SWE 411

Software Requirements Specification (SRS) Template

The document in this file is an annotated outline for specifying software requirements, adapted from the IEEE Guide to Software Requirements Specifications (Std 830-1993).

Tailor this to your needs, removing explanatory comments as you go along. Where you decide to omit a section, you might keep the header, but insert a comment saying why you omit the data.

SWE 411

Group 16

(Team Name)

Muqin

Software Requirements Specification

Document

Document Revision History

Date	Version	Description	Author
<dd/mmm/yy>	<x.x>	<details>	<name>

Table of Contents

1. Introduction

1.1 Purpose

The purpose of this Software Requirements Specification (SRS) document is to provide a comprehensive and detailed description of the functional and non-functional requirements for the Muqin app. This document aims to ensure a clear understanding of the app's requirements and functionalities, facilitating the design, development, testing, and maintenance phases.

The intended audience for this SRS includes:

Developers: To understand the system's requirements and functionalities, guiding the coding and implementation process.
Project Managers: To oversee the project's progress, ensuring that the development aligns with the specified requirements.
Marketing Staff: To gain insights into the app's features and capabilities, aiding in the promotion and positioning of Muqin in the market.
Users: To understand the capabilities and features of Muqin, ensuring the app meets their expectations and needs.
Testers: To derive test cases and scenarios based on the specified requirements, ensuring the app's quality and reliability.
Documentation Writers: To create user manuals, help guides, and other related documentation, providing users with clear instructions and information about Muqin.

This document serves as the primary reference point for all stakeholders, ensuring alignment, clarity, and consistency throughout the project's lifecycle.

1.2 Scope

Muqin is envisioned as a comprehensive mobile application designed to revolutionize the reading experience for its users.

Included:

1. **Mobile Application Development:** Muqin will be available on both Android and iOS platforms, ensuring a wide reach and accessibility.
2. **AI Integration:** The application will harness the power of pretrained models and APIs, particularly large language models, to offer features like book summarization and visual content generation.
3. **User Experience:** A central feature of Muqin is its user-friendliness. From account creation to reading, users will find an intuitive and engaging interface.
4. **Reading and Management:** Users will have access to a catalog of books, with features like search, personal library management, reading controls, and bookmarking.
5. **Enhanced Features:** Beyond traditional reading, Muqin will offer text-to-speech, visualization, and book summarization capabilities, enhancing user engagement.
6. **Security and Compliance:** Muqin will prioritize user data privacy and ensure secure data transmission and storage.

Excluded:

1. **Print Functionality**: The application will not offer an option to print books or excerpts.

2. **Author Interaction**: Muqin will not facilitate direct interactions between authors and readers, such as Q&A sessions or book signings.
3. **Physical Book Sales**: The Muqin app will not facilitate the sale or distribution of physical books or any printed materials.
4. **Live Reading Sessions or Book Clubs**: The application will not support live reading sessions, book clubs, or any kind of real-time group discussions within the app.
5. **Editing or Modification of Book Content**: Users will not be able to edit or modify the text of the books available in the app.
6. **Parental Controls**: The app will not include any settings or controls specifically for parental monitoring or restrictions.
7. **Print Functionality**: There will be no feature to print books or any excerpts from the books available in the app.
8. **Direct Author Interaction**: The Muqin app will not facilitate direct interactions or communications between authors and readers, such as Q&A sessions, book signings, or personalized messages.
9. **E-book Conversion Services**: Muqin will not offer services to convert physical books into e-books.
10. **Multi-Language Support**: While the app may support Arabic and English, it will not provide support for other languages in the initial release.
11. **Internationalization**: The initial release of Muqin will not be tailored for international audiences and may not comply with all foreign data privacy laws.

Benefits:

Muqin aims to provide a modern reading experience by integrating AI capabilities. Users can engage with content visually, audibly, and traditionally, making reading more accessible and immersive. The app's personalization features ensure that users receive content tailored to their preferences, enhancing user retention and satisfaction.

Objectives and Goals:

The primary objective of Muqin is to offer a multi-faceted reading platform that caters to the diverse needs of modern readers. By integrating AI, the app seeks to push the boundaries of traditional reading apps. The goal is to achieve widespread adoption, high user satisfaction, and to promote a culture of reading, especially within the Arabic community.

1.3 Definitions, Acronyms and Abbreviations

1. **Muqin (موقن):** Means sure or certain in Arabic.
2. **SRS:** Software Requirements Specification. A document that describes the features, behaviors, and attributes of the intended software.
3. **AI:** Artificial Intelligence. A branch of computer science that focuses on creating systems capable of performing tasks that usually require human intelligence.
4. **API:** Application Programming Interface. A set of rules and protocols that allow different software entities to communicate with each other.

1.4 References

Provide a complete list of all documents referenced elsewhere in the SRS.

