincible Warrior
RIMA HOOJA
Recommended For You
WINGS OF FIRE
Steve Jobs
by Walter Isaacson
MAHARANA
PRATAP
DARK
MATTER
Rohit Dawesar
NO MATTER
THE
SILENT
PATIENT
CHETAN
BHAGAT
One
Arranged



4. **User Profile:**



5. **Admin Profile:** Consisting of various tabs.



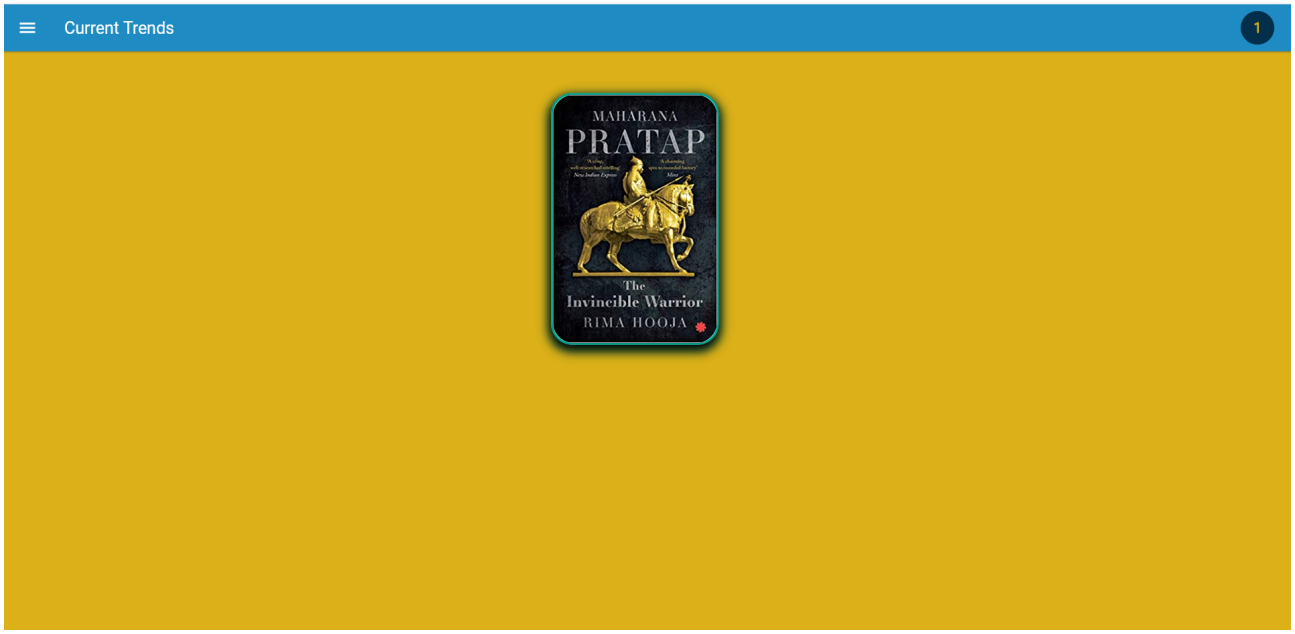
Sr. No.	Status	Name	Email Id.	Profile Image URL	Books Bought	Books Rented	Personal Books
1.	admin	User-0 User-0	user0user0@relix.app	ReLis Logo	Books Bought (1)	Books Rented (1)	Personal Books (10)
2.	admin	User-1 User-1	user1user1@relix.app	ReLis Logo	Books Bought (1)	Books Rented (1)	Personal Books (9)
3.	admin	User-2 User-2	user2user2@relix.app	ReLis Logo	Books Bought (1)	Books Rented (1)	Personal Books (9)
4.	admin	User-3 User-3	user3user3@relix.app	ReLis Logo	Books Bought (1)	Books Rented (1)	Personal Books (9)
5.	admin	User-4 User-4	user4user4@relix.app	ReLis Logo	Books Bought (1)	Books Rented (1)	Personal Books (9)
6.	admin	User-5 User-5	user5user5@relix.app	ReLis Logo	Books Bought (1)	Books Rented (1)	Personal Books (9)
7.	admin	User-6 User-6	user6user6@relix.app	ReLis Logo	Books Bought (1)	Books Rented (1)	Personal Books (0)
8.	admin	User-7 User-7	user7user7@relix.app	ReLis Logo	Books Bought (1)	Books Rented (1)	Personal Books (9)
9.	admin	User-8 User-8	user8user8@relix.app	ReLis Logo	Books Bought (0)	Books Rented (1)	Personal Books (10)
10.	admin	User-9 User-9	user9user9@relix.app	ReLis Logo	Books Bought (1)	Books Rented (0)	Personal Books (10)

