" بسم الله الرحمن الرحيم "

Department of Computer Science

COMP4300- Graduation Project

Second Semester 2024/2025

Title of Project: SmartLib

Group Member with IDs:

Anas Karakra	**1200467**
Yousef Sharbeh	**1202057**
Mahmoud Duhaidi	**1200340**

Supervisors: Dr. Ahmad Abusnaina

Abstract

SmartLib is a website that makes it easier for readers to access and enjoy digital books. using technologies such as suggestion system, Web development, Text-to-speech, Text-analysis. This website aims to make the book-finding process easier and make the reading process smoother. Also, the Library supports gamification system to maintain reader’s engagement.

When a reader selects a book, a suggestion system uses various factors determined by the reader's preferences and more functionality to assist in book selection. Once the book is chosen, a Text-analysis algorithm analyzes its text, preparing a text summary for the reader. Additionally, readers have the option to listen to the text of the book using text-to-speech functionality and can translate the text as needed.

Table of Contents

Table of Figure:

Chapter 1: Introduction

1.1 Overview:

Libraries and books are fundamental to share knowledge, education and inspiring imagination. Libraries provide access to diverse resources for learning, while books enrich our understanding and preserve culture and history. In the digital age, the role of libraries is evolving, which it integrating the technologies to enhance Reader's experience.

Our SmartLib is a website that makes it easier for readers to access and enjoy digital books. By using technologies such as machine learning, Deep Learning, web development, Text-to-speech, Text-analysis. This website aims to make the book-finding process easier and enhance the reading experience for readers who struggle to find their target books efficiently and make the reading process smoother. Also, the Library supports gamification system to maintain reader's engagement.

On our website there is a suggestion system that analyzes a variety of factors, including the reader's suggestion preferences (last updated book, most rating book, most reading book and most favorited book). This suggestion system assists readers in selecting books that align closely with their preferences, which leads to improving their experience of finding books and making it faster and easier.

Additionally, the website provides features like text summarization algorithm that saves readers time by summarizing books for readers whose want to have a quick look at the text rather than reading the complete book, also text-to-speech functionality, enabling readers to listen to the content of the book in addition to reading it visually, translation capabilities, allowing users to translate the text of the book into their preferred language.

Moreover, the Library supports a gamification system to keep readers interacting with the system continuously. By adding elements such as points, and levels. which the website encourages continuous interaction with it.

1.2 Aims & Objectives:

1.2.1 Aims:

The primary aim of the SmartLib website is to facilitate readers' access to and enjoyment of reading digital books and aims to improve the reading experience and make the reading process more efficient.

1.2.2 Objectives:

- **We developed suggestion system for personalized book recommendations.**
    - We used a suggestion system that analyzes a variety of factors, including the reader's suggestion preferences (last updated book, most rating book, most reading book and most searched book). This system assists readers in selecting books that align closely with their preferences.
- **We implemented text summarization model**.
    - We used a text summarization algorithm that analyze the text, extracting main points, and preparing a text summarize of book for reader
- **We Integrated text-to-speech functionality for audiovisual reading experiences.**
    - website offers advanced features such as text-to-speech functionality, enabling readers to listen to the content of the book in addition to reading it visually
- **We Integrated language translation for different language accessibility.**
    - It has translation capabilities, allowing users to translate the text of the book into their preferred language.
- **Support a gamification system.**
    - The library supports a gamification system to keep readers interact with the system continuously. By adding elements such as points, and levels. which the website encourages continuous interaction with it.

Chapter 2: Background and Literature Review

This chapter explores the evolution of electronic libraries and the technology that is used in libraries.

2.1 Background:

In today's digital age, the SmartLib project aims to use technologies such as suggestion system, Web development, Text-to-speech, Text-analysis to completely change how users access and enjoy digital books. By simplifying the book-finding process and enhancing the reading experience, this website addresses the challenges faced by readers.

The using of suggestion system plays an important role in analyzing various factors such as suggestion preferences (last updated book, most rating book, most reading book and most favorite book) to provide personalized book suggestion, which improving the efficiency of book selection (Géron, 2019).

The text summarization is a summary of text that represents the most relevant information within the context by using automatic text summarization. If there is text in a document or multiple documents, it uses extractive text summarization to give a summary of the same text but with reordered sentences. This process involves techniques such as natural language processing (NLP) that use the Text Rank algorithm to understand the context and semantics of the text. Once the text is obtained, it split that text into individual sentences, and if the sentences have stop words or stemming words, they are removed. Then, feature extraction represents each sentence using data structure like vector representation, and the similarity between sentence vectors is calculated and stored in a matrix. The similarity matrix, which includes verbs, objects, and adjectives, is converted into a

graph. Then, we calculate the sentence rank, that gives ranked sentences from

which we get top-ranked sentences for the final summary. To sum this up, the integration of text summarization enables AI models to analyze the content of a book and provide a summary by identifying key points and extracting essential information. This process involves NLP techniques, such as the Text Rank algorithm, to understand the text's context and semantics. Once the key points are identified, the AI model generates a summary that captures the main ideas and essential details of the text, enhancing reader comprehension (Martin, 2019).

the inclusion of translation capabilities enables users to translate book text into arabic language, using translation APIs like Google Translate that work by using a large collection of languages. When you type in text to translate, the API figures out the starting language and translate it into the desired language using Neural Machine Translation (NMT) algorithm, that employs deep learning techniques, specifically neural networks, to improve translation quality by analyzing entire sentences or paragraphs at a time, rather than individual words or phrases. NMT algorithms translate words by using neural networks to encode input sentences into vectors, decode them into target language words, and learn from large dictionary datasets to improve accuracy. This approach allows the algorithm to better capture context, and distinguish between languages, resulting in more accurate and natural translations. Then, it gives the user back the translated text (Wu, 2016).

Furthermore, the website offers advanced features including text-to-speech functionality, allowing users to listen to the content of the book in addition to visually reading it, which meets need of the users who prefer auditory learning, and this achieved by Google Text-to-Speech API use a machine learning algorithm called WaveNet for speech compound, it is a type of neural network known as a convolutional neural network (CNN). It differs from traditional CNNs used in image processing because it is specifically designed to handle sequential data, such as audio waveforms. And it is a strong generative model that produces good human speech. WaveNet is a deep learning model that creates realistic human speech by generating audio one tiny piece at a time. It looks at a wide context of previous audio samples to predict the next one, ensuring the speech flows naturally. During training, it learns from real speech recordings to understand how speech sounds. When generating speech, it uses this learned knowledge, along with details about pronunciation, to produce natural and meaningful audio.

In the Google Text-to-Speech API, everything starts with text analysis. This API first normalizes the text into a standard format and then analyzes it to understand the structure of the text based on words, phrases, and sentences. After this analysis comes phonetic transcription, a step that turns text into a series of phonetic units.

It then predicts prosody features, for example, stress, and rhythm, to make sure it will sound natural. That is, understanding the context and text meaning to apply proper prosody makes generated speech more realistic.

finally makes use of this WaveNet model in the API to generate the audio waveform. Here, this model conditions on the phonetic and prosody features derived above and generates the waveform sample by sample in an autoregressive manner. Additional processing on the generated waveform may be done to result in high-quality speech output with clarity after filtering, noise reduction, and volume adjustment all of these explain in (Aaron van den Oord, 2016).

The gamification system keeps readers interact with SmartLib website using points and levels. Readers earn points for daily logins (10 points), uploading books (50 points), completing books (100 points), rating and reviewing books (20 points), and reviewing the system (30 points). As they collect points, they progress through different levels and unlock new features. At 500 points, they reach the bronze level and unlock the translation feature. At 1500 points, they reach the silver level and unlock the audio feature. At 3000 points, they reach the gold level and unlock the text-summarize feature.

Through the seamless integration of these technologies, the SmartLib project provides users with a seamless and comfortable reading experience that makes them excited to explore and enjoy digital books.

2.2 Related Work

2.2.1 Book-Bub:

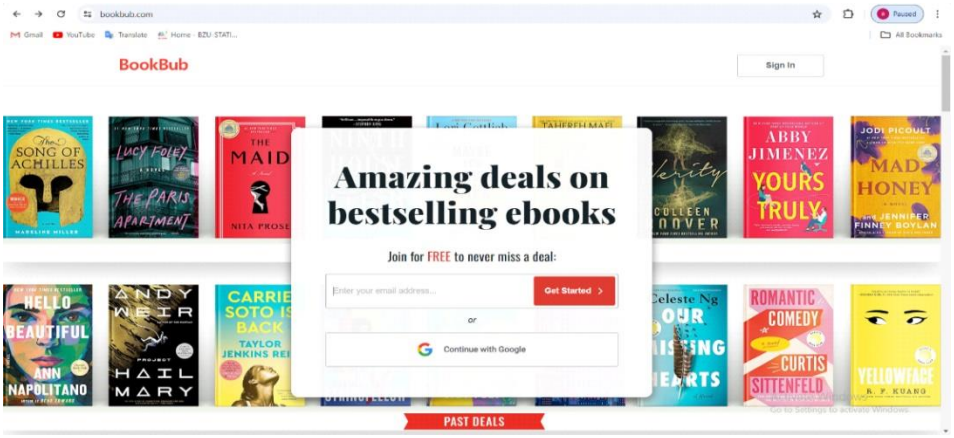


Figure 1:Book-Bup website

(Book Bub, 2024) is a website that provides books based on the user's personal preferences, which helps reader to find the target book. Users register on the website, choose their favorite type of book, such as action, adventure, and romance etc. and then the website offer books suits with user recommendations. There are two methods for recommendations. The first way is to select recommendations, and the books related to the recommendations will appear. The second way is to not select any recommendations, and then the bestsellers book will appear automatically, Also Users receive daily new letters containing e-book that match their preferences.

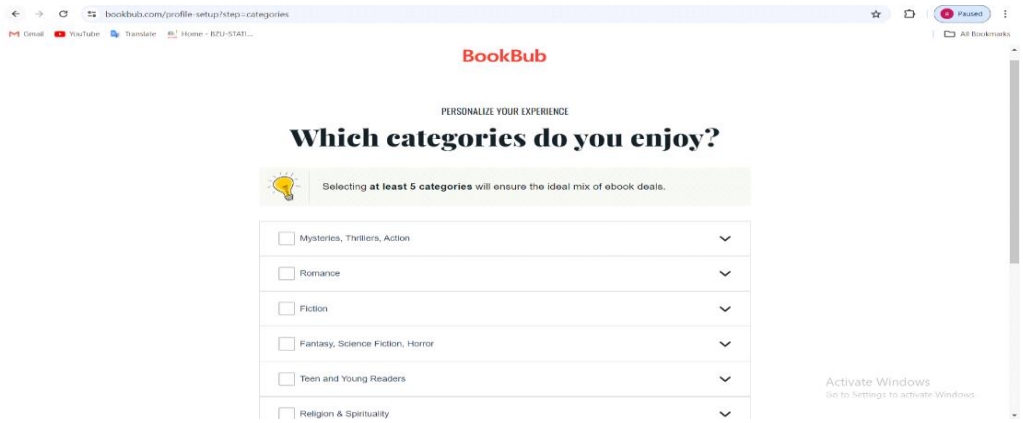


Figure 2:Book-Bub website

This website make find target book easier and faster and save time of users.

2.2.2 Wordkraft AI Book Summarizer:

(WordKraft, 2024) AI Book Summarizer is a service that summarizes books, helping readers understand key topics and points without having to read the entire text of a book. Users simply enter the title and author of the book, then select the type of summary whether it's a brief overview, or a chapter summary, or a deep concept summary, then it's had an advanced options like choose language of text summarization and tone of voice like professional, serious, emotional that make ai read the text, after that the AI model provides a condensed version of the content.

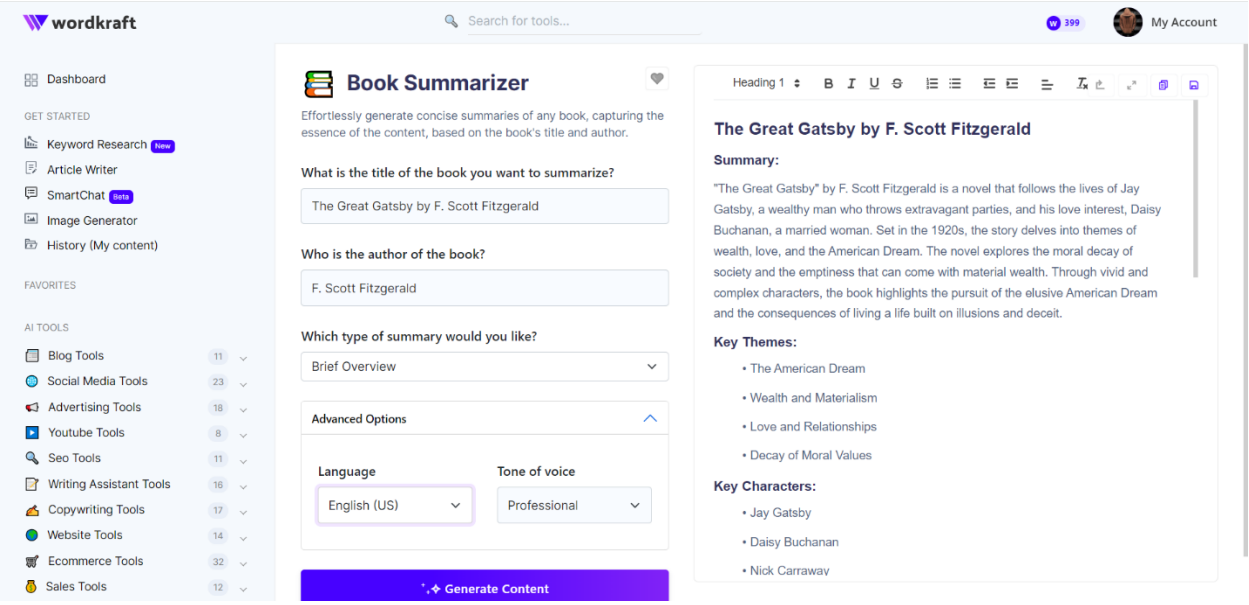


Figure 3:wordKraft website

This service is useful for readers who looking to quickly understand the key topics and points of a book without investing significant time in reading it cover to cover.

2.2.3 open Library:

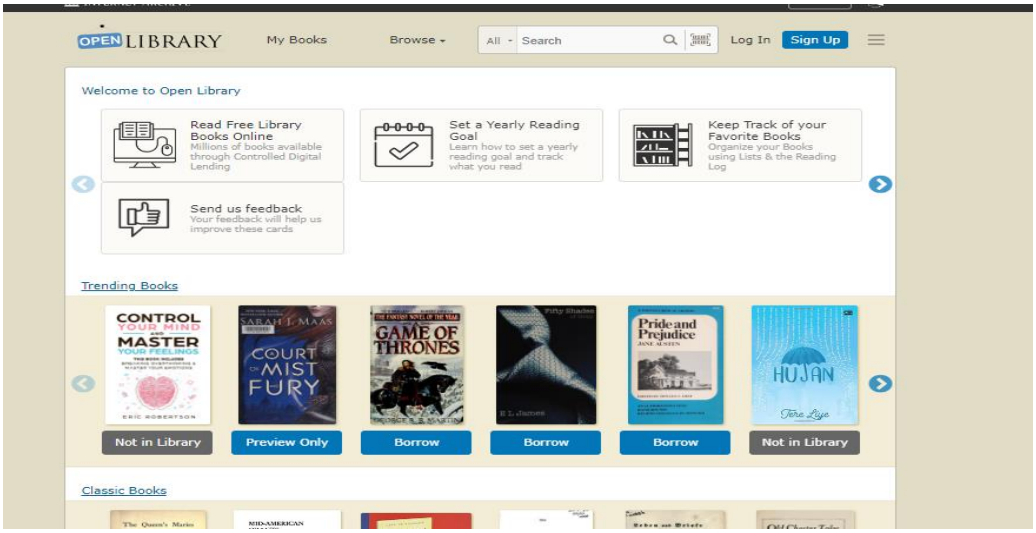


Figure 4:open library website1

(open library, 2024) aims to provide free access to a wide range of books in digital format, and this website has a big collection of books in difference topic that makes reader read book audiovisual.