-

1.5 Overview

This Software Requirements Specification (SRS) document for Muqin is structured to provide a comprehensive understanding of the system's requirements, functionalities, and design considerations. The document is organized into three main sections:

1. **Introduction:** This section sets the stage for the document, outlining the purpose, scope, and key terminologies. It also provides references that have been instrumental in shaping the SRS.
2. **Overall Description:** Delving deeper into the product, this section offers a bird's-eye view of Muqin. It discusses the product's perspective, its core functions, and presents a class diagram to visually represent the system's architecture. Additionally, it touches upon the characteristics of the intended users and any constraints that the system might operate under.
3. **Specific Requirements:** This is the heart of the SRS, detailing the nitty-gritty of Muqin's requirements. It encompasses:
   - **Detailed Requirements:** A breakdown of the system's requirements.
   - **Use Case Diagram:** A visual representation of user interactions with the system.
   - **Interface Requirements:** Descriptions of user, hardware, and software interfaces.
   - **Functional Requirements Description:** Detailed explanations of each functional requirement.
   - **Performance and Database Requirements:** Specifications related to system performance and database design.
   - **Software System Attributes:** Key attributes such as security, reliability, availability, maintainability, and portability are discussed, ensuring a holistic understanding of the system's characteristics.

The document concludes with an **Appendix**, providing any additional information or references that complement the main content.

Throughout this SRS, readers will find a systematic and organized presentation of Muqin's requirements, ensuring clarity, alignment, and a solid foundation for the development phase.

2. Overall Description

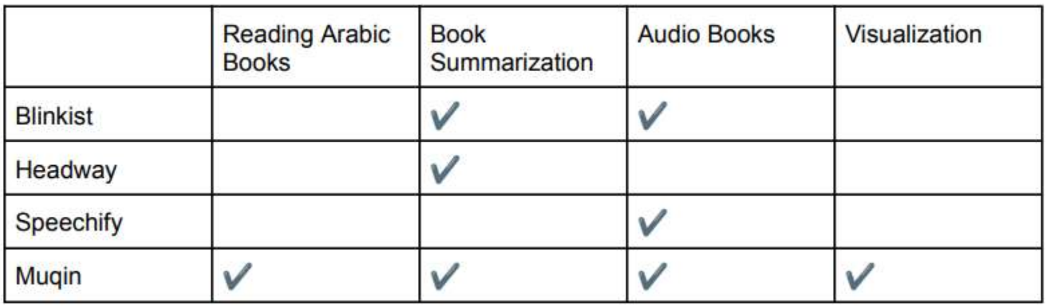
2.1 Product Perspective

Muqin is a mobile application built natively for Android and iOS devices. It is designed specifically for Arabic content and optimized to enhance the reading experience for native Arabic speakers.

The core problem Muqin aims to solve is low readership and exposure to literature among Arabic speakers. It targets Arabic readers across age groups and reading abilities, as well as students and language learners.

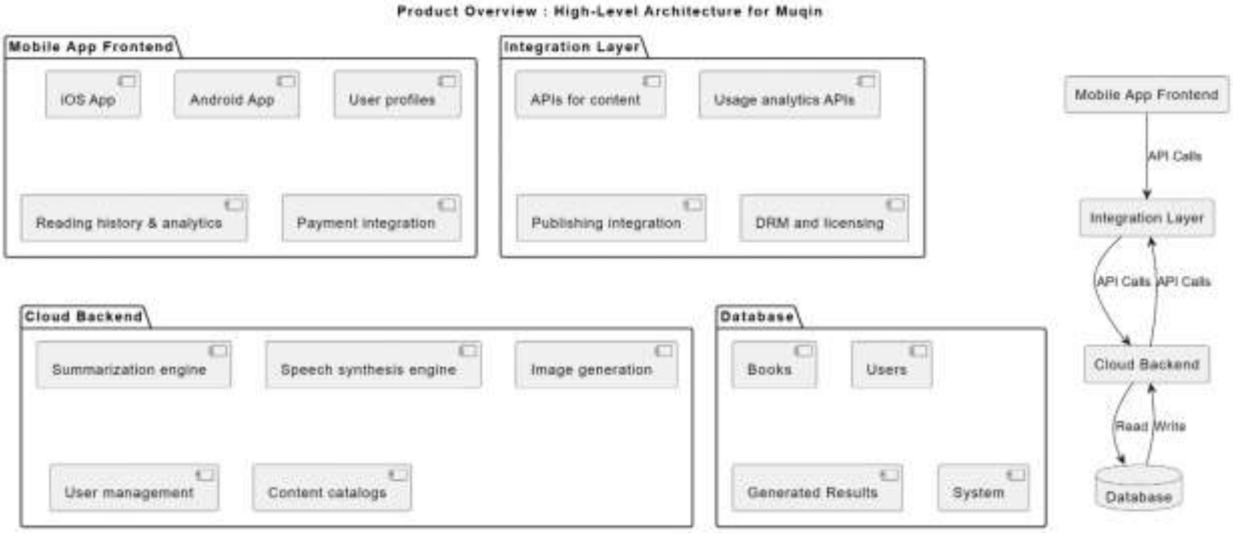
Muqin's key functionalities powered by AI include:

- Summarization – using the power of AI to summarize books.
- Voice acting - Uses speech synthesis to generate audiobook.
- Content visualization - Creates illustrations, and graphics to visualize book content and bring it to life.



	Reading Arabic Books	Book Summarization	Audio Books	Visualization
Blinkist		✓	✓	
Headway		✓		
Speechify			✓	
Muqin	✓	✓	✓	✓