6. **Favourite Page:**

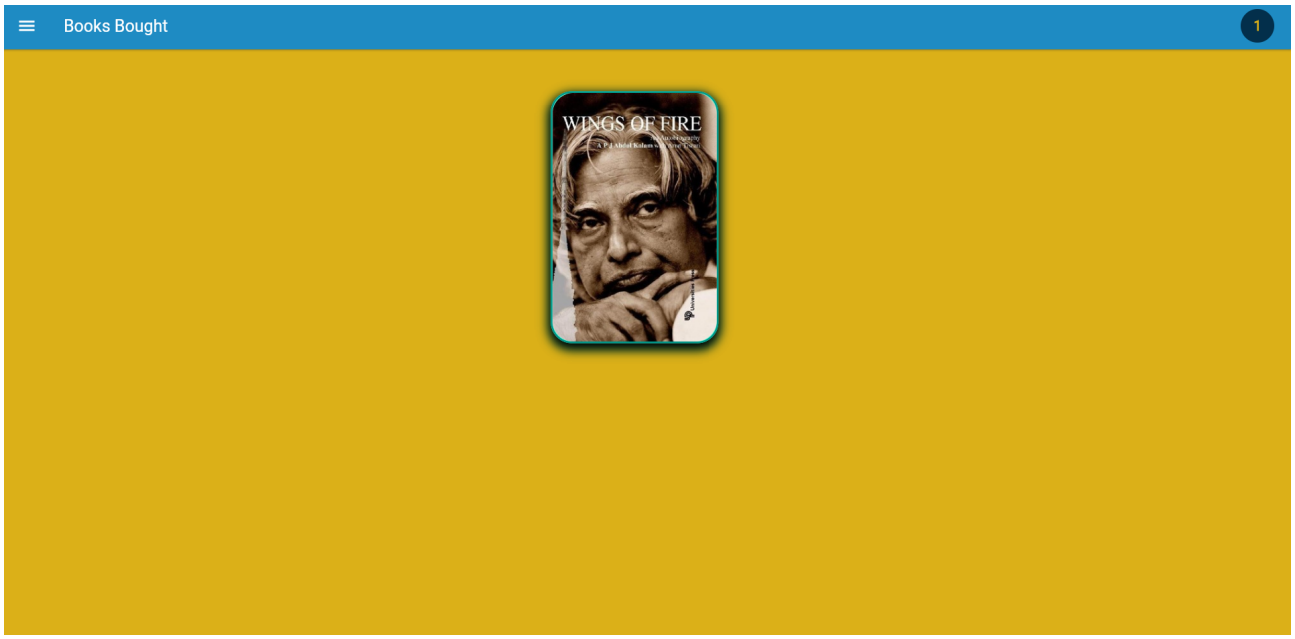
7. **Wish-List Page:**

8. **Genre Page:**

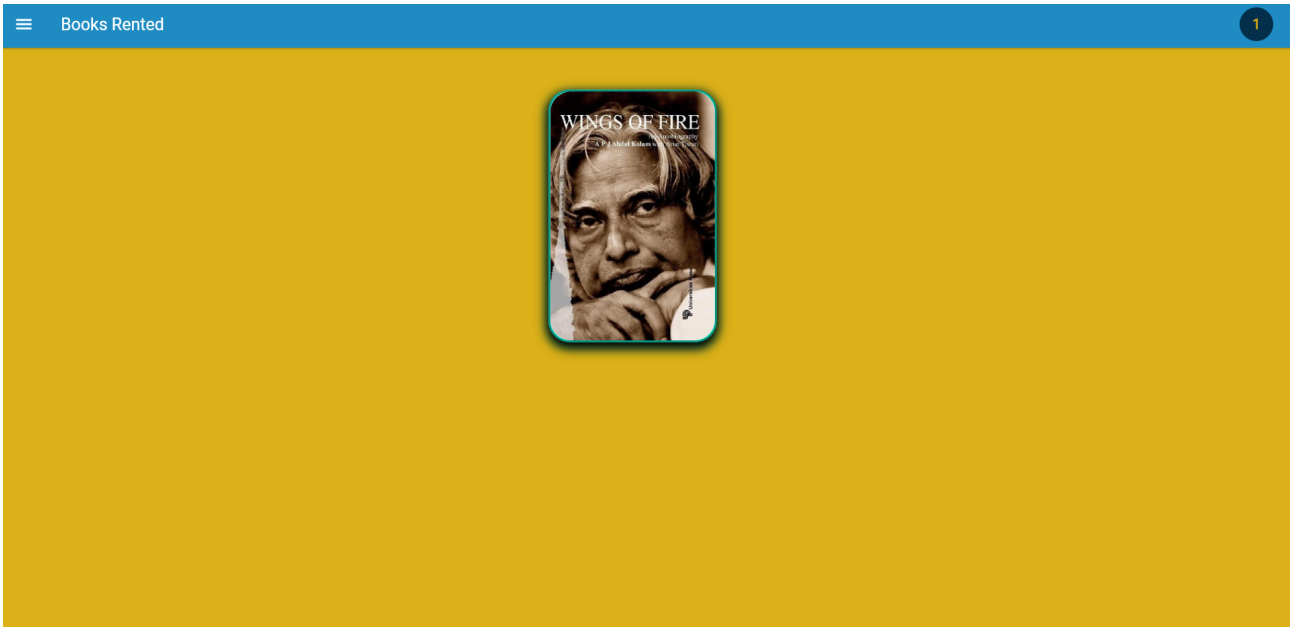
9. **Trending Page:**



10. **Books Bought Page:**



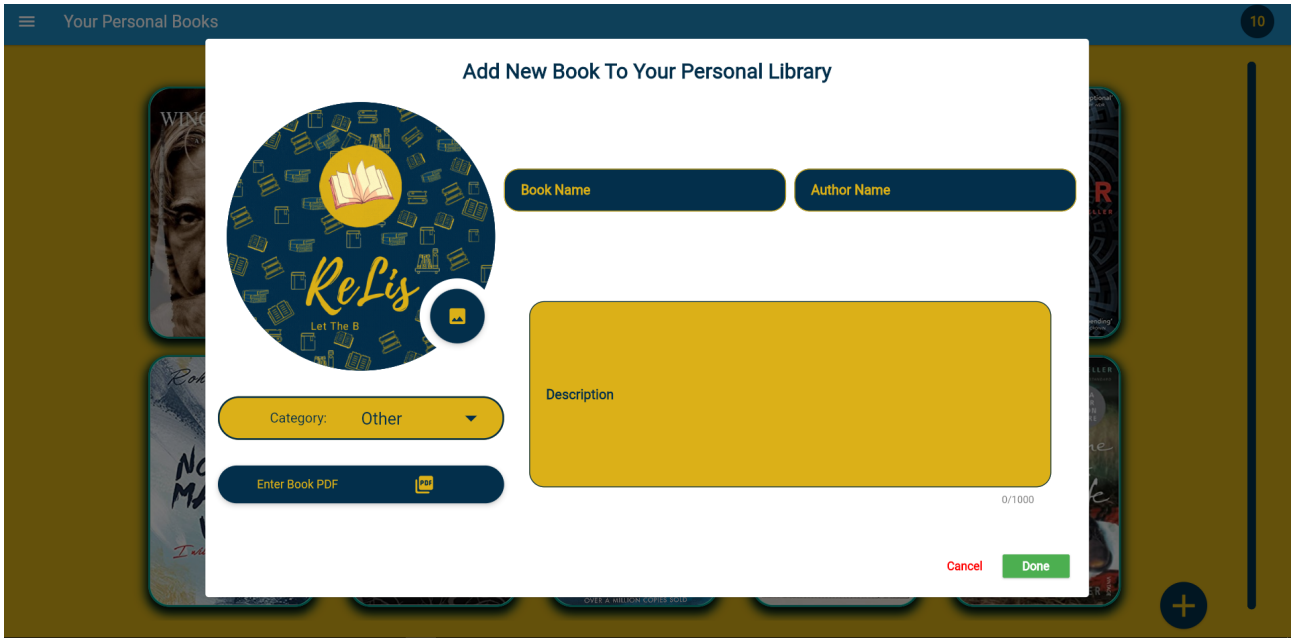
11. **Books Rented Page:**



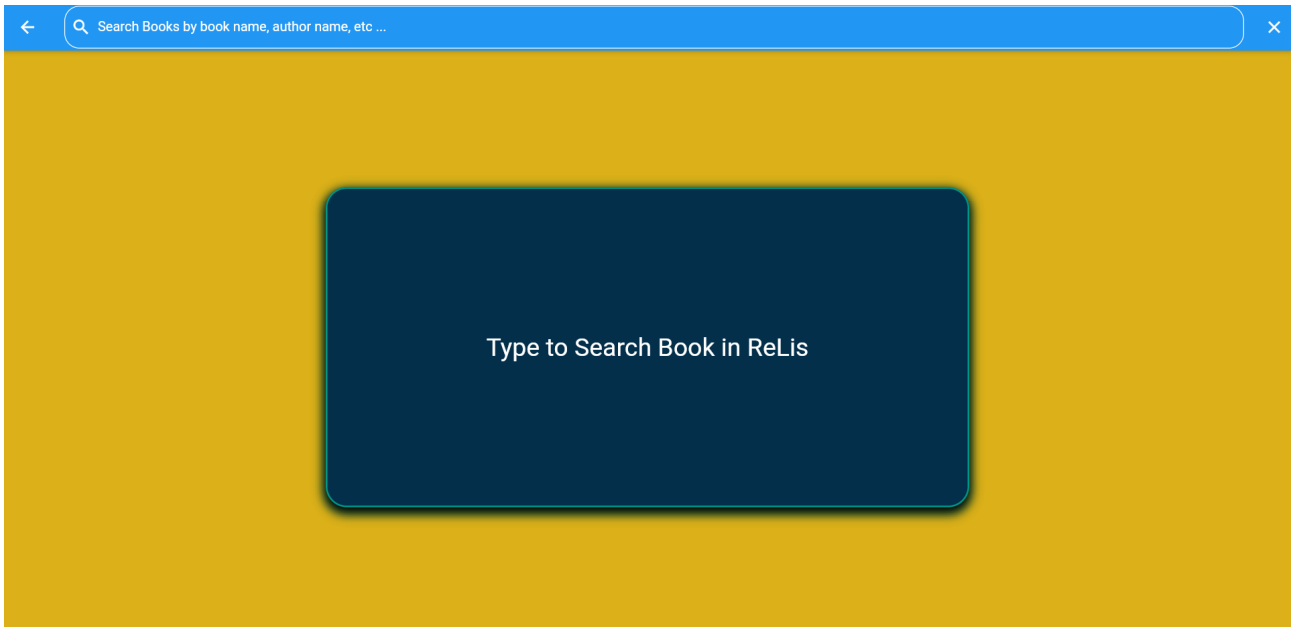
12. **Personal Books Page:**

    (a) **View Personal Books:**

(b) **Add new Book in Personal Books:**



13. **Search Book Page:**



14. **Book Preview Page:**

Objects and Actions

1. **Login Module:**

   (a) **Sign Up Page:** In this page, there will be fields to enter the full name, email-id and password followed by a Sign up button to register.Also, there would be option, that if anyone already has an account he/she can be redirected to the Sign-in page.

   (b) **OTP Page:** In this page, the user has to enter a one time password to verify his credentials. There are buttons to verify the OTP or to request the system to resend the OTP.

   (c) **Sign In Page:** In this page, there will be email-id and password followed by a sign-in button to login. Also, there would be option of forgot password or a button for a new user to be redirected to the Sign Up page.

2. **Home Page:** This page includes various components such as Top Picks, Recommended Books, History etc, which give glimpses of different books in the particular section.

3. **Admin Profile:** The admin profile page has different admin options (tabs). When a particular tab is clicked, the particular admin options is shown.

   (a) **Users Tab:** If Users tab is clicked, then we can see details of all users like user name, email id, user status, list of books bought/rented/wished/favoured by users if any and options to update, delete or block the user (user info.).

   (b) **Books Tab:** If Books tab is clicked, then we can see details of all books like book name, author name, book description, book cover image, book price, etc and options to update, delete or block the book (book info.).

4. **User Profile:** In this page, the user can view his details which they provided during registration. This page is crucial, since this gives the customer an option to view his data as well as upload any status and also check details such as books bought, rented, books added to the wish-list, cart or his favourites.

5. **Review New Book:** The review new book option displays list of books which are needed to be published. After verifying the book, admin can publish a book or can discard it.

6. **View/ Update/ Remove Users:** This tab consists records of users which can be edited / remove by the Admins. The record displays user's name, type, status, books bought / rented / favourited and wished, etc.

7. **Add/ View/ Update/ Remove Books:** This tab consists records of books which can be edited / remove by the Admins. The record displays book's name, author, genre, description, cover image, etc. Also, the admin can add new book in the database.