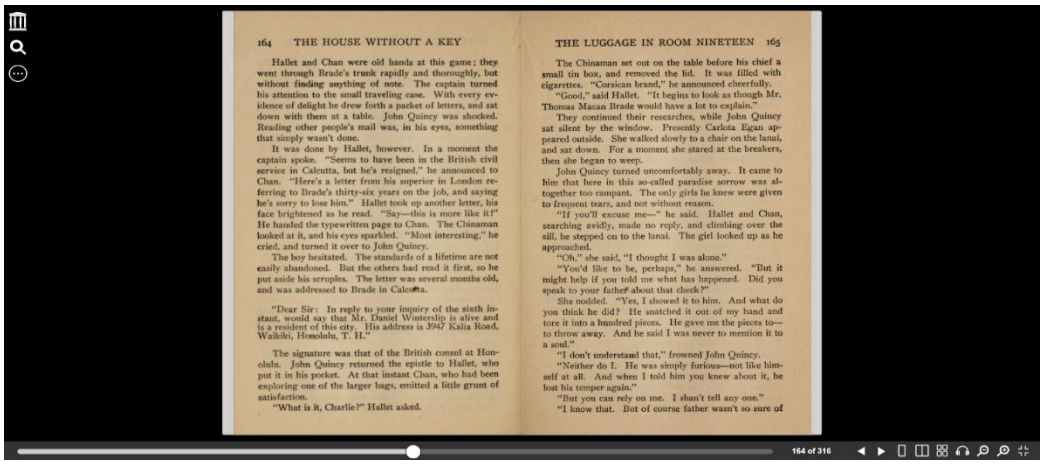


Figure 5:open library website 2

User search for books by title, author, subject, or even browse category (classics, modern, bestsellers, academic texts) or can chose book by website recommendation. When a user finds a book, it has the option to borrow the book electronically or borrow physical copies that may be available in libraries near the user, after that user can choose the reading

format that suits its preferences, also user can use features like ai reading book, bookmarks, notes, and reading progress tracking to enhance user experience.

Table of feature and comparison with these websites:

Websites/ feature	**Recommendation system**	**Text Summarize**	**Read audiovisual**	**Translate text**
Book-Bub	Yes	No	visual	No
Wordkraft	No	Yes	visual	Yes
Open Library	No	No	both	No
SmartLib	Yes	Yes	both	Yes

Chapter 3: Approach

This chapter shows our set working plan, methodology, and the resources that will be used. It will also address the ethical considerations of this project.

3.1 Methodology

we developed the SmartLib project involved several key steps to ensure efficient progress and successful outcomes.

First, we carefully planned out the project, figuring out what we needed to do and how we could do it best. We investigated different technologies and methods to make sure we were on the right track, and also, we used the Agile development process to make sure that we could be flexible and work quickly.

Then, we explored various technologies to see which ones would work best for our project. We wanted to use things like machine learning and text analysis to make our library smart and helpful for readers. We aimed to address several common problems that readers face when using digital libraries.

One problem reader often encounter is struggling to find the book from digital library, and this leading to waste time, and to solve this, we used in our website a suggestion system that analyzes a variety of factors, including the reader's suggestion preferences (last updated book, most rating book, most reading book and most favorite book). This system assists readers in selecting books that align closely with their preferences, which leads to improving their experience to find book and make it faster and easier.

Another challenge readers face is may their doesn’t have much time to read entire book, or they just want to know summary of specific theme of book, and to solve this we use a Text Rank (extractive text summarization) which combine text of book, then processed the text, then represented

sentences(feature extraction),then ranked sentences, finally give summary of text for reader's.

Some readers may struggle to find (human) audio versions of all books, and to solve this our website offers advanced features such as text-to-speech functionality, enabling readers to listen to the content of the book in addition to reading it visually.

Some readers may struggle to find translations or words of book. To solve this issue, we make a website that has translation capabilities, allowing users to translate the text of the book into arabic language.

Lastly, Readers who are new to digital libraries or have specific preferences, struggle with staying engaged and motivated to use the website continuously, and to solve this we make our website support a gamification system to keep readers interact with system continuously. By adding elements such as points, and levels. which the website encourages continuous interaction with it.

When we designed the project, we focused on making it easy for the reader to use. We made the interface to be friendly, especially for readers and for those how has struggle to read a book.

We make sure that the project will work well on a website application. We also set up a system that will include anything readers may need to use in our library.

During the whole process, we stayed flexible and open to changes. We wanted to be able adapt quickly to any new ideas or feedback.

3.2 Resources

To successfully complete the SmartLib project, we required two main types of resources: hardware and software.

3.2.1 Software resources:

A. Web development tools: These will include CSS for style, HTML and JavaScript for front-end, Full stack framework (Django (Python framework)).

B. Database management system: To manage and store data, we'll need a system like PostgreSQL or MySQL.

C. Text Summarize Libraries: to create and train our text processing models, such as Sumy in python.

D. Development environment: Writing and testing code will require an Integrated Development Environment (IDE), such as PyCharm or Visual Studio Code.

E. Version Control System: Git will be used as the version control system to handle collaborative modifications to the code.

F. Cloud services: To host the web application, we may require services like AWS or Google Cloud.

G. API's: to translate text Cloud Translation API (Google APIs), and to text-to-speech Cloud Text-to-Speech API (Google Cloud).

3.2.2 Hardware resources:

Computers: To complete development and execute text summarization algorithm, a computer with processing speed and memory is needed.

Graphics Processing Unit (GRU): A powerful GPU may be required to run text summarization algorithm effectively, depending on their complexity.

3.3 Ethical Considerations

Data Privacy and Protection: We will follow strict rules about data privacy and protection on passwords, which we will encrypt these data in database.

Copyrights reserved: We will follow strict rules about book copyrights, which will prevent the upload of any book that may violate copyright laws.

Chapter 4: System Analysis

This chapter shows functional and non-functional requirements and Use Case and Diagrams.

4.1 functional Requirements

UR1: The system shall allow Readers to create an account:

- SR1.1: Reader creates an account then fill their information (Username, Email, password).
- SR1.2: The password length of 8 or more digits contains (character, number and symbol) after filling it will be encrypted.
- SR1.3: Reader choose their preferences of book types.

UR2: The system shall allow Readers to login into system:

- SR2.1: User fill their information (Username, password) into system.

UR3: The system shall provide Readers with training simulation instructions:

- SR3.1: User follow the system instruction of how to use website or can skip instruction.

UR4: The system shall allow Readers to search for Book:

- SR 4.1 Reader should type on search bar then the system should allow the Reader to search for specific criteria (name, author, type, barcode).
- SR 4.2 Reader should filter book after search result by (date, language, alphabetical, number of pages, rating).

UR5: The system shall provide Readers with recommendation book:

- SR 5.1: Reader should view book based on their preferences with recommendation from system (Last updated book based on (year, month), Most rating book based on number of stars, Most reading book, Most searched book).

UR6: The system shall allow Readers to view Book:

- SR 6.1: Reader should show book information.
- SR 6.2: Reader shall be able to share book, by link.
- SR 6.3: Reader shall show rating and review of book.
- SR 6.4: Reader shall add notes of book.
- SR 6.5: Reader shall read book audiovisual.
- SR 6.6: Reader attached with mark-up of its read progress.
- SR 6.7: Reader provides text summarization of book (**Own Model**).
- SR 6.8: Reader shall switch between multiple language (Arabic and English) of book text/words.

UR7: The system shall allow Readers to Rating and Review the Book:

- SR 7.1: Reader should give five stars or less as a rate for book.
- SR 7.2: Reader should write comments as a review for book.

UR8: The system shall allow Readers to review system:

- SR 8.1: Reader can give feedback about system.
- SR 8.2: Manager can view feedback about system.

UR9: The system shall allow Readers to upload a book:

- SR 9.1: Reader can upload books using friendly interface that guide the Reader through the upload process.
- SR 9.2: System checks the file format and size of file, which means only accept pdf format and maximum size limit.

UR10: The system shall allow Readers update its information:

- SR 10.1: Reader shall allow to edit its profile information (name, password, email, preference)

UR11: The system shall allow Readers delete its Account:

- SR 11.1: Reader shall allow to delete their account.

UR12: The system sends to Reader's notification:

- SR 12.1: Reader should be reminded by the system to continue reading book.
- SR 12.2: Reader should receive notification from system that the manager accepts or rejects (with explain of reject reason book) upload their book.

UR13: The system should support gamification for Readers:

- SR 13.1: Reader should get points in daily login.
- SR 13.2: Reader should get points in every upload of book.
- SR 13.3: Reader should get points when complete book reading.
- SR 13.4: Reader should get points when Rating and Review the Book.
- SR 13.5: Reader should get points when review system.
- SR 13.6: Reader should reach level bronze when there have 500 point, and open (translate feature).
- SR 13.7: Reader should reach level silver when they have 1500 points, and open (audio feature).
- SR 13.8: Reader should reach level gold when they have 3000 points, and open (text-summarize feature).

4.2 Nonfunctional Requirements

1) Performance:

1.1 Speed: website Pages loading less than 2 seconds

1.2 The system must be able to process the summarize book and produce the summarized text in a maximum of 1 minute.

2) Security:

2.1 Data protection: provide Password encryption to prevent any hacking to password information

2.2 all website transactions must be safe and secure.

3) usability:

3.1 contains a training simulation to facilitate the use of the system

3.2 The interface contains appropriate colors to suit users who suffer from poor vision

4) Reliability:

4.1 Availability: System should word 24/7 with downtime not exceeding 0.02% per year

4.2 all function of the system work as expected under normal operating conditions and with a lot of user pressure

5) Compatibility:

5.1: Browser: compatible with all browsers with latest versions.

5.2: Device: compatible and designed for all devices such as laptop, pc, iPad and tablet, etc.

4.3 Use Case

Use case1: Readers Registration and Login:

AI Library Book System: Reader Registration and Login	
Actor	Reader
Description	The Reader create account and can login into system with it.
Pre-conditions	• Reader has access to the internet
Sequence/Flow of Events	• Reader goes to registration page. • Readers fill in their information then click on the register button. • System validates information and stores it on database. • Reader login to the system with new account.

Post-conditions/Response	• **login**:
	  • The system validates the Reader information.
	  • If successful login, then the Reader redirected to homepage.
	  • If the login fails (incorrect information), an error message is displayed prompting the Reader to retry enter correct information.

Use case 2: Reader search for book:

AI Library Book System: Reader search for book	
Actor	Reader
Description	The Reader search for book on search bar or category menu.
Pre-conditions	• Reader is logged in the system • Reader goes to search bar • Reader goes to category menu

Sequence/Flow of Events	• If Reader goes to search bar:
	  • Reader type specific criteria (name, author, type, barcode) on search bar.
	  • Reader filter book after search result by (date, language, alphabetical, number of pages, rating).
	• If Reader goes to category menu:
	  • Reader chose the book category type
	  • Reader filter book after chosen the book category type by (date, language, alphabetical, number of pages, rating).

Post-conditions/Response	The Reader can select a book from the search results or category menu and redirected to book page.

Use Case 3: Reader View Book:

AI Library Book System: Reader view book	
Actor	Reader
Description	The Reader can interact with book page.
Pre-conditions	• Reader Logged in system. • Reader clicks on target book.

Sequence/Flow of Events	• If Reader, click on write notes button:
	  • The reader writes the desired notes text, then the system saves it on database.
	• If Reader Read book:
	  • Reader interaction with section that have pages of book.
	• If Reader markup for progress:
	  • Reader drags flag to a specific page number, then the system saves it on database.
	• If Reader, click on play auto-reader button:
	  • Reader Play auto-reader.
	• If Reader, click on switch language text:
	  • Reader change between language of text of book (Arabic, English)
	• If Reader, click on summarize button:
	  • Reader interaction with summarize text section.
	• If Reader Click on rating and review button:
	  • Readers give stars for ratings.
	  • Reader gives comment for reviewing.
	  • System saves rating and review in database and Update recommendation system.

Post-conditions/Response	• The system saves the notes in the database. • The system saves the markup of progress of reading books in the database. • The system saves the Reader rating and review of book in the database.

Use Case 4: Reader View Notification:

AI Library Book System: Reader View Notification	
Actor	Reader
Description	The Reader view notification.
Pre-conditions	• Reader Logged in system. • Reader clicks on notification button.
Sequence/Flow of Events	• Reader navigates the notification from system.
Post-conditions/Response	The notification store in database as seen.

Use Case 5: Reader Send Feedback for system:

AI Library Book System: Reader Send Feedback for system	
Actor	Reader, manager
Description	The Reader send feedback about system.
Pre-conditions	• Reader Logged in system. • Reader clicks on feedback button.
Sequence/Flow of Events	• Reader writes comment to give feedback about system
Post-conditions/Response	• store feedback on database. • Send notification for manager with feedback.

Use Case 6: Reader Upload Book:

AI Library Book System: Reader Upload Book	
Actor	Reader, manager
Description	The Reader upload a new book on the system.
Pre-conditions	• Reader Logged in system. • Reader clicks on Upload book button.
Sequence/Flow of Events	• Reader Load the book from its local storage. • System validates pdf format and file size. • System sends book information to manager to accept/reject upload it in system.

Post-conditions/Response	• If manager accept upload the book:
	  • The system stores it on a database.
	  • The system sends notification to Reader that the manager accepts to upload book.
	• If manager reject upload the book:
	  • The system send notification to Reader that the manager rejects to upload the book with description.

Use Case 7: Reader Update its information:

AI Library Book System: Reader Update its information	
Actor	Reader
Description	The Reader update its personal information on system.
Pre-conditions	• Reader Logged in system. • Reader clicks on Profile Information button. • Readers click on the Update account information button.
Sequence/Flow of Events	• Reader clicks on Update account information button and then Reader shall Update email, name, password and preferences.
Post-conditions/Response	• Store the updated information on database.

Use Case 8: Reader Delete its account:

AI Library Book System: Reader Delete its account	
Actor	Reader
Description	The Reader delete its Account on system.
Pre-conditions	• Reader Logged in system. • Reader clicks on Profile Information button.
Sequence/Flow of Events	• Reader clicks on delete account button, then the account deleted
Post-conditions/Response	• The Reader profile deleted from database.

Use Case 9: Manager Maintain System:

AI Library Book System: Manager Maintain system	
Actor	Manager
Description	The Manager control system.
Pre-conditions	• Manager Logged in Admin Panel.

Sequence/Flow of Events	• If Manager click on book section:
	  • Can insert, delete, update book.
	• If Manager click on account section:
	  • Can insert, delete, update account.
	• If Manager click on upload book request:
	  • Can accept/reject upload book.

Post-conditions/Response	• If the manager maintains books, then it's action store on database.
	• If the manager maintains an account, then its action is stored on database.
	• If manager accept upload the book:
	  • The system stores it on a database.
	  • The system send notification to reader that manager accept to upload book.
	• If manager reject upload the book:
	  • The system send notification to reader that the manager rejects to upload the book with description.