8. **Book Preview:** When the user clicks on book preview they will be navigated to a page where they will get detailed information about the book like the book name, the author, description, category, price, etc.

9. **Search Books:** When the user clicks on search books they will be navigated to a page where they will get a search bar where the user has to provide book id or name, etc to get detailed information about the book like the book name, the author, description, category, price, etc. There is another way that the user scans the barcode of hard copy of the book to get the details.

10. **Bill Feedback:** When the user clicks on proceed to pay after adding books to buy or rent in the cart the user will be redirected to a payment gateway using an API and after payment he has to provide a feedback about the book and the service..

11. **Wishlist Favourites:** When the user clicks on Wishlist or Favourites icon on the homepage below any book, they get stored on the respective pages for easy access of the user.

12. **History Reading Statistics:** When the user clicks on History Reading Statistics tab the user will be redirected to the page where he can see his previously read books, and their details such as the book name, the author, description, category, price, etc. and also be able to check their progress and time spent during reading.

4.5 System Architecture

Use Case ID:	**1a**		
Use Case Name:	**Sign Up**		
Created By:	Jainam Zobaliya	Last Updated By:	
Date Created:	17 September, 2021	Date Last Updated:	

Primary Actors:	Users
Secondary Actors:	-
Description:	The user can register to use the system by providing the correct details.
Trigger:	When the user comes to website provided user is not logged in or by clicking on New to ReLis on the Sign In Page.
Preconditions:	-
Postconditions:	-
Normal Flow:	Employee must have a stable internet connection
Alternative Flows:	-
Exceptions:	-
Includes:	-
Priority:	High
Frequency of Use:	Low
Business Rules:	-
Special Requirements:	-
Open Issues	-
Assumptions:	User must use their own and correct details.
Notes and Issues:	-

Use Case ID:	**1b**		
Use Case Name:	**OTP**		
Created By:	Jainam Zobaliya	Last Updated By:	
Date Created:	17 September, 2021	Date Last Updated:	

Primary Actors:	Users
Secondary Actors:	-
Description:	The users have to enter the otp sent to their email id / mobile no.
Trigger:	When the user clicks on SignUp on the Sign-Up Page.
Preconditions:	User needs to be enter correct details on the Sign-Up page.
Postconditions:	-
Normal Flow:	The user receives otp on the entered email id / mobile no, on clicking the Sign-Up button on Sign Up Page.
Alternative Flows:	-
Exceptions:	1. Server is down; 2. Database Connectivity Issue; 3. Network Unavailability.
Includes:	-
Priority:	High
Frequency of Use:	Low
Business Rules:	-
Special Requirements:	-
Open Issues	-
Assumptions:	User must use their own and correct email id / mobile no..
Notes and Issues:	-

Use Case ID:	**1c**		
Use Case Name:	**Login**		
Created By:	Jainam Zobaliya	Last Updated By:	
Date Created:	17 September, 2021	Date Last Updated:	

Primary Actors:	Users
Secondary Actors:	-
Description:	The user can log in to the system by providing the correct log in credentials.
Trigger:	When the user clicks on Already have an account on the Sign-Up Page.
Preconditions:	User needs to be Signed Up (registered) in the web applications.
Postconditions:	-
Normal Flow:	-
Alternative Flows:	-
Exceptions:	1. Server is down; 2. Database Connectivity Issue; 3. Network Unavailability.
Includes:	-
Priority:	High
Frequency of Use:	Medium
Business Rules:	-
Special Requirements:	-
Open Issues	-
Assumptions:	User must use their own and correct log in credentials.
Notes and Issues:	-

Use Case ID:	**2**		
Use Case Name:	**Admin Profile**		
Created By:	Jainam Zobaliya	Last Updated By:	
Date Created:	18 September, 2021	Date Last Updated:	

Primary Actors:	Admins
Secondary Actors:	-
Description:	The admin can use the admin rights on this page. By clicking on appropriate tab, the admin functions can be accessed.
Trigger:	When the admin clicks on Admin on the Profile Page.
Preconditions:	User needs to be logged in by using admin credentials in the web applications.
Postconditions:	-
Normal Flow:	-
Alternative Flows:	-
Exceptions:	1. Server is down; 2. Database Connectivity Issue; 3. Network Unavailability. 4. The data is not in prescribed format.
Includes:	-
Priority:	Medium
Frequency of Use:	Medium
Business Rules:	-
Special Requirements:	-
Open Issues	-
Assumptions:	User must use their own and correct log in credentials.
Notes and Issues:	-

Use Case ID:	**3**		
Use Case Name:	**Book Review**		
Created By:	Jainam Zobaliya	Last Updated By:	
Date Created:	21 September, 2021	Date Last Updated:	

Primary Actors:	Admins
Secondary Actors:	-
Description:	The admins can review the books, if any user wants to publish their books on the webapp.
Trigger:	When the admin clicks on Book Review on the Profile Page under Admin section.
Preconditions:	User needs to be logged in by using admin credentials in the web applications.
Postconditions:	The book is removed from the current section after choosing any of the option - Publish or Discard.
Normal Flow:	-
Alternative Flows:	-
Exceptions:	1. Server is down; 2. Database Connectivity Issue; 3. Network Unavailability. 4. The data is not in prescribed format.
Includes:	-
Priority:	Medium
Frequency of Use:	Low
Business Rules:	The books to be published must not be a copy of other author's work.
Special Requirements:	-
Open Issues	-
Assumptions:	User must use their own and correct log in credentials.
Notes and Issues:	-

Use Case ID:	**4**		
Use Case Name:	**View/Update/Remove User**		
Created By:	Jainam Zobaliya	Last Updated By:	
Date Created:	26 September, 2021	Date Last Updated:	

Primary Actors:	Admins
Secondary Actors:	-
Description:	The admins can view / update / remove a user record.
Trigger:	When the admin clicks on Edit/Remove button beside a particular user on the Profile Page under Admin section.
Preconditions:	User needs to be logged in by using admin credentials in the web applications.
Postconditions:	The user is updated as per the changes made by the admin.
Normal Flow:	-
Alternative Flows:	-
Exceptions:	1. Server is down; 2. Database Connectivity Issue; 3. Network Unavailability. 4. The data is not in prescribed format.
Includes:	-
Priority:	Medium
Frequency of Use:	Low
Business Rules:	The changes are reflected to the database.
Special Requirements:	-
Open Issues	-
Assumptions:	User must use their own and correct log in credentials and the changes done must be correct.
Notes and Issues:	-

Use Case ID:	**5**		
Use Case Name:	**Add / View/Update/Remove Books**		
Created By:	Jainam Zobaliya	Last Updated By:	
Date Created:	26 September, 2021	Date Last Updated:	

Primary Actors:	Admins
Secondary Actors:	-
Description:	The admins can add / view / update / remove a book record.
Trigger:	When the admin clicks on Edit/Remove button beside a particular book on the Profile Page under Admin section.
Preconditions:	User needs to be logged in by using admin credentials in the web applications.
Postconditions:	The book is updated as per the changes made by the admin.
Normal Flow:	-
Alternative Flows:	-
Exceptions:	1. Server is down; 2. Database Connectivity Issue; 3. Network Unavailability. 4. The data is not in prescribed format.
Includes:	-
Priority:	Medium
Frequency of Use:	Low
Business Rules:	The changes are reflected to the database.
Special Requirements:	-
Open Issues	-
Assumptions:	User must use their own and correct log in credentials and the changes done must be correct.
Notes and Issues:	-

Use Case ID:	**6**		
Use Case Name:	**User Profile**		
Created By:	Dharmil Gada	Last Updated By:	
Date Created:	26 September, 2021	Date Last Updated:	

Primary Actors:	Users
Secondary Actors:	-
Description:	The users can view and edit their profile.
Trigger:	When the user clicks on Profile button.
Preconditions:	User needs to be logged in to access their profile in the web applications.
Postconditions:	The user can view/ edit their profile.
Normal Flow:	-
Alternative Flows:	-
Exceptions:	1. Server is down; 2. Database connectivity issue; 3. Network unavailability; 4. The data is not in prescribed format.
Includes:	-
Priority:	Medium
Frequency of Use:	Medium
Business Rules:	The changes are reflected to the database.
Special Requirements:	-
Open Issues	-
Assumptions:	User must use their own and correct log in credentials and the changes done must be correct.
Notes and Issues:	-

Use Case ID:	7		
Use Case Name:	**Home Page**		
Created By:	Dharmil Gada	Last Updated By:	
Date Created:	26 September, 2021	Date Last Updated:	

Primary Actors:	Users
Secondary Actors:	-
Description:	The home page shows different books and has various buttons to navigate to other pages in web application.
Trigger:	After log-in, the user will be navigated to home page.
Preconditions:	User needs to be logged in to access the home page in the web applications.
Postconditions:	The user can view current trending books, books recommended for them, top picks etc. The user can pick his/her favourite book and start reading it.
Normal Flow:	-
Alternative Flows:	-
Exceptions:	1. Server is down; 2. Database connectivity issue; 3. Network unavailability 4. The data is not in prescribed format
Includes:	-
Priority:	Medium
Frequency of Use:	High
Business Rules:	The changes are reflected to the database.
Special Requirements:	-
Open Issues	-
Assumptions:	User must use their own and correct log in credentials and the changes done must be correct.
Notes and Issues:	-