Use Case 10: Review Notification about system:

AI Library Book System: Review Notification about system	
Actor	Manager
Description	The Manager review notification about system.
Pre-conditions	• Manager Logged in Admin Panel. • Manager clicks on notification button.
Sequence/Flow of Events	• Manager reviews feedback about the system on feedback section. • Manager review upload book request.

Post-conditions/Response	• Review feedback:
	  • The feedback notification is stored on database as seen.
	• Upload book request:
	  • Upload book request notification store on database as seen.

4.4 Use Case Diagram

A use case diagram shows the relationships between actors (User, Manager, Google API) and how the system operates.

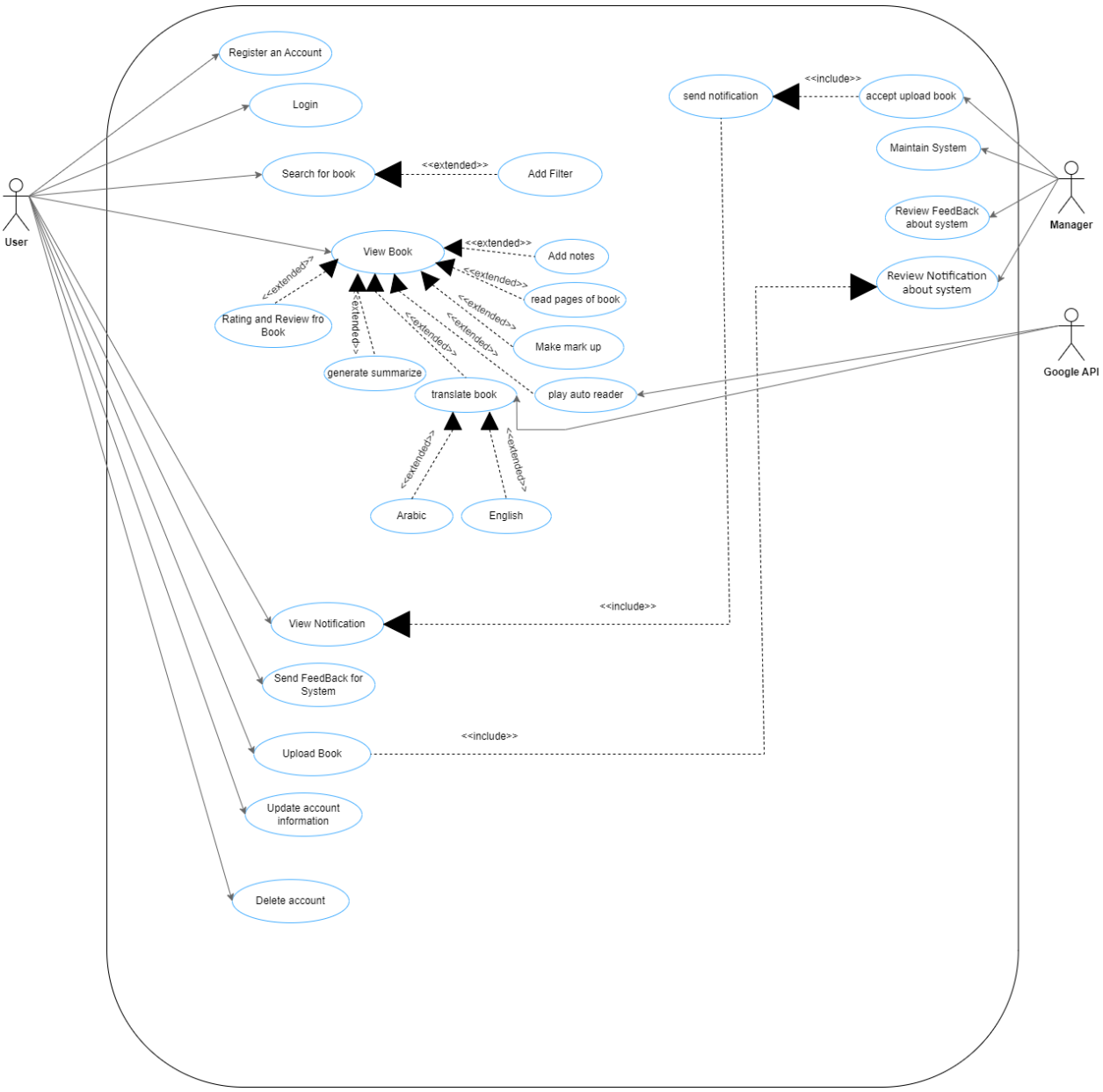


Figure 6:use case diagram

4.6 Class Diagram

Class diagram describes the structure of a system by showing the system's classes, their attributes, and the relationships among objects and how they interact with each other.

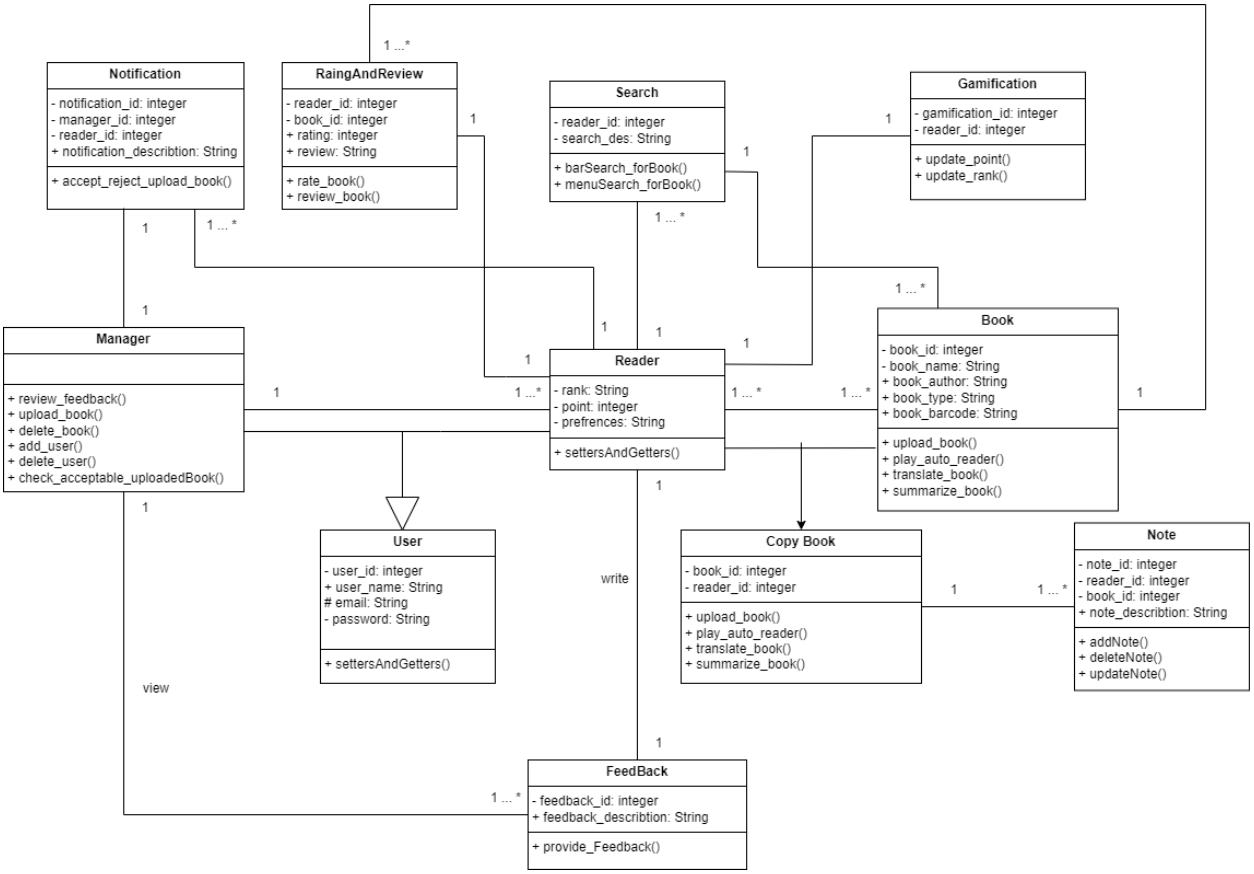


Figure 7: class diagram

4.6 Sequence diagrams

A sequence diagram that represents interactions among objects or components in a system and the order in which these interactions occur. It is used to model the how of messages, events, and actions between objects in a system.