Use Case ID:	**8**		
Use Case Name:	**Book Preview**		
Created By:	Dharmil Gada	Last Updated By:	
Date Created:	26 September, 2021	Date Last Updated:	

Primary Actors:	Users
Secondary Actors:	-
Description:	The book preview page shows details about the selected book.
Trigger:	When the user clicks on the book preview button.
Preconditions:	User needs to be logged in to access the book preview feature in the web applications.
Postconditions:	The user gets detailed information regarding the book like the name. the author, description, etc.
Normal Flow:	-
Alternative Flows:	-
Exceptions:	1. Server is down; 2. Database connectivity issue; 3. Network unavailability 4. The data is not in prescribed format
Includes:	-
Priority:	Medium
Frequency of Use:	High
Business Rules:	The changes are reflected to the database.
Special Requirements:	-
Open Issues	-
Assumptions:	User must use their own and correct log in credentials and the changes done must be correct.
Notes and Issues:	-

Use Case ID:	**9**		
Use Case Name:	**Search Books**		
Created By:	Dev Vora	Last Updated By:	
Date Created:	17 September, 2021	Date Last Updated:	

Primary Actors:	Users
Secondary Actors:	-
Description:	The user can search the books based on book Id, name, author's name or the barcode
Trigger:	When the user comes to website provided user is logged in the dashboard appears and the search bar is available at the top
Preconditions:	-
Postconditions:	-
Normal Flow:	Must have a stable internet connection
Alternative Flows:	-
Exceptions:	-
Includes:	-
Priority:	Intermediate
Frequency of Use:	High
Business Rules:	-
Special Requirements:	-
Open Issues	-
Assumptions:	User must provide correct details for getting exact results
Notes and Issues:	-

Use Case ID:	**10**		
Use Case Name:	**Bill & Feedback**		
Created By:	Dev Vora	Last Updated By:	
Date Created:	3 October, 2021	Date Last Updated:	

Primary Actors:	Users
Secondary Actors:	-
Description:	The users have to pay for the books they want to buy or rent and provide feedback after the services.
Trigger:	When the user comes to website provided user is logged in, he adds books to his cart and then clicks
Preconditions:	-
Postconditions:	-
Normal Flow:	Users must have a stable internet connection
Alternative Flows:	-
Exceptions:	-
Includes:	-
Priority:	High
Frequency of Use:	High
Business Rules:	-
Special Requirements:	-
Open Issues	-
Assumptions:	User must be logged in and use credible payment details
Notes and Issues:	-

Use Case ID:	**11**		
Use Case Name:	**Wishlist & Favourite**		
Created By:	Dev Vora	Last Updated By:	
Date Created:	8 October,2021	Date Last Updated:	

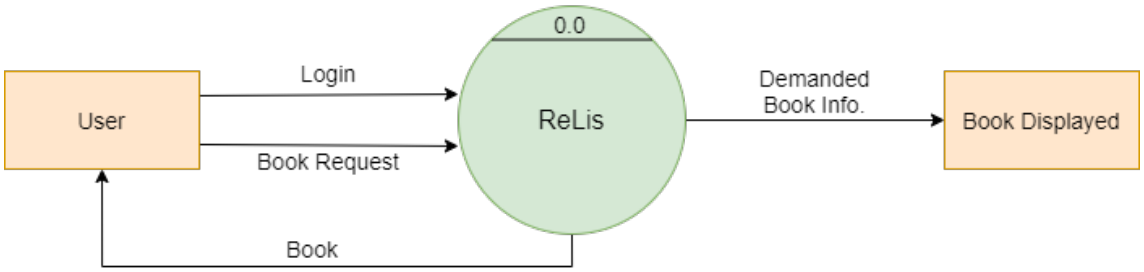
Primary Actors:	Users
Secondary Actors:	-
Description:	The user can save their books as wishlist to buy later or add them as their personal favourites
Trigger:	When the user comes to website provided user is logged in, they are redirected to .
Preconditions:	-
Postconditions:	-
Normal Flow:	Users must have a stable internet connection
Alternative Flows:	-
Exceptions:	-
Includes:	-
Priority:	High
Frequency of Use:	Low
Business Rules:	-
Special Requirements:	-
Open Issues	-
Assumptions:	User must be logged in.
Notes and Issues:	-

Use Case ID:	**12**		
Use Case Name:	**History & Reading Statistics**		
Created By:	Dev Vora	Last Updated By:	
Date Created:	8 October, 2021	Date Last Updated:	

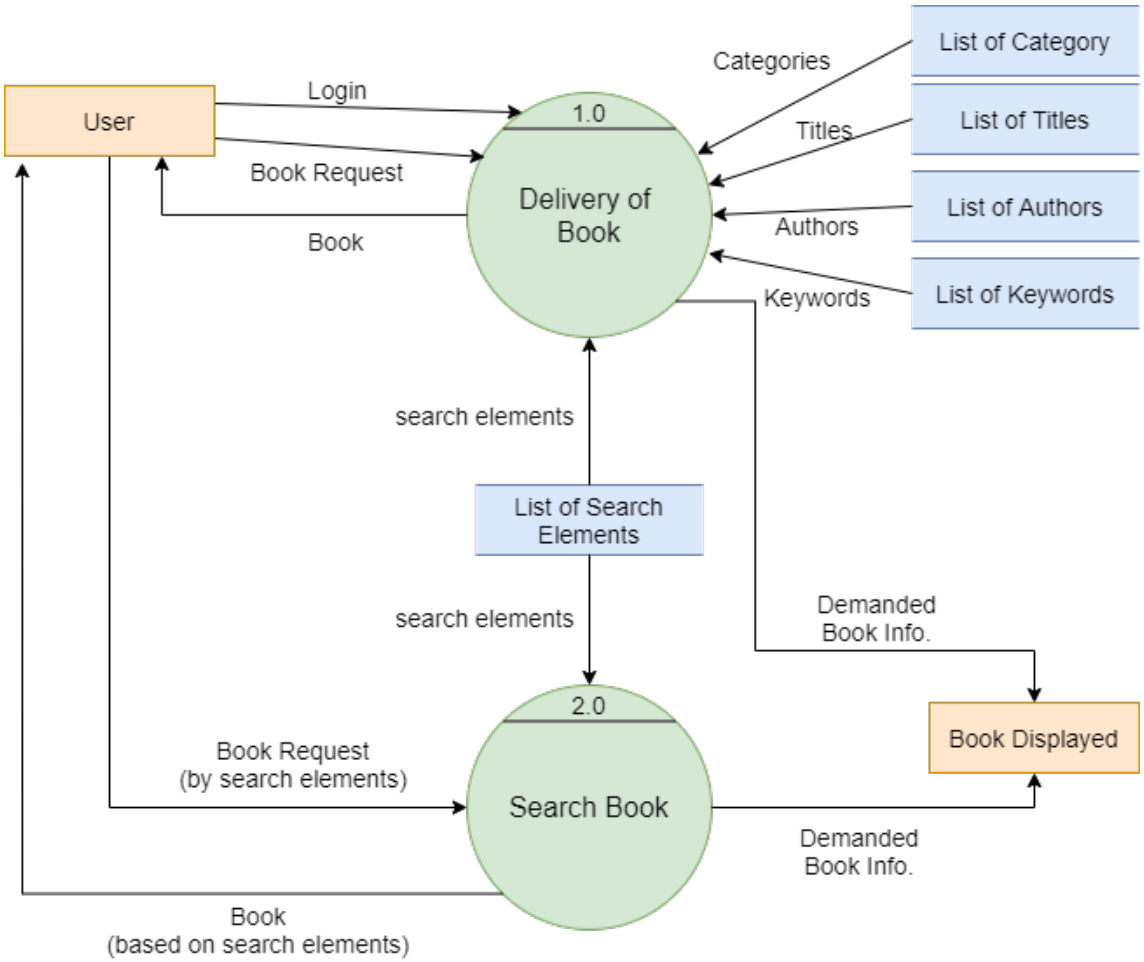
Primary Actors:	Users
Secondary Actors:	-
Description:	The user can view his reading statistics and past history of books provided he is logged in.
Trigger:	When the user comes to website provided user is logged in he can click on the history tab to view these details
Preconditions:	-
Postconditions:	-
Normal Flow:	Users must have a stable internet connection
Alternative Flows:	-
Exceptions:	-
Includes:	-
Priority:	Intermediate
Frequency of Use:	Intermediate
Business Rules:	-
Special Requirements:	-
Open Issues	-
Assumptions:	User must use be logged in
Notes and Issues:	-