4.6.1 View Book:

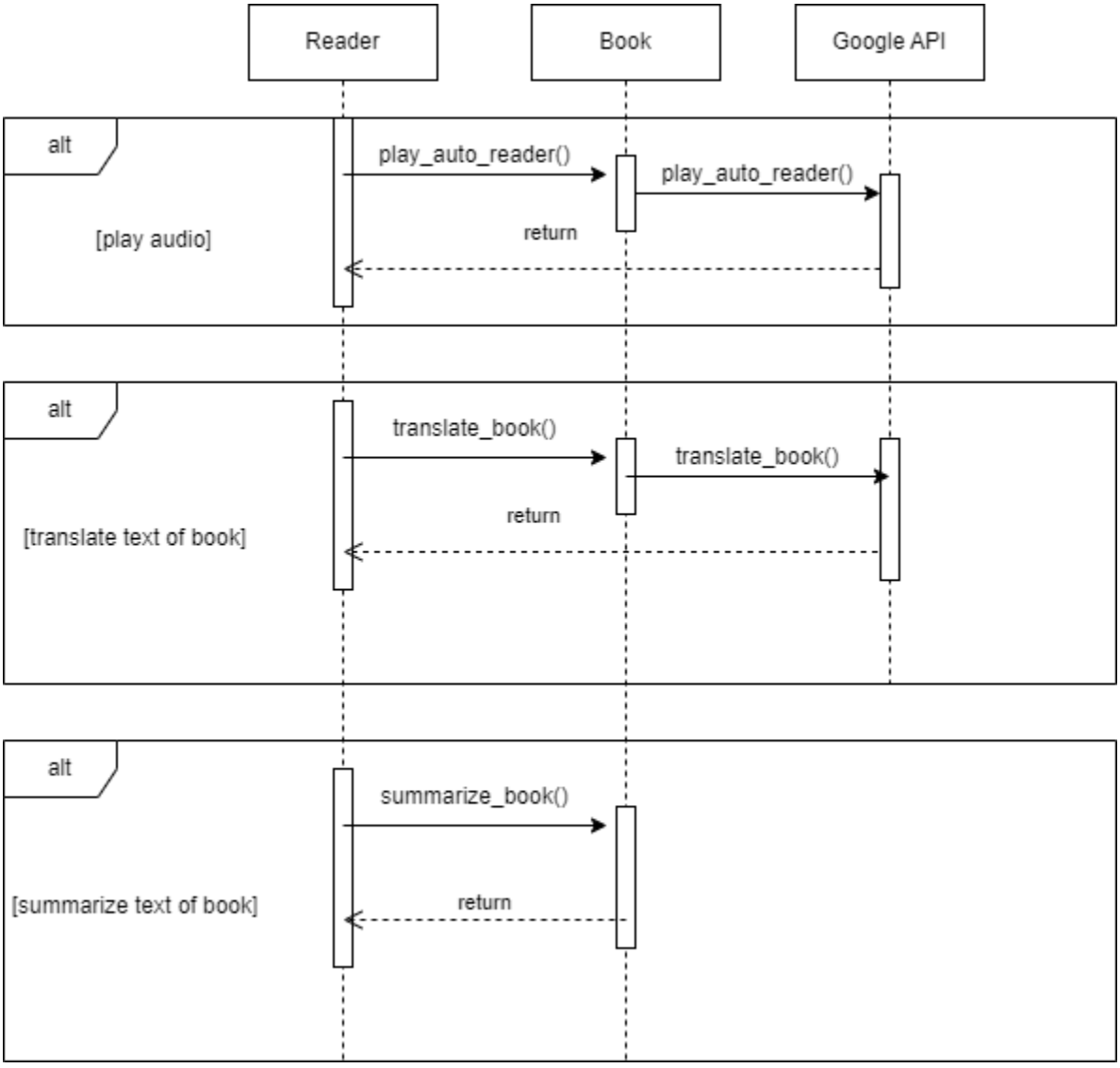


Figure 8:Sequence diagrams for View Book

4.6.2 Upload Book:

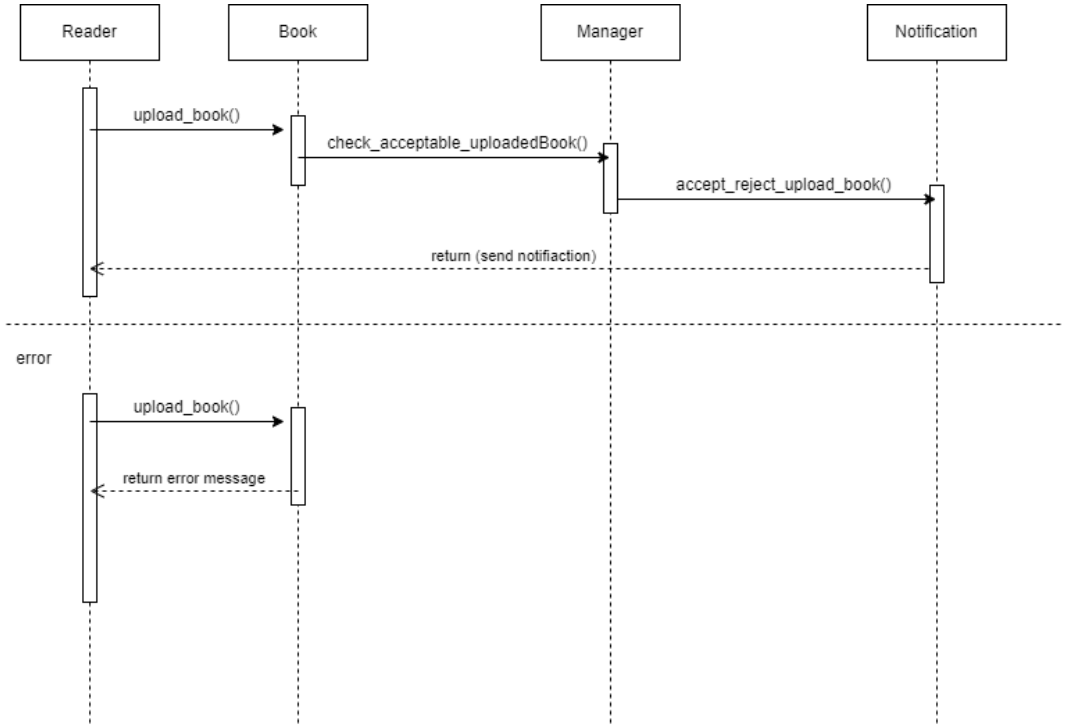


Figure 9:Sequence diagrams for Upload Book

4.6.3 Feedback for system:

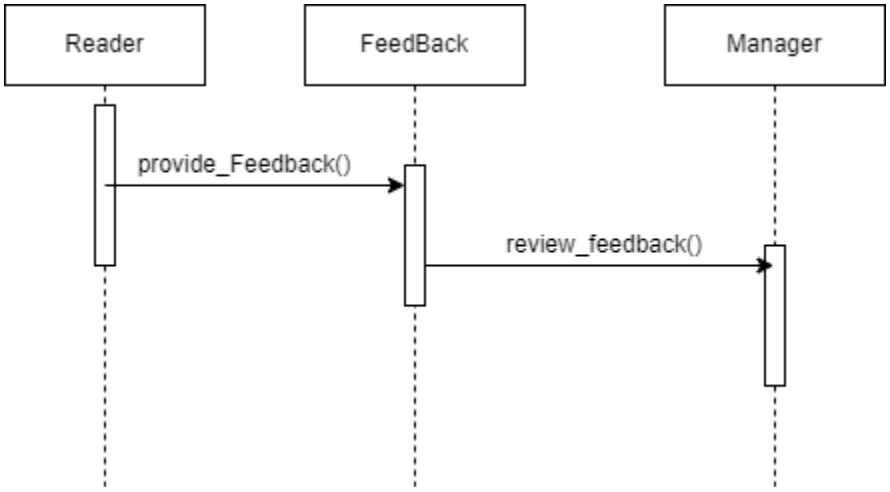


Figure 10:Sequence diagrams for Feedback for system

4.6.4 Search Book:

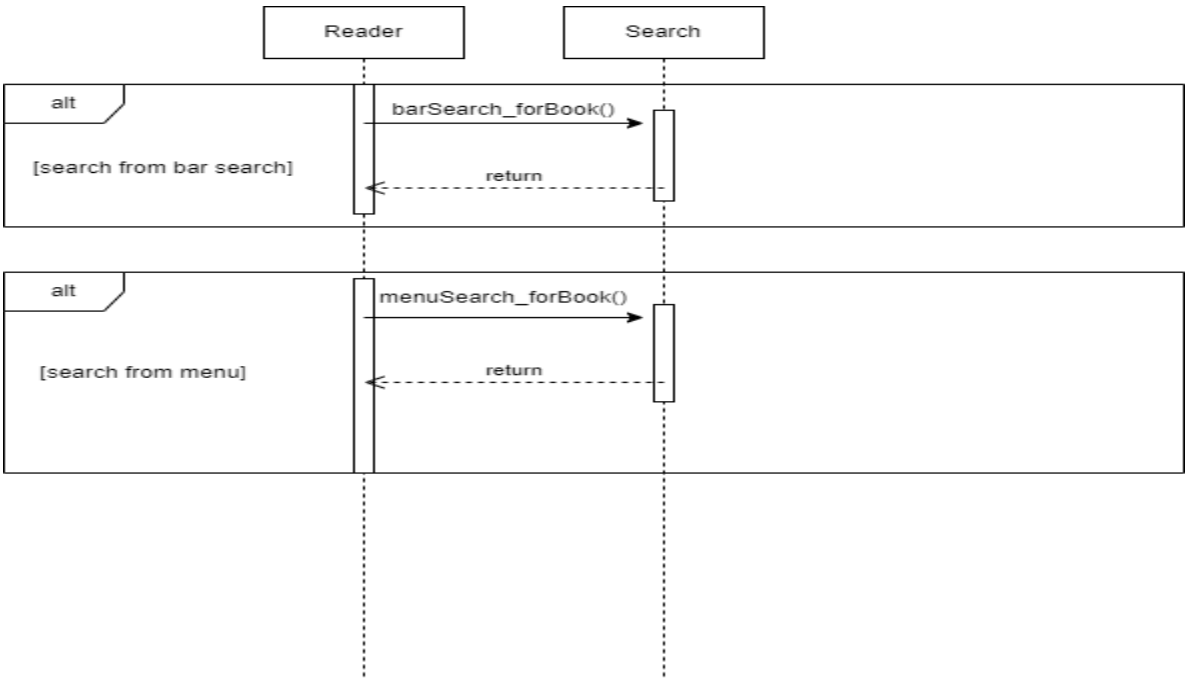


Figure 11:Sequence diagrams for Search Book

4.6.5 Rating and Review Book:

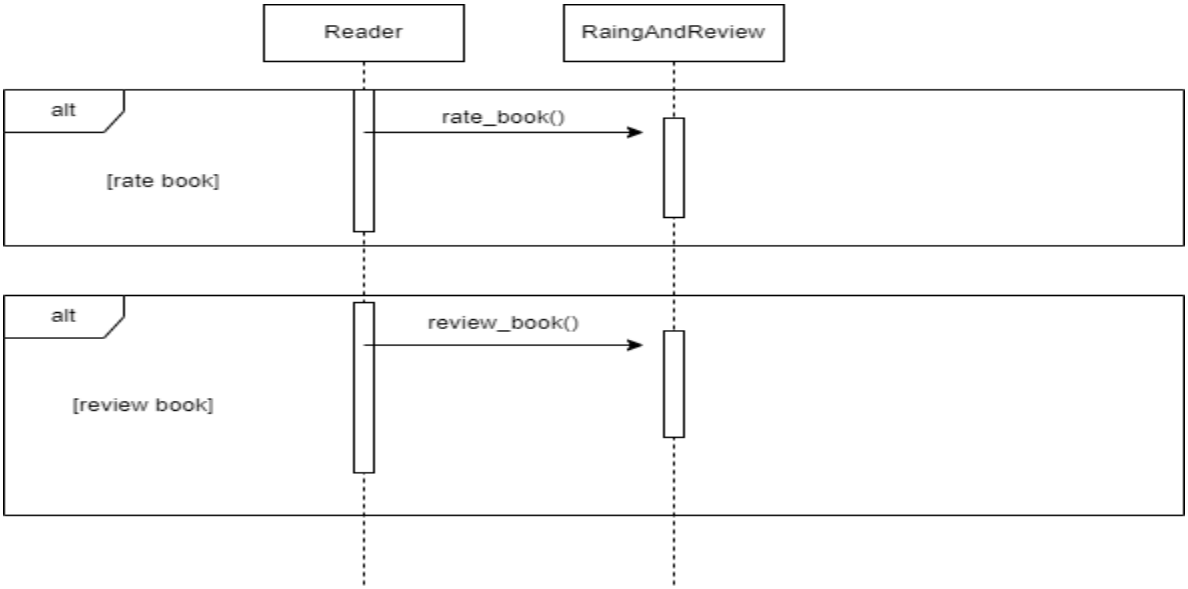


Figure 12: Sequence diagrams for Rating and Review Book

4.8 Activity diagram

An activity diagram is a type of UML (Unified Modeling Language) diagram that is used to model the workflow of a system or a process and to represent the control flow from one activity to another.

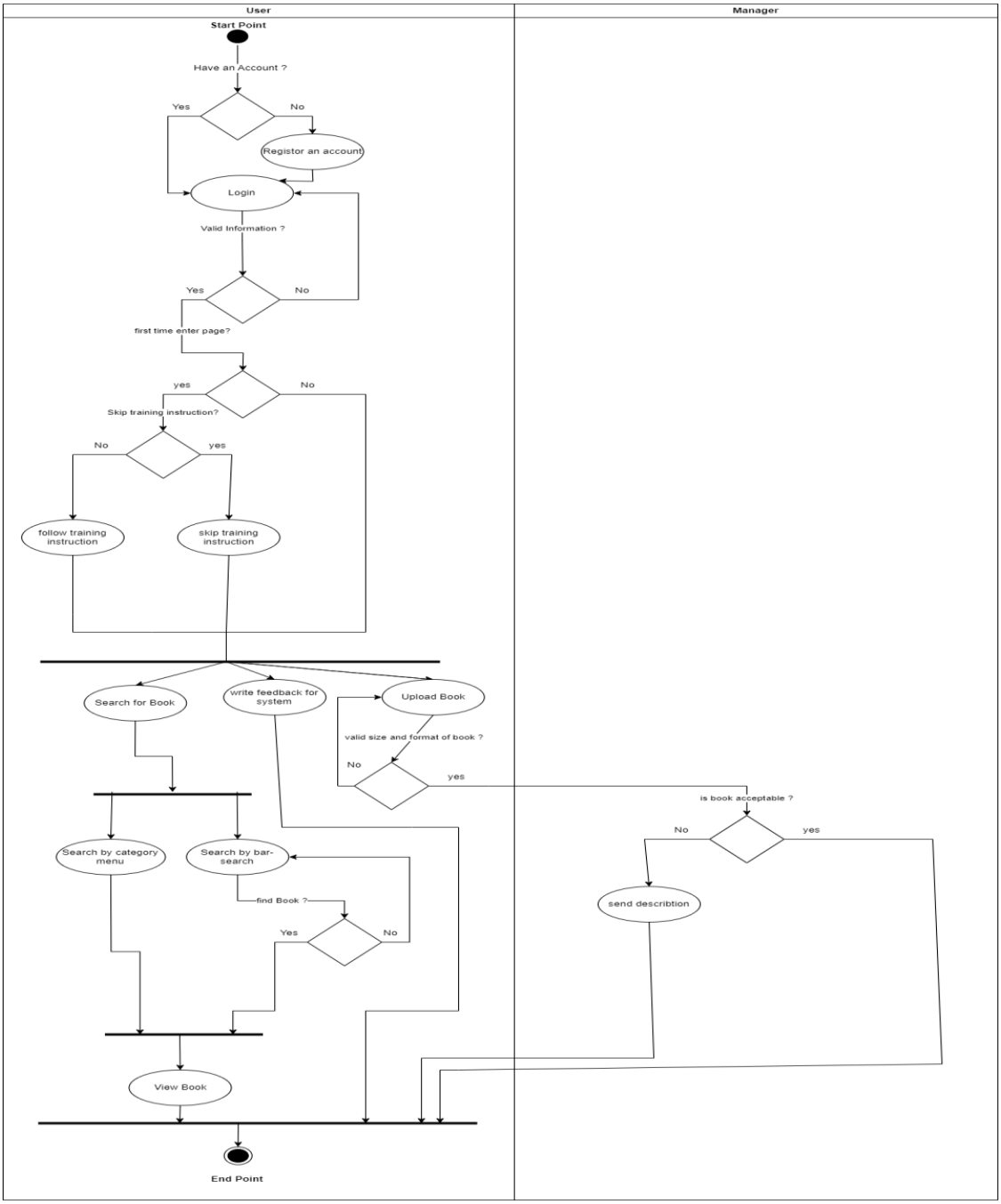


Figure 13:Activity diagram.

4.8 State chart diagrams

A state chart diagram describes the behavior of an object as a series of states and describes the transitions among these states.