4.6 Data flow specifications

4.6.1 Level 0 DFD with description



4.6.2 Level 1 DFD with description



Chapter 5

SOFTWARE TEST DOCUMENT - STD

5.1 INTRODUCTION

5.1.1 System Overview

Libraries are used to store books, but we need a system to navigate to a specific bookeffortlessly. Moreover, it is not possible for all book lovers to visit the library andbuy or rent books, so the smart E-Library system helps to solve this issue. ReLisis a smart e-library system which is Eco-friendly as there is no involvement of hardcopies and thus, it saves paper.

Also, based on the current scenario, screen time formost people has increased drastically which causes negative effects on our body andthus the feature of audio books is beneficial to users.This web app is designed to be run on any device by a user for buying or rentingan book or audio book after choosing from the available book on the Home page orsearching the book through book id, name , author's name or barcode from the webapp. The web app will require a working web browser in the device to work. Theinterfaces will be user friendly and everything will be done digitally. This app willbe made user friendly as it will be used by people in age group 14-65.

5.2 Test Approach

Tests will be conducted to check the efficiency of the software. The reason for this test is to check for any errors and limitations. A list of various planned test are as follows.

5.2.1 System Testing

The entire system will be tested as per the requirements to ensure that there is proper interaction with the database, there is proper input given to the software and that the output generated is relevant.

5.2.2 White box Testing

White box testing, sometimes called glass box testing is a test case design method that uses the control structure of the procedural design to derive test cases. Using white box testing methods, the software engineer can derive test cases that:

- Guarantee that all independent paths within a module have been exercised at least once.

- Exercise all logical decisions on their true and false sides.
- Execute all loops at their boundaries and within their operational bounds.
- Exercise internal data structure to assure their validity.

The coding team took all care to test the code and guarantee that it meets all the specifications as well as logically correct. All loops were tested and all internal data structures evaluated and verified.

5.2.3 Black box Testing

Black box testing is a method of software testing that's tests the functionality of application as opposed to its internal structures of workings (see Whitebox testing). Specific knowledge of application's code/internal structure and programming knowledge in general is not required. The tester is only aware of what the software is supposed to do, but not how i.e. When he/she enters a certain input, he/she gets a certain output; without being aware of how the output was produced in the first place. Test cases are built around specifications and requirements i.e., what the application is supposed to do. It uses external descriptions of the software, including specifications, requirements and design to derive test cases.

5.2.4 Unit Testing

Unit testing focuses verification effort on the smallest unit software design- the module. Using the procedural design description as a guide, important control paths are tested to uncover errors within the boundary of the module. The module interface is tested to ensure that information properly flows into and out of the program unit under test. The local data structure is examined to ensure that data stored temporarily maintains its integrity during all steps in algorithmic execution. Boundary conditions are tested to ensure that the module operates properly at boundaries established to limit or restrict processing. All independent paths (bases paths) through the control structure are exercised to ensure that all elements in a module have been executed at least once. And finally all error-handling paths are tested. Application interface of our system was unit tested at all levels of implementation, right from start of code writing, to integrating the code with other modules. Every module was tested fully to check its syntax and logical correctness. Error handling was implemented into relevant modules so that the code doesn't crash on errors.

5.2.5 Integration Testing

Integration testing is the phase in software testing in which individual software modules are combined and tested as a group. It occurs after unit testing and before validation testing.In our project we would be testing integration of different modules such as booking and payment etc.

5.3 Test Plan

5.3.1 Test Plan Objectives

- Testing is the process of executing the program the program with the intention of finding an error.

- A good test is one that has high probability of finding an as yet undiscovered error/bug.
- A successful test is that which uncovers as-yet-undiscovered error.

5.3.2 Feature to be tested

Feature to be tested from user point of view:

- User Interface (Registration and Login).
- Functionalities of the application.
- Reliability of sharing suggestions.