4.8.1 Feedback for book:

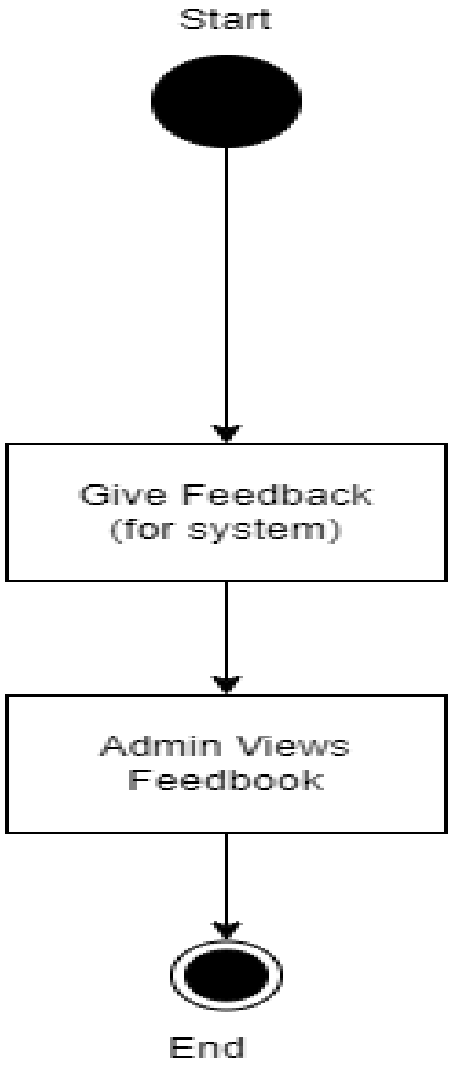


Figure 14: State chart diagrams for Feedback

4.8.2 Search Bar:

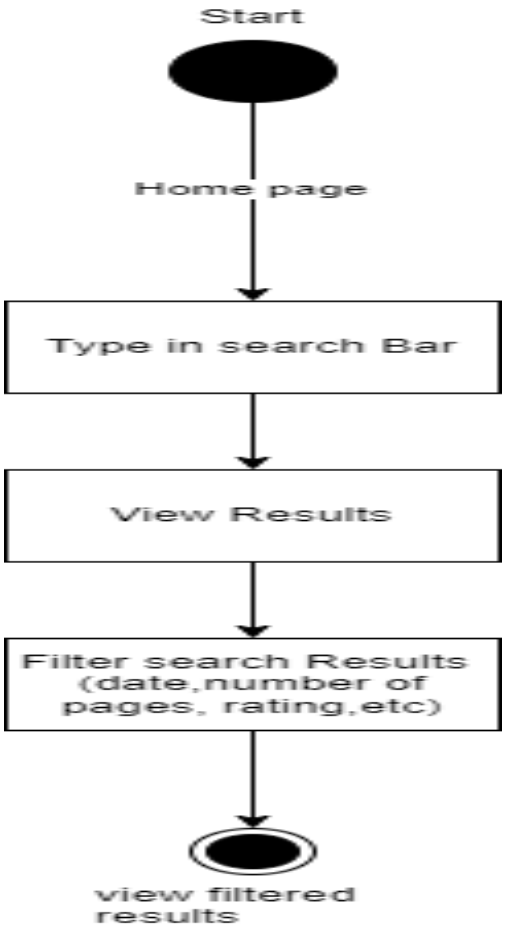


Figure 15:State chart diagrams for Search Bar

4.8.3 Gamification:

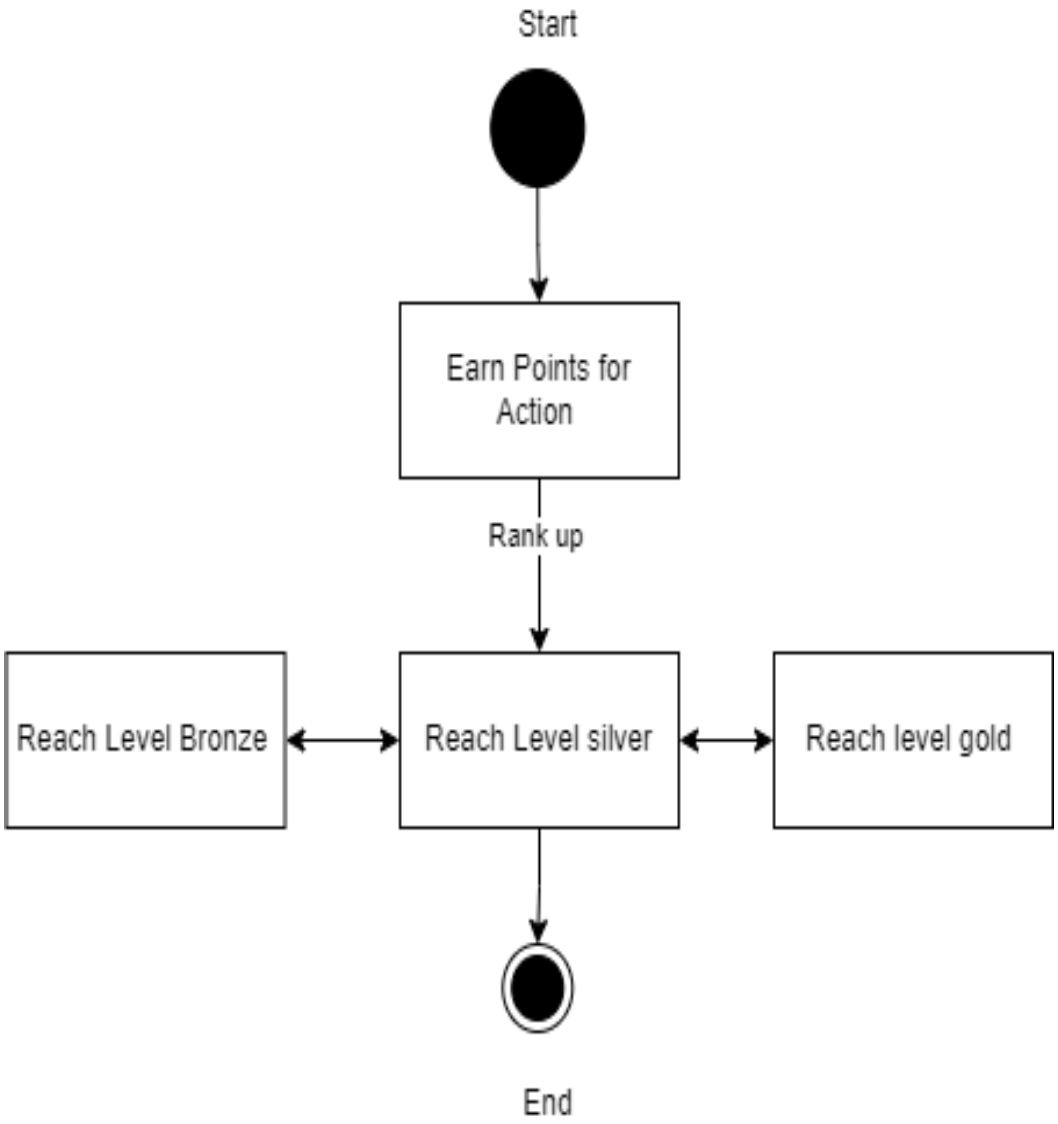


Figure 16: State chart diagrams for gamification

4.8.4 View Book:

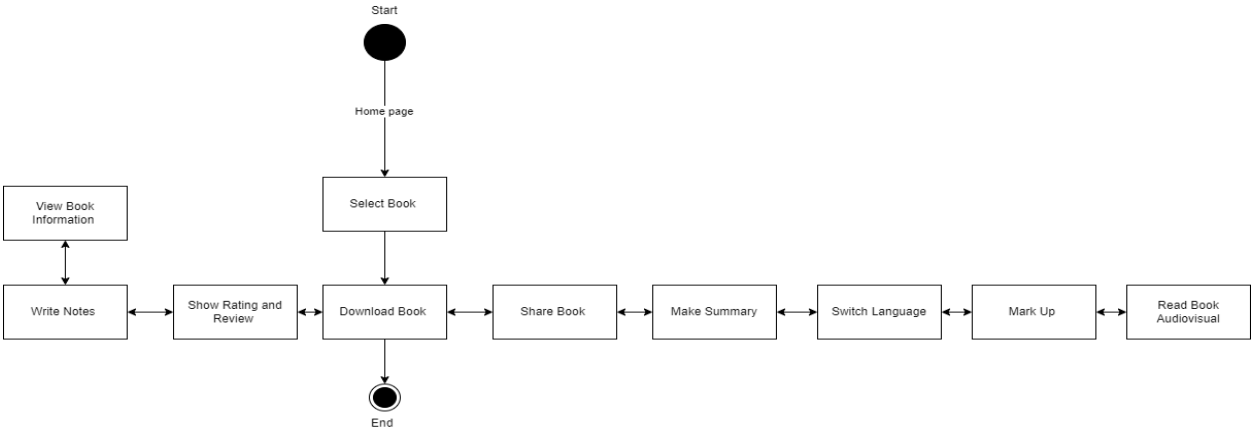


Figure 17:State chart diagrams for view book

4.8.5 Login Diagram:

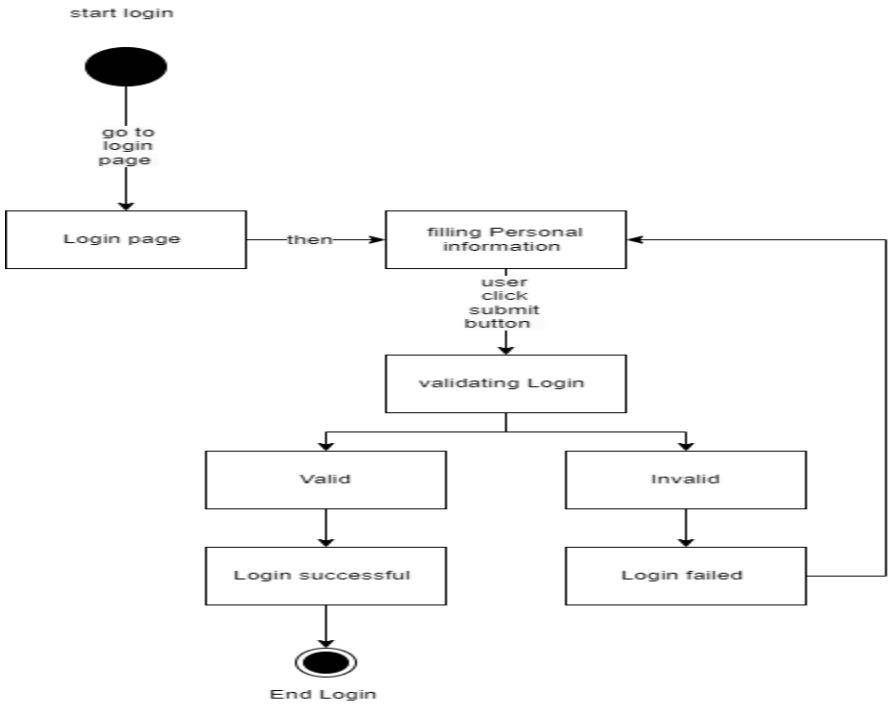


Figure 18:State chart diagrams for login

4.8.6 maintain Account:

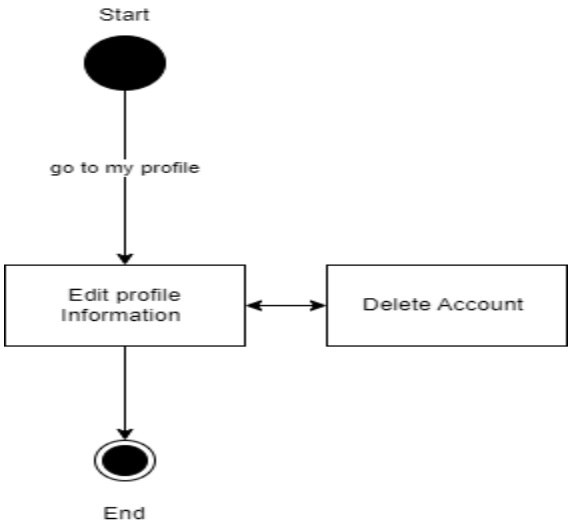


Figure 19:State chart diagrams for maintain account

4.8.7 notification:

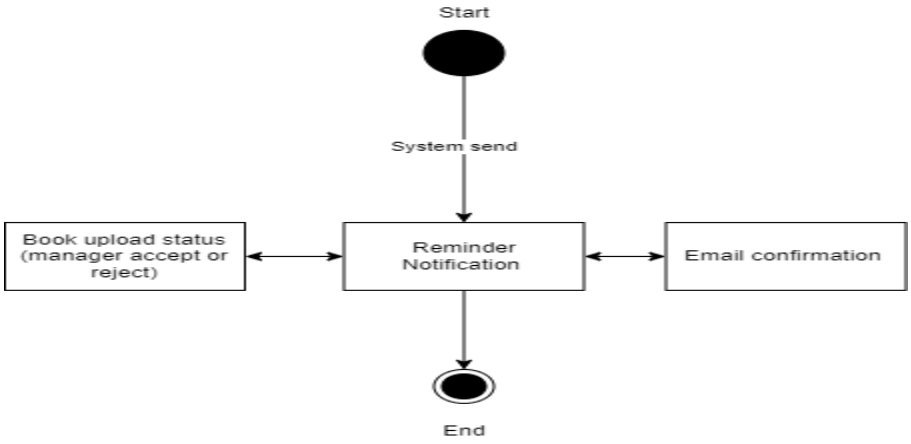


Figure 20:State chart diagrams for notification

4.8.8 Preferences Diagram:

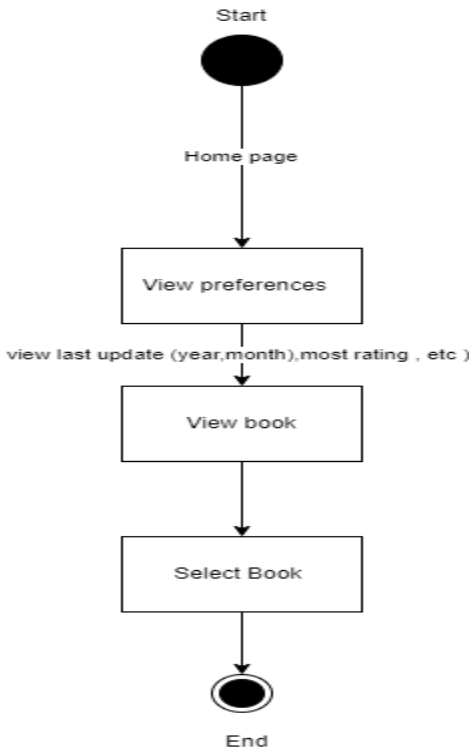


Figure 21:State chart diagrams Preferences

4.8.9 Rating and Review:

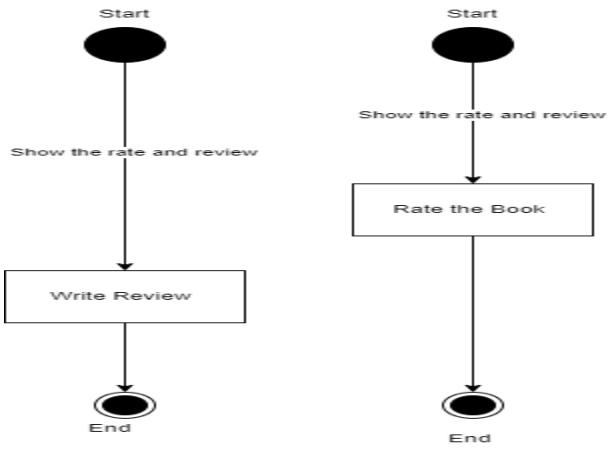


Figure 22:Rating and Review Book for rating and review

4.8.10 Registration Diagram:

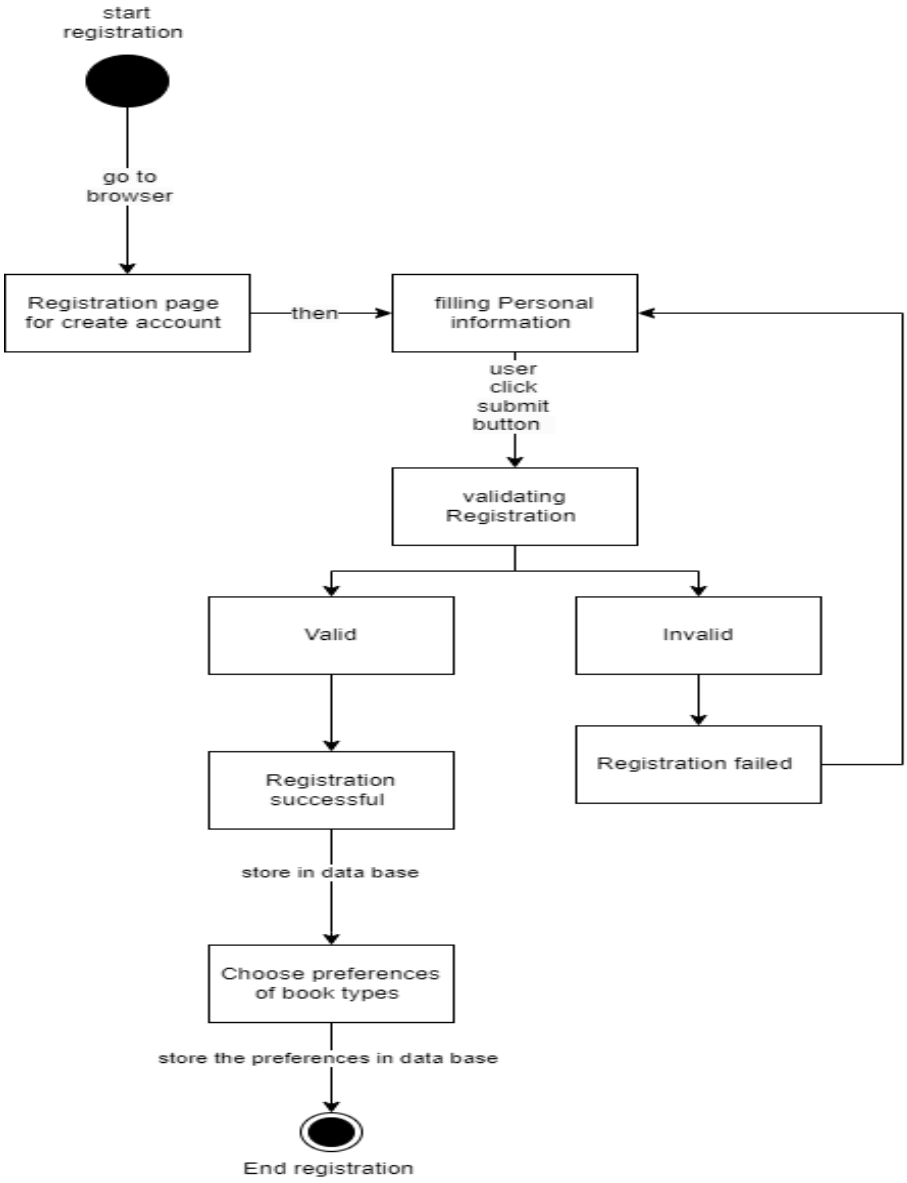


Figure 23:State chart diagrams for Registration

4.8.11 Training simulation:

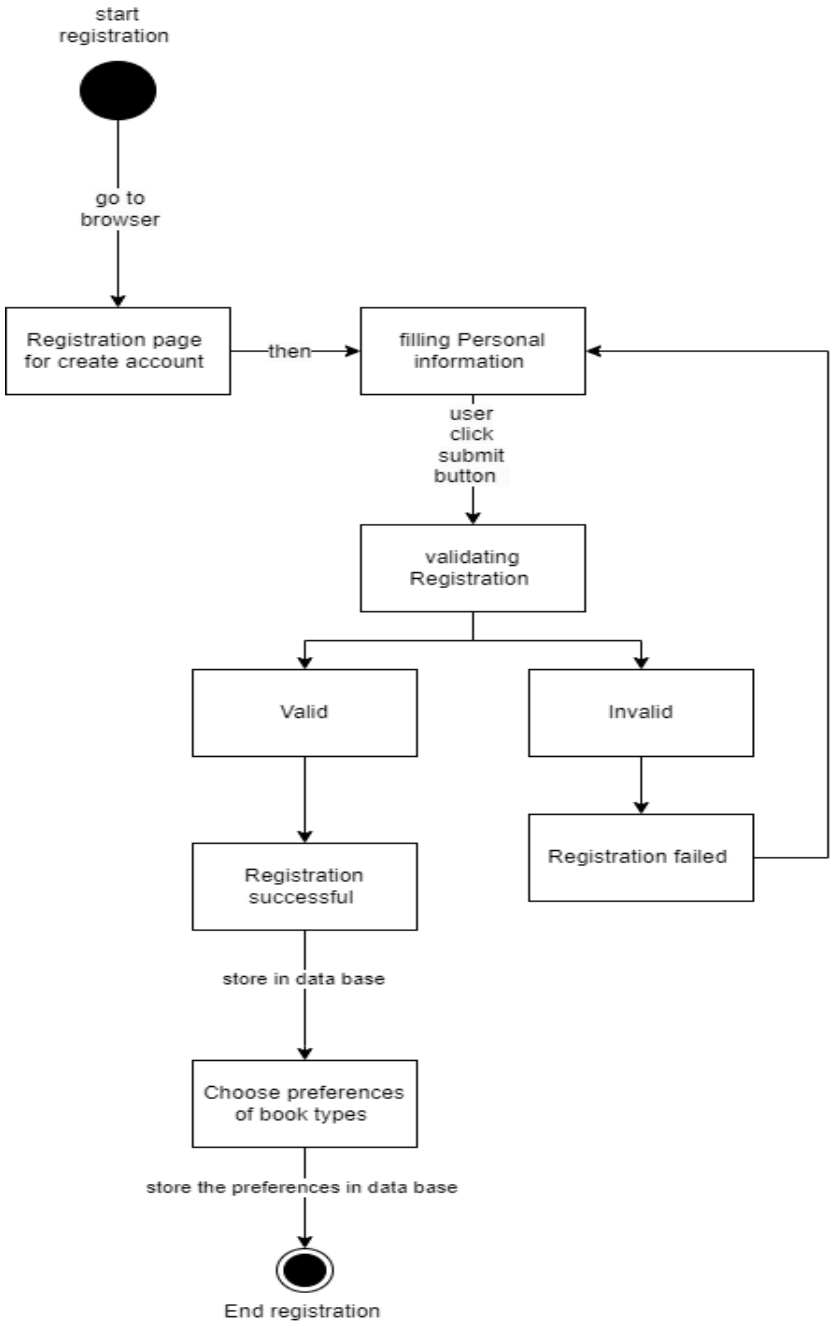


Figure 24:State chart diagrams for Training simulation

4.8.12 Upload Book:

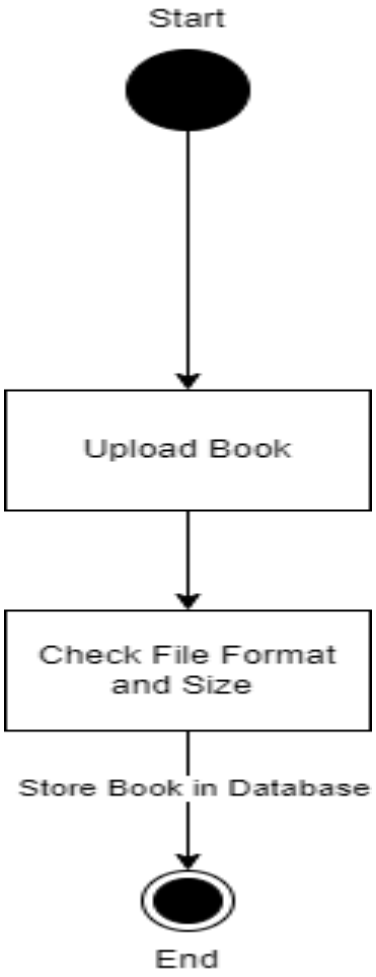


Figure 25: State chart diagrams for Upload Book

Chapter 5: System Design

This chapter shows Design Goals, Deployment Diagram and System Architecture.

5.1 Design Goals

System design is a process of defining the architecture, components, modules, interfaces, and data for a system to satisfy specified requirements: high-level system design of a web application The project evolved into a flexible web application that operates on various websites. This was achieved by selecting optimal styles for the front end and the back end. as well as the best database for storing user objects. During the design process, the system needs and features were closely examined to ensure alignment with the project's goals. Inspiration was drawn from similar apps to identify potential features that could enhance the quality of the service. The system design process involved creating a prototype for the web application, designing the database and classes, implementing the model, coding the frontend, and connecting the frontend with the model in the backend. The project was then tested and deployed.

High cohesion	We shall put the classes and the methods that have most interact together and have dependency on a one component.
Low coupling	We shall put the classes and the methods that less interaction together in a separated component.
Ease of learning and user friendliness	Our website is created to be easy to use and use a little of clicking, and it will have training simulation.

5.2 Deployment Diagram

Deployment diagrams are used to model the physical components of a system.

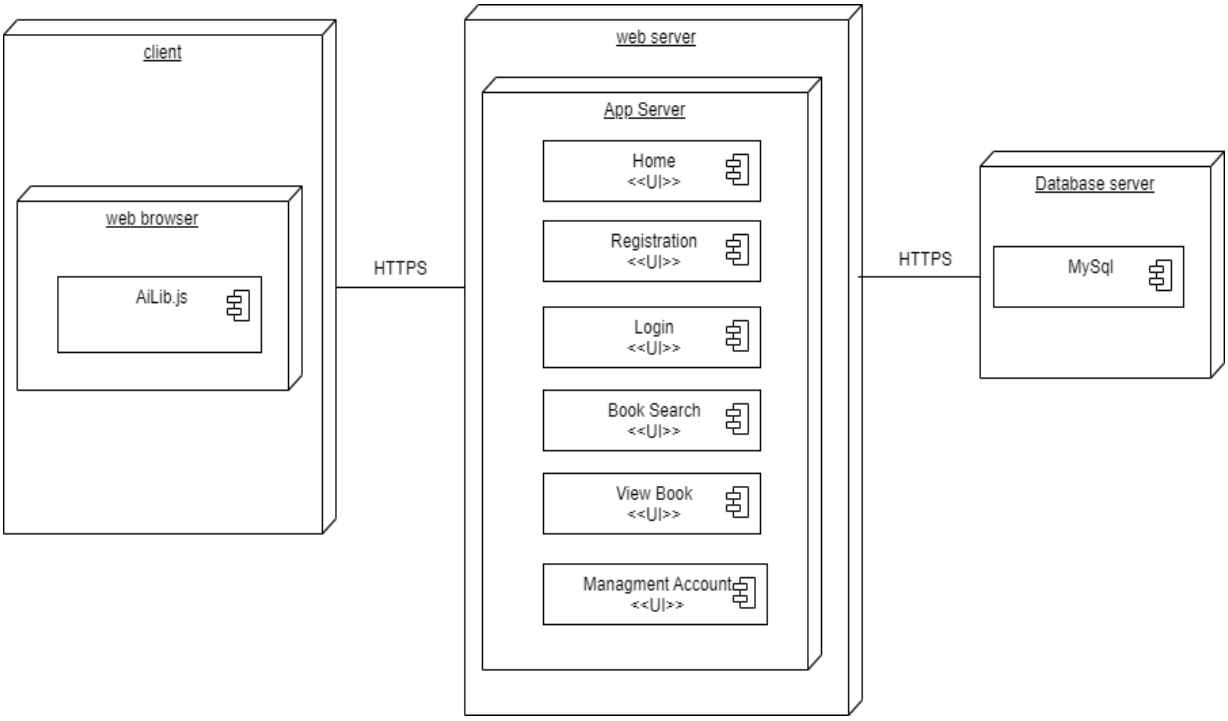


Figure 26:Deployment Diagram

5.3 System Architecture

System architecture is the overall structure of a system that clearly defines is components and relationships in the system.

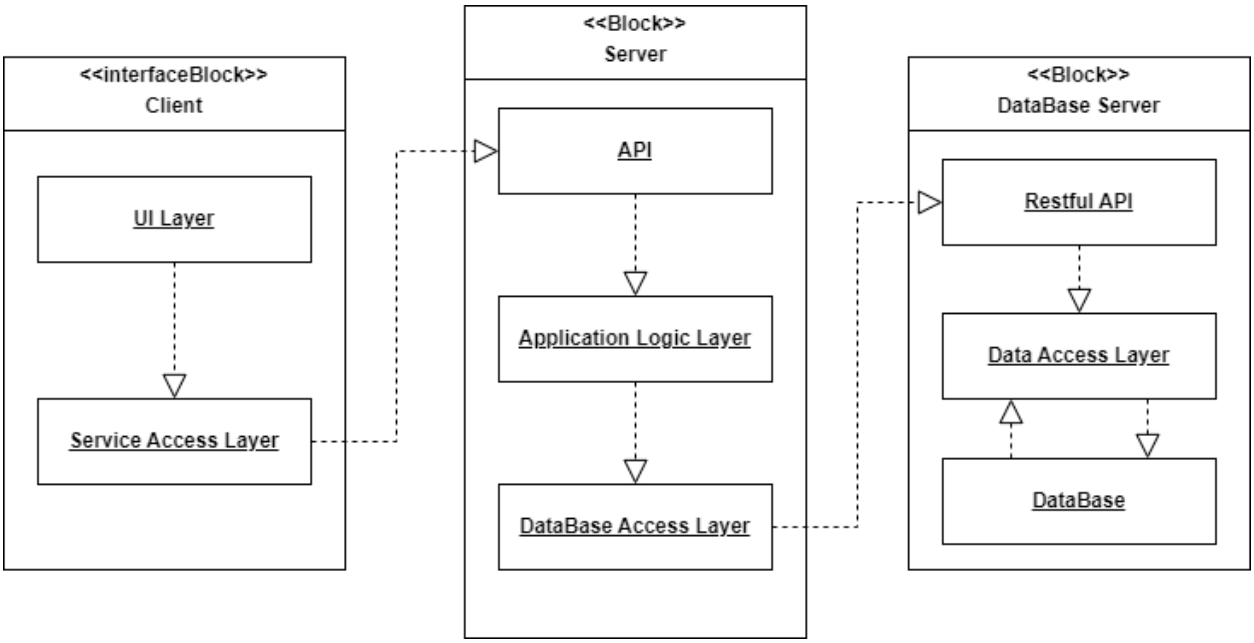


Figure 27:System Architecture

Chapter 6: Implementation

This chapter shows implementation of application.

6.1 Overview:

We provide a detailed look at the procedures we followed to implement the SmartLib Project. We utilized a variety of technologies and methodologies to create a flexible and scalable application. Here are the main

components of our implementation:

• Frontend: We used HTML, CSS, JavaScript, and Django full-stack Framework.

• Backend: We used Python with the Django full-stack framework.

• Database: We used PostgreSQL.

• JWT Authentication and Authorization: We implemented JWT (JSON Web Token) for secure authentication and Authorization.

This mix of technologies allowed us to build a robust and efficient system.

• Translation, text-to-speech Api's

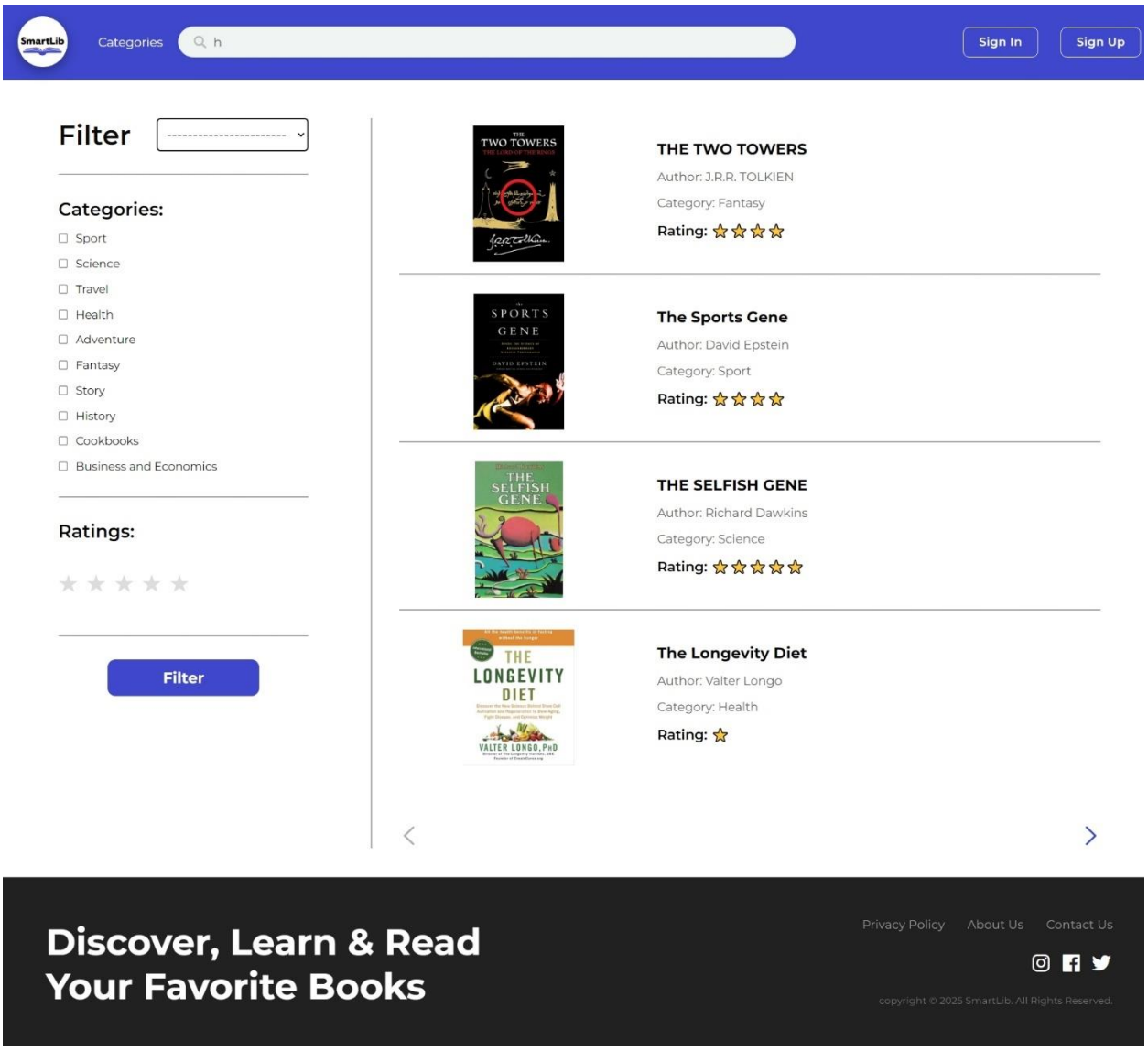
• Text-summarization algorithm

6.2 Screenshot for actual application:

6.2.1: Guest Page

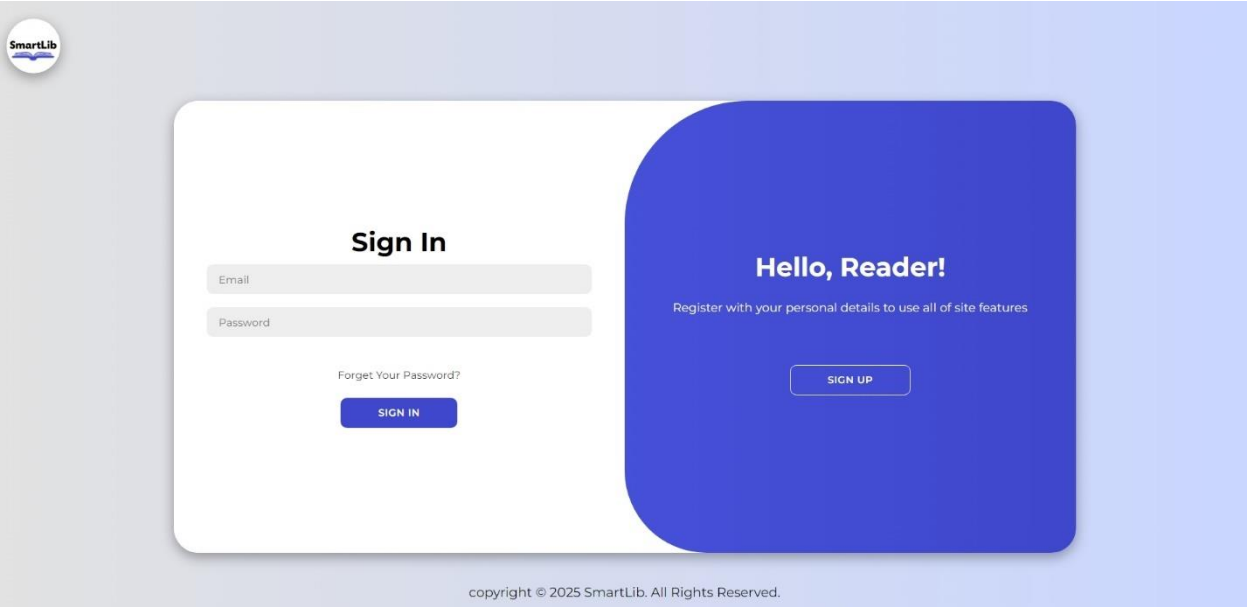
This is the main guest page which the guest can explore the library and can view the books without login in the system