5.3.3 Testing tools and Environment

Hardware Configuration:

- 8 GB Ram
- 120 GB Hard Disk
- 1.2 GHZ Processor

Software Configuration:

- Android Studio v.4.2.2
- Microsoft Visual Studio Code v.1.60
- Flutter SDK 2.1.0
- Node.JS v.14.17.6

5.3.4 Test Cases

Test case Id	Test case	Description	Input	Expected Output	Actual Output	Status
1	Registration	Enter User personal details.	Click on Register	Entered data is verified, validated and stored in the database with an OTP generated	User is registered	Pass
2	Saving Data in Database	Verify the values from the database	User enter their details	User entries should be stored in the database	Entry is saved in the database	Pass
3	Email and OTP verification	Done during the time of registration	Enter correct email address and OTP	User details are verified	Email and OTP verified	Pass
4	Login	Login fails on entering wrong credentials	Wrong password and username	Display message “Login Failed and try again”	Invalid Credentials	Fail
5	Login	Login successful on entering correct credentials	Correct password and username	User is redirected to the Home page	Login successful	Pass

6	Add/Publish Book Form	Enter the details of the book	Entering correct details about the book	Book is added to the database	Book is successfully added	Pass
7	Buy or Rent Book page	User can buy/rent a book	Choose a interested book/audio book and complete the payment	User will be able to read the book/audio book	User will be able to read the book/audio book	Pass
9	Search Book	Searching the books	Provide book details/barcode to search the book	Required book is found and shown to user in website	Required book is found and shown to user in website	Pass

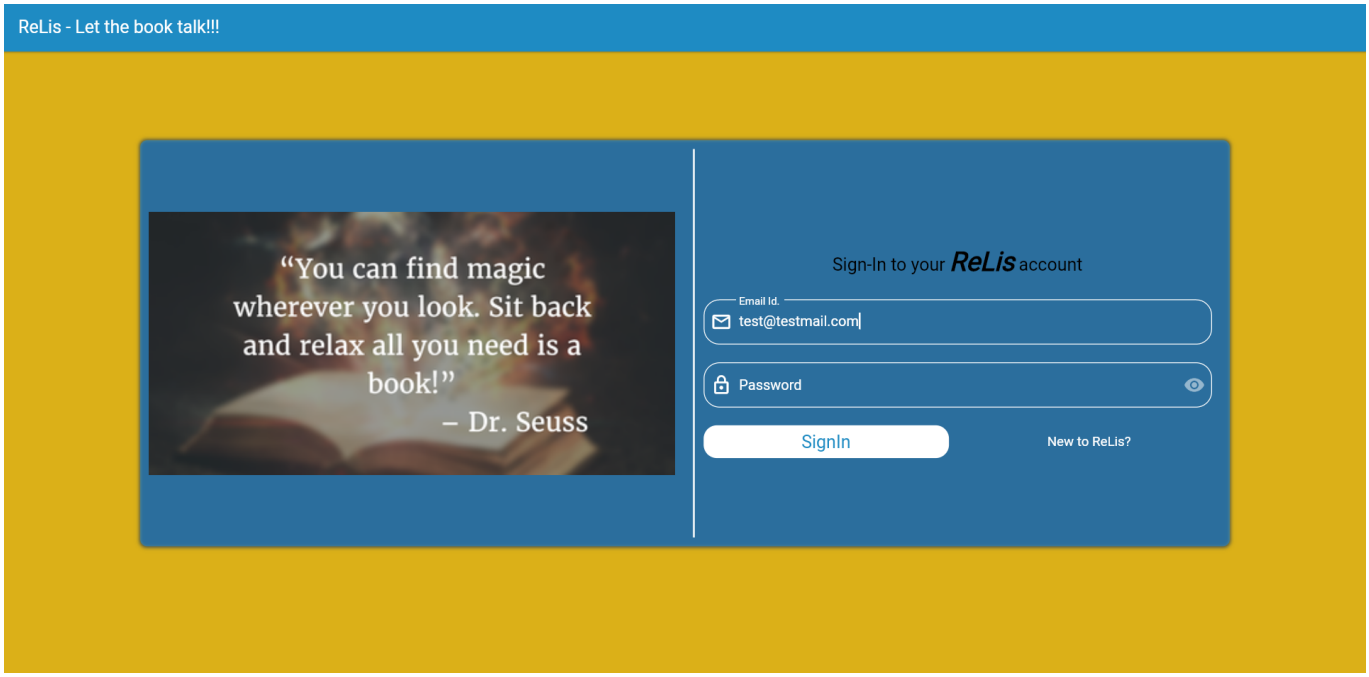
```dart
import 'package:flutter_driver/flutter_driver.dart';
import 'package:test/test.dart';

Run | Debug
void main() {
  Run | Debug
  group("Flutter Auth App Test", () {
    final emailField = find.byValueKey("email-field");
    final passwordField = find.byValueKey("password-field");
    final signInButton = find.text("Log In");
    final userInfoPage = find.byType("BillPage");
    final snackbar = find.byType("SnackBar");

    FlutterDriver driver;
    setUpAll(() async {
      driver = await FlutterDriver.connect();
    });
```

test_driver > app_test.dart > ...