6.2.2: Guest Search Book Page



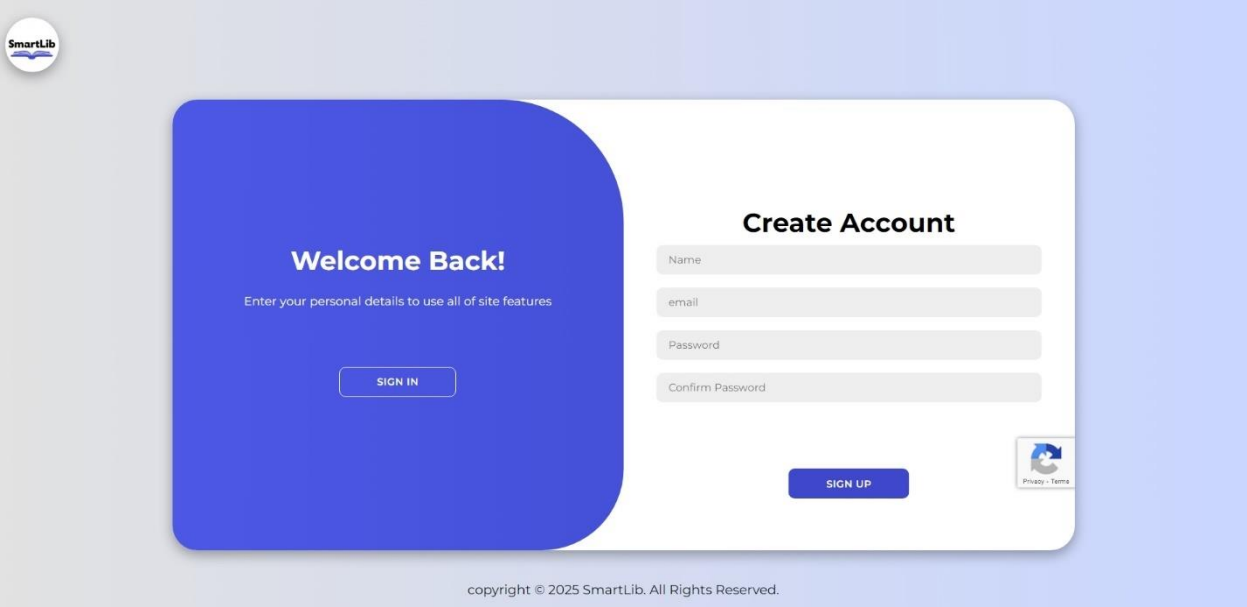
The guest can search for specific books based on multiple options.

6.2.3: Login Page



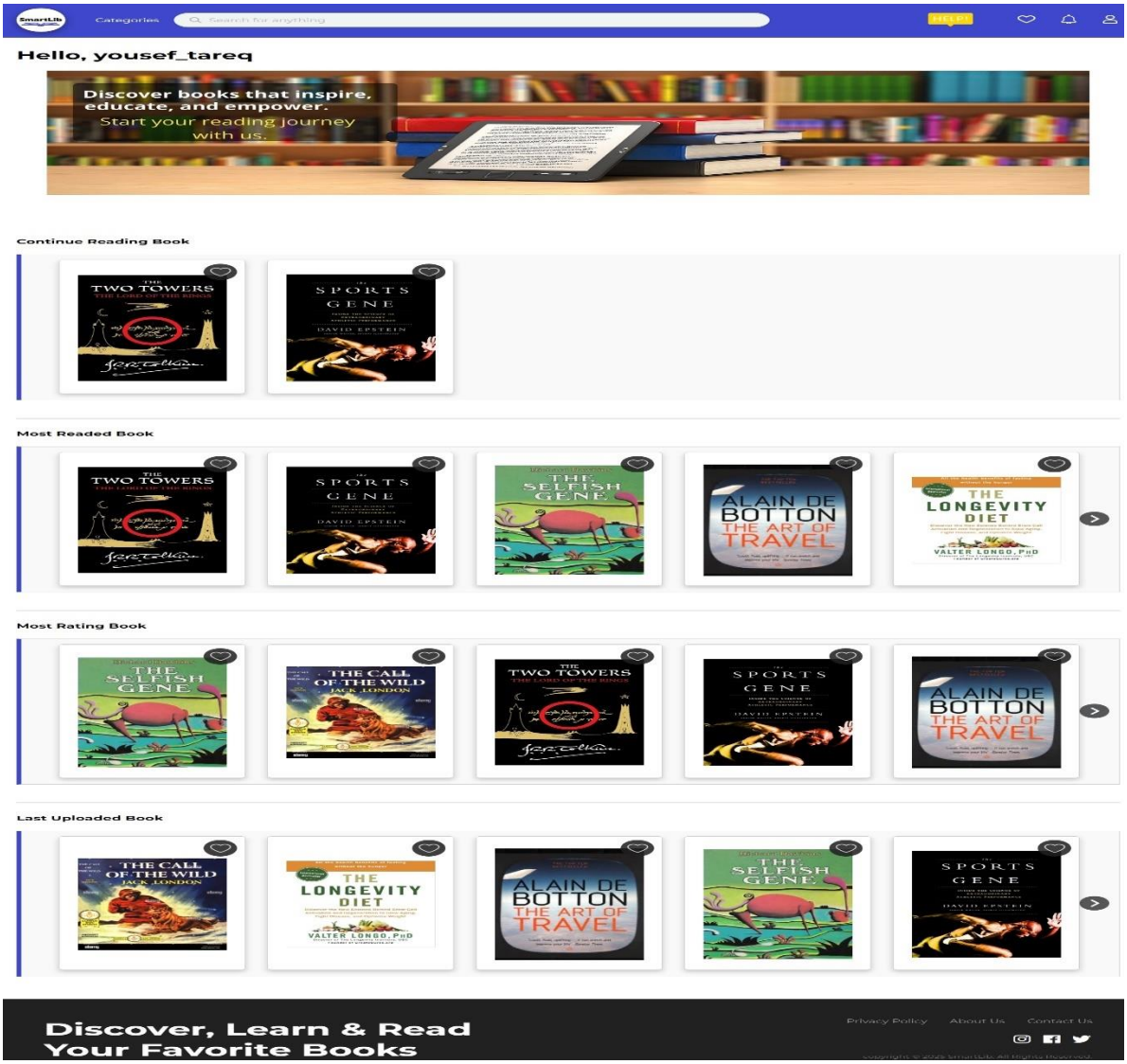
This Page lets the user to access the website by his email and password.

6.2.4: Register Page



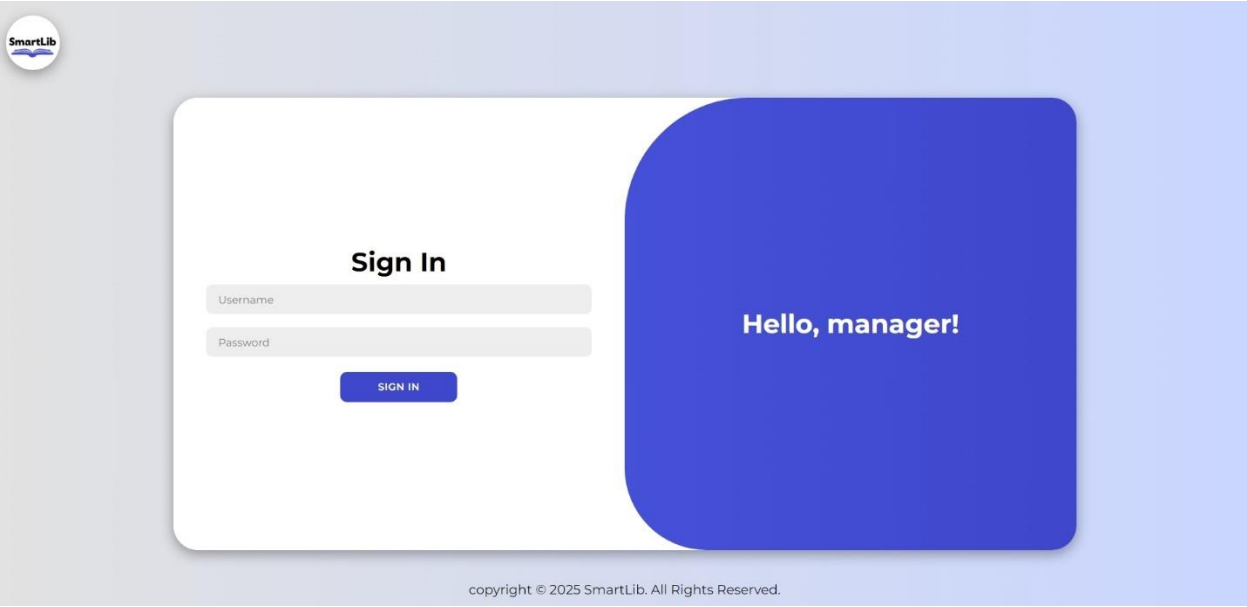
This page the user can register to our website System to be able to access the website.

6.2.5: User Home Page



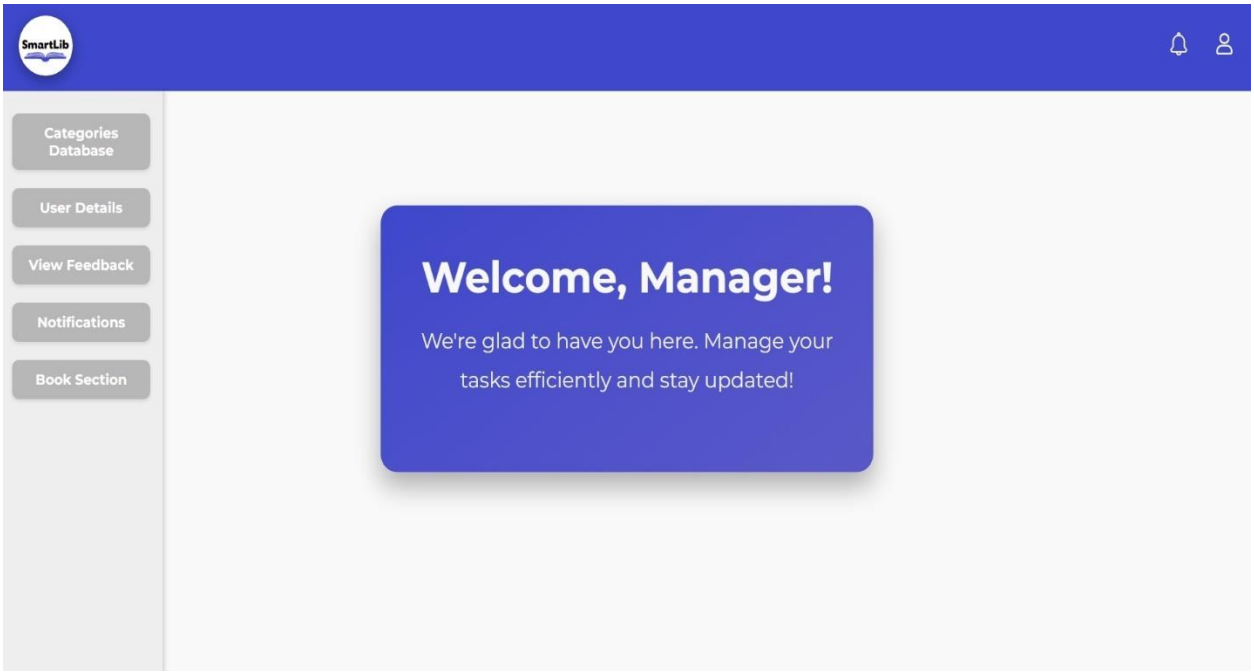
This is the Home page for User in this page the user can navigate it counting reading book, most readed Book, most rating book and last uploaded book, and add book to wish list, see notification, send feedback and make search for book.

6.2.6: Login for Manager (Admin)



This page the manager can login into the manager panel

6.2.7: Manager Home Page



This page after the manager login the system, can the manager access categories making search, delete and add and view the category in system, also can the manager view Readers make update, add, delete and search, Also can view book and make update, add and delete for books, also can view the notification that come from reader to make accepted or rejected for this notification, Finally, the manager can view all the system's comments that comes from readers.