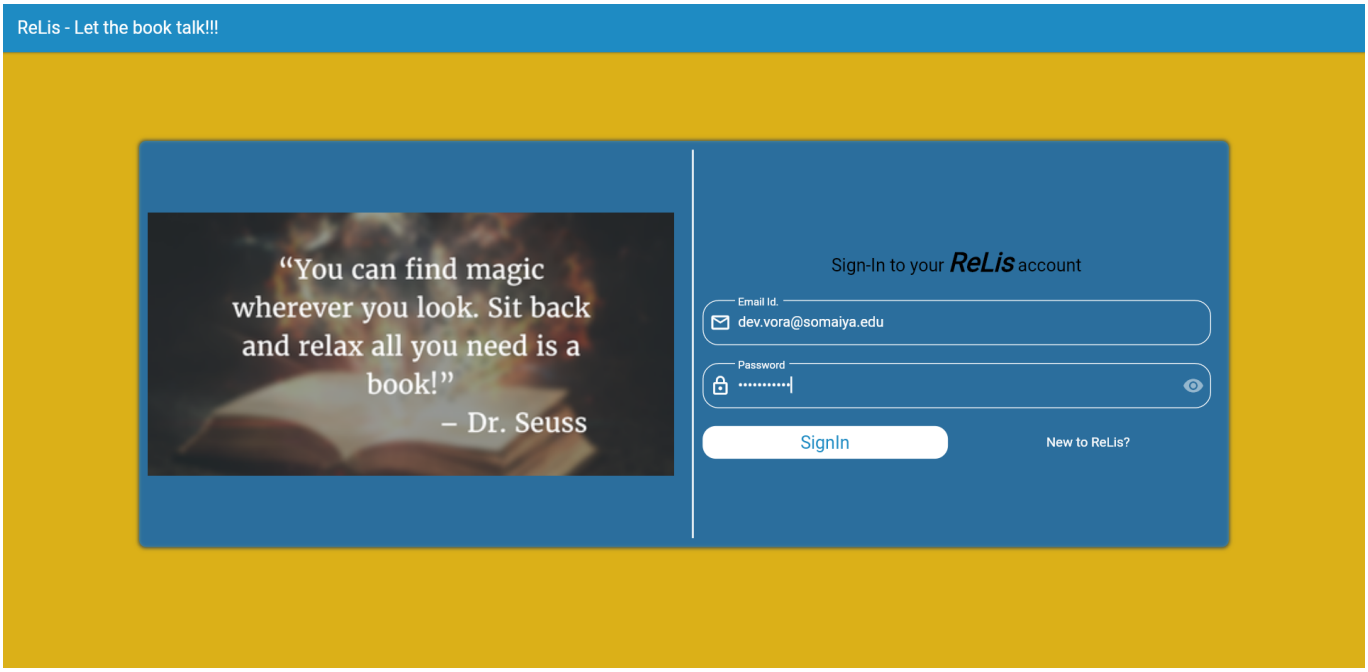
```
Run | Debug
23    test(
24        "login fails with incorrect email and password, provides snackbar feedback",
25        () async {
26      await driver.tap(emailField);
27      await driver.enterText("test@testmail.com");
28      await driver.tap(passwordField);
29      await driver.enterText("test");
30      await driver.tap(signInButton);
31      await driver.waitFor(snackbar);
32      assert(snackbar != null);
33      await driver.waitUntilNoTransientCallbacks();
34      assert(userInfoPage == null);
35      await Future.delayed((Duration(seconds: 10)));
36    });
37
      Run | Debug
38    test("logs in with correct email and password", () async {
39      await driver.tap(emailField);
40      await driver.enterText("dev.vora@somaiya.edu");
41      await driver.tap(passwordField);
42      await driver.enterText("Devvora0308");
43      await driver.tap(signInButton);
44      // await driver.waitFor(Duration(seconds: 10)););
45      assert(userInfoPage != null);
46      await driver.waitUntilNoTransientCallbacks();
47      await Future.delayed((Duration(seconds: 10)));
48    });
49  });
50  }
51
```



```
00:00 +0: Flutter Auth App Test (setUpAll)

VMServiceFlutterDriver: Connecting to Flutter application at http://127.0.0.1:52436/N67szywxb4I=/
VMServiceFlutterDriver: Isolate found with number: 1192022569101471
VMServiceFlutterDriver: Isolate is paused at start.
VMServiceFlutterDriver: Attempting to resume isolate
I/GED     (19106): ged_boost_gpu_freq, level 100, eOrigin 2, final_idx 15, oppidx_max 15, oppidx_min 0
VMServiceFlutterDriver: Connected to Flutter application.
00:02 +0: Flutter Auth App Test login fails with incorrect email and password, provides snackbar feedback

I/flutter (19106): User is currently signed out!
VMServiceFlutterDriver: tap message is taking a long time to complete...
00:09 +0 -1: Flutter Auth App Test login fails with incorrect email and password, provides snackbar feedback
```




```
00:09 +0 -1: Flutter Auth App Test logs in with correct email and password

00:09 +0 -2: Flutter Auth App Test logs in with correct email and password
```

Chapter 6

RESULT AND DISCUSSION

Role	Description	Team Member
Project Manager	Overall planning, risk analysis, execution planning, monitoring and closure planning	Dharmil Gada
System Analyst	To develop a design of the required system and ensuring that all requirements are satisfied	Dharmil Gada, Dev Vora
Technical Lead	Leading the development team	Jainam Zobaliya
Development Team	Implementing the functionalities required in the application through coding	Dev Vora, Jainam Zobaliya, Dharmil Gada
Database Administrator	To design and maintain the database and to add, modify, delete, retrieve data from the database efficiently	Jainam Zobaliya, Dev Vora
UI/UX Designer	To prepare a prototype of the website and UI	Dev Vora, Jainam Zobaliya
Testing Team	To check whether the functionalities work as expected and to find out bugs in the system	Dev Vora, Jainam Zobaliya, Dharmil Gada

Chapter 7

CONCLUSION AND FUTURE WORK

7.1 Conclusion

This unique project of ours will be solving the major problem of visiting to library and reading books in the danger times of the Novel Coronavirus Pandemic as this Pandemic has nearly stopped all such exciting and interesting activities. Recent developments in Internet and WWW have brought significant change in the way the generation, distribution and the accessing of enormous amount of information published through various modes and means. Today digital libraries have overtaken managing the information world using both commercial and open source software. ReLis is a Digital Library System which serves the sole purpose of providing user the ability to read and publish books. The system also provides useful functionalities like Audio-book, translation, buy/rent/publish a book, bar-code search, user statistics and many more. This web-app has the capacity and innovation to become an end to end real world project. All 3 of us, the group members are looking forward and take this project to turn this into actual reality and try and create more and more functionality in terms of technical and also we're looking ahead to bring timely updates by implementing new and necessary functionalities and try to reduce the bug in order to help people organize and enjoy reading.

7.2 Future Scope

1. Build and Deploy the web-app.
2. Use an optimal Algorithm for the translating and recommending books and other features for better performance.
3. Add the net Banking options.
4. Add the reading time notification.
5. Create the app for Android / IOS mobile.
6. Optimize Dashboard for the admins to monitor various activities.
7. To develop secure and trusted website.

REFERENCES

1. https://flutter.dev/

2. https://nodejs.org/en/docs/

3. https://www.overleaf.com/learn

4. https://www.researchgate.net/publication/269954368_Development_and_Characteristic_of_Digital_Library_as_a_Library_Branch

5. http://ieeexplore.ieee.org.library.somaiya.edu/stamp/stamp.jsp?tp=&arnumber=7559626

6. http://ieeexplore.ieee.org.library.somaiya.edu/stamp/stamp.jsp?tp=&arnumber=4118583

7. http://ieeexplore.ieee.org.library.somaiya.edu/stamp/stamp.jsp?tp=&arnumber=7416895

8. http://ieeexplore.ieee.org.library.somaiya.edu/stamp/stamp.jsp?tp=&arnumber=8663914

9. https://medium.com/flutter-community/how-to-make-a-flutter-app-with-high-security

10. http://lib.unipune.ac.in:8080/xmlui/bitstream/handle/123456789/7603/09_chapter%202.pdf?sequence=9&isAllowed=y

11. https://www.appknox.com/blog/why-is-flutter-the-ideal-framework-for-optimum-app-s

12. https://www.softwaretestinghelp.com/security-testing-of-web-applications/

13. https://www.statista.com/statistics/250007/downloading-or-borrowing-e-books-on-a-

ACKNOWLEDGEMENT

I, **(Project Manager, Developer)** and the **entire software team** have taken 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. This Project became a reality with the kind support and help of many individuals. We would like to extend our sincere thanks to all of them.

We are highly indebted to **K.J. Somaiya College Of Engineering** for their guidance and constant supervision as well as for providing necessary information regarding the project also for their support in completing the project. We thank all everyone for their positive support and guidance.

We would like to express our gratitude towards **Prof. Purnima Ahirao** for guiding us throughout the project. We feel inspired and motivated to created more such projects in the near Future. Thank You.

Group Members:

Dharmil Gada - 1814076
Dev Vora - 1814120
Jainam Zobaliya - 1814123