6.3 Algorithms used:

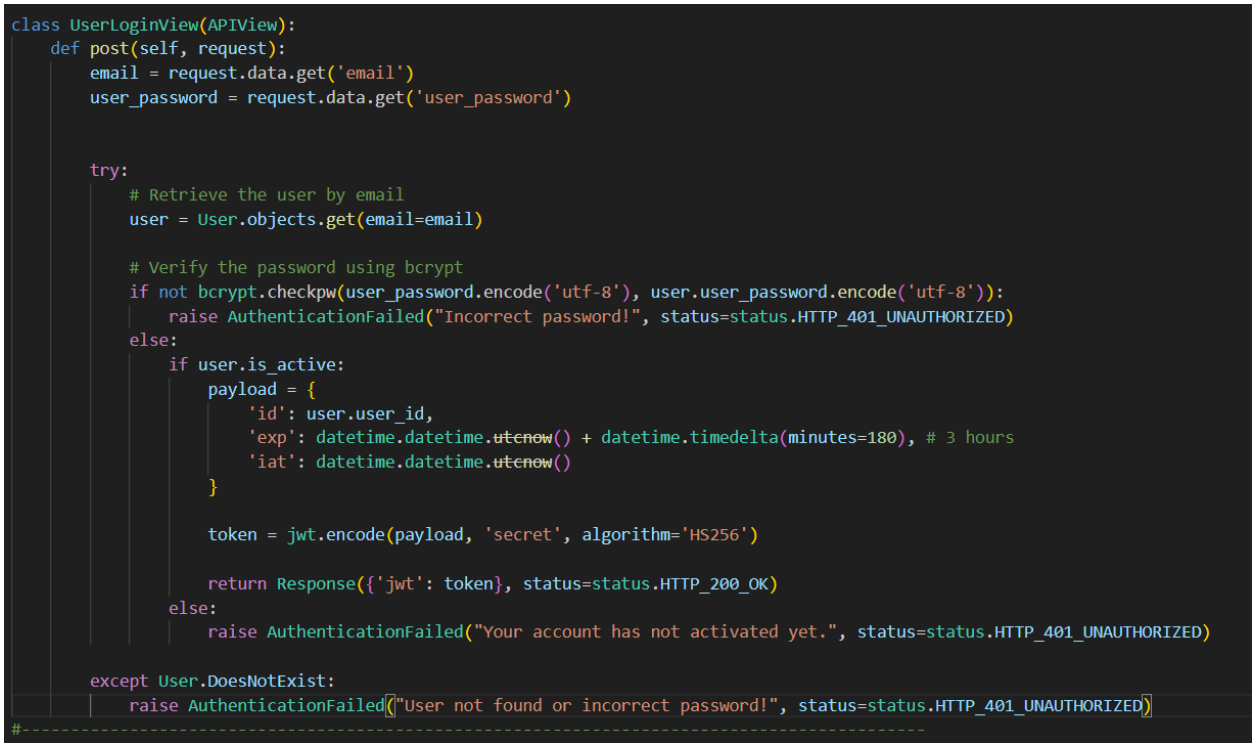
- **Text rank:**

the Text Rank algorithm used to understand the context and semantics of the text. Once the text is obtained, it split that text into individual sentences, and if the sentences have stop words or stemming words, they are removed. Then, feature extraction represents each sentence using data structure like vector representation, and the similarity between sentence vectors is calculated and stored in a matrix. The similarity matrix, which includes verbs, objects, and adjectives, is converted into a graph. Then, we calculate the sentence rank, that gives ranked sentences from which we get top-ranked sentences for the final summary.

```
import json
from django.http import JsonResponse
from django.views.decorators.csrf import csrf_exempt
from sumy.parsers.plaintext import PlaintextParser
from sumy.nlp.tokenizers import Tokenizer
from sumy.summarizers.text_rank import TextRankSummarizer
def summarize_text(request, book_id):
    if request.method == 'POST':
        try:
            data = json.loads(request.body)
            selected_text = data.get('text', '').strip()

            if not selected_text:
                return JsonResponse({'error': 'No text provided for summarization.'}, status=400)

            parser = PlaintextParser.from_string(selected_text, Tokenizer("english"))
            summarizer = TextRankSummarizer()

            summary_sentences = summarizer(parser.document, 3)

            summary = ' '.join([str(sentence) for sentence in summary_sentences])

            return JsonResponse({'result': summary})
        except Exception as e:
            return JsonResponse({'error': str(e)}, status=500)

    return JsonResponse({'error': 'Invalid request method. Use POST.'}, status=400)
```

- **Jwt:**

In login process we set jwt payload (the user id, expiration date, current date) then encode the payload.



```python
class UserLoginView(APIView):
    def post(self, request):
        email = request.data.get('email')
        user_password = request.data.get('user_password')


        try:
            # Retrieve the user by email
            user = User.objects.get(email=email)

            # Verify the password using bcrypt
            if not bcrypt.checkpw(user_password.encode('utf-8'), user.user_password.encode('utf-8')):
                raise AuthenticationFailed("Incorrect password!", status=status.HTTP_401_UNAUTHORIZED)
            else:
                if user.is_active:
                    payload = {
                        'id': user.user_id,
                        'exp': datetime.datetime.utcnow() + datetime.timedelta(minutes=180), # 3 hours
                        'iat': datetime.datetime.utcnow()
                    }

                    token = jwt.encode(payload, 'secret', algorithm='HS256')

                    return Response({'jwt': token}, status=status.HTTP_200_OK)
                else:
                    raise AuthenticationFailed("Your account has not activated yet.", status=status.HTTP_401_UNAUTHORIZED)

        except User.DoesNotExist:
            raise AuthenticationFailed("User not found or incorrect password!", status=status.HTTP_401_UNAUTHORIZED)
#-----------------------------------------------------------------------------------------
```

- **Search algorithm:**

  The search engine filters the results based on multiple parameters that taken from user.

```python
#-- list searched book
class BookSearchView(APIView):
    def get(self, request):
        # Get parameters from the request
        search_query = request.query_params.get('search', '').strip()
        sort_by = request.query_params.get('sort_by', '')
        category_ids = request.query_params.get('category', None)
        min_rating = request.query_params.get('min_rating', None)

        # Base queryset, only filter if search query is provided
        books = Book.objects.filter(status=Book.Status.ACCEPTED)
        if search_query:
            books = books.filter(book_name__icontains=search_query)

        # Filter by multiple categories if provided
        if category_ids:
            category_id_list = category_ids.split(',')
            books = books.filter(category__category_id__in=category_id_list)

        # Filter by minimum rating if provided
        if min_rating:
            books = books.filter(book_rating_avg__gte=min_rating).order_by('book_rating_avg')

        # Apply sorting
        if sort_by == 'reviewed':
            books = books.order_by('-book_reading_counter')
        elif sort_by == 'favourite':
            books = books.order_by('-book_favourite_counter')
        elif sort_by == 'newest':
            books = books.order_by('-book_uploaded_date')
```

Chapter 7: System Testing

7.1 Overview:

A testing phase is vital in the software development process since it guarantees the software product is free of any bugs, meets the user requirements and expected functionality. It removes software errors by executing the program to find missing requirements, errors or gaps. This means it gives every software developer the opportunity to rectify the program issues prior to providing the product to the end-users. Tests increase the performance and quality of the software and guarantees that the software meets the user and business requirements.

Software testing strategies can include Unit testing, Integration testing, System testing, and lastly, UAT (User acceptance testing). This means that each of them has a specific function or job that they do in the testing process as a whole:

Unit Testing allows the testers to focus on solving the individual parts or block of the application, thus checking that they work correctly on their own.

Integration Testing is performed to check how the different modules interact with each other and what the output is when they are brought together.

System Testing is when the complete system is tested against the set requirements.

UAT is conducted before the software gets released to ensure that it is suitable in the operational environment.

In the contemporary era, a combination of manual and automated testing methods are being used. Tools such as Selenium, JUnit and Postman increase effectiveness while.

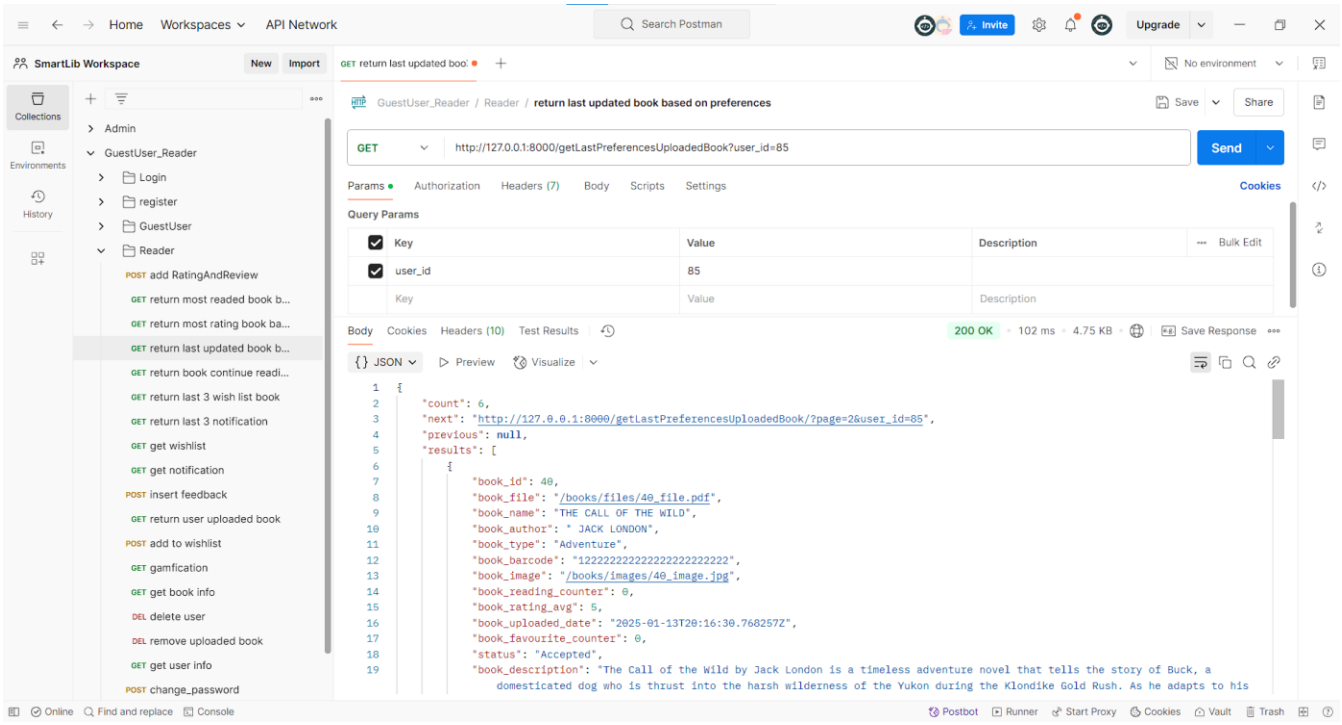
7.2 List of features to be tested:

Test Case Table

TC ID	Description	Preconditions	Test Data	Steps	Expected Result	Actual Result	Status
TC_Login_1	Verify successful login	User must have a registered account	**Valid:** Email: anas@gmail.com; Password: abs123456789 **Invalid:** Email: Null; Password: abc123456789	1. Navigate to the Login page form gust page header. 2. Enter email and password.	User navigates to the home page.	As Expected	done
TC_addCategory_2	adding new categry for library	manager must be logged in	**Valid:** New category: sport **Invalid:** New category: -category alredy exist	1.click on categoies database. 2. Enter new category	for valid: add new category sucssesfully for invalid: category alredy exist	As Expected	done
TC_user update profile_3	user update password	User must be logged in	**Valid:** current password: abc123456789 new passwod:anas123456789 **Invalidcurrent password: bc123456789 new passwod:anas123456789	1. Navigate to the Profile page from dropbox in header. 2. Enter current and new password	for valid: password change sucssesfully for invalid: current password not match	As Expected	done
TC_sendFeedback_4	Verify user can add product to cart after update	User must be logged in	**Valid:** send feedback:it is a great system **Invalid**send feedback:null	1. Navigate to the feedback page from dropbox in header 2. enter feedback message	for valid: feedback send secssusflly for invalid: please fill the message.	As Expected	done

TC_send rating and review_5	send review and rating for book	user must be logged in	**Valid:** rating: 3 stars review: it's a great book **Invalid** rating: 3 stars review: null	1. click on book 2. enter rating and review	for valid: rating and review send secssusflly for invalid: please fill the message.	As Expected	done

TC ID	Description	Preconditions	Test Data	Steps	Expected Result	Actual Result	Status
TC_logout_6	logout form system	User must be logged in		1. Navigate to the logout button from dropbox in header 2. click on logout button	User navigates to the gust page.	As Expected	done
TC_search_7	search for book	User must be logged in	**Valid:** search: java for programmer	1.enter name of book in search field	for valid: nagivate to search page	As Expected	done
TC_user update prefrencess_8	user update his prefrencess	User must be logged in	**Valid:** new prefrencess: history **invalid** new prefrencess: null	1. Navigate to the Profile page from dropbox in header. 2. click in prefrencess 3.click in prefrencess	for valid: prefrencesschange sucssesfully for invalid: please choose the prefrencess	As Expected	done
TC_delete account_9	user delete his account	User must be logged in and have activated account		1. Navigate to the Profile page from dropbox in header. 2. click in prefrencess 3.click in delete account	nagivate to login page	As Expected	done
TC_add book to wishlist_10	user add a new book in his wishlist	user must be logged in		1. click on favarite button in book	book be in his wishlis	As Expected	done


Home
Workspaces
API Network
Search Postman
Invite
Upgrade
SmartLib Workspace
New
Import
Collections
Environments
History
Admin
GuestUser_Reader
Login
register
GuestUser
Reader
POST add RatingAndReview
GET return most readed book b...
GET return most rating book ba...
GET return last updated book b...
GET return book continue readi...
GET return last 3 wish list book
GET return last 3 notification
GET get wishlist
GET get notification
POST insert feedback
GET return user uploaded book
POST add to wishlist
GET gamfication
GET get book info
DEL delete user
DEL remove uploaded book
GET get user info
POST change_password
GET return last updated boo
No environment
GuestUser_Reader / Reader / return last updated book based on preferences
Save
Share
GET
http://127.0.0.1:8000/getLastPreferencesUploadedBook?user_id=85
Send
Params
Authorization
Headers (7)
Body
Scripts
Settings
Cookies
Query Params
Key
Value
Description
Bulk Edit
user_id
85
Body
Cookies
Headers (10)
Test Results
200 OK
102 ms
4.75 KB
Save Response
JSON
Preview
Visualize
{
"count": 6,
"next": "http://127.0.0.1:8000/getLastPreferencesUploadedBook/?page=2&user_id=85",
"previous": null,
"results": [
{
"book_id": 40,
"book_file": "/books/files/40_file.pdf",
"book_name": "THE CALL OF THE WILD",
"book_author": " JACK LONDON",
"book_type": "Adventure",
"book_barcode": "12222222222222222222222",
"book_image": "/books/images/40_image.jpg",
"book_reading_counter": 0,
"book_rating_avg": 5,
"book_uploaded_date": "2025-01-13T20:16:30.768257Z",
"book_favourite_counter": 0,
"status": "Accepted",
"book_description": "The Call of the Wild by Jack London is a timeless adventure novel that tells the story of Buck, a
domesticated dog who is thrust into the harsh wilderness of the Yukon during the Klondike Gold Rush. As he adapts to his
Online
Find and replace
Console
Postbot
Runner
Start Proxy
Cookies
Vault
Trash

Chapter 8: Conclusion and future work

8.1 Review of the project:

We created the SmartLib website to help people who want to enjoy reading and build a reading habit using a gamification system.

Our vision is to create a wonderful and enjoyable reading experience for readers.

The main goal of the SmartLib website is to make it easier for readers to access and enjoy digital books, improving their reading experience and making the process more efficient.

8.2 Future Works:

- Design SmartLib in application on mobile.
- Develop gamification system, which set more challenges for reader to get point.
- Add new features, translate and summarize books with different language.
- Make the reader and manager able to upload books with different extensions, not only pdf

Bibliography

Aaron van den Oord, S. D. (2016). Retrieved from https://arxiv.org/abs/1609.03499
Book Bub. (2024, 4 15). Retrieved from Book Bub: https://www.bookbub.com/
Géron, A. (2019). *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow.*
Martin, D. J. (2019). *Speech and Language Processing.*
open library. (2024, 4 15). Retrieved from open library: https://openlibrary.org/
WordKraft. (2024, 4 15). Retrieved from WordKraft: https://wordkraft.ai/
Wu, Y. (2016). Retrieved from https://arxiv.org/abs/1609.08144

Final-Report_SmartLib.docx

ORIGINALITY REPORT

83%	5%	3%	83%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Birzeit University Main Library Student Paper	82%
2	www.coursehero.com Internet Source	<1%
3	Submitted to University of Hertfordshire Student Paper	<1%
4	Submitted to Eiffel Corporation Student Paper	<1%
5	Submitted to Robert Kennedy College Student Paper	<1%
6	waikato.researchgateway.ac.nz Internet Source	<1%
7	repository.um.edu.my Internet Source	<1%

Exclude quotes	On	Exclude matches	< 4 words
Exclude bibliography	On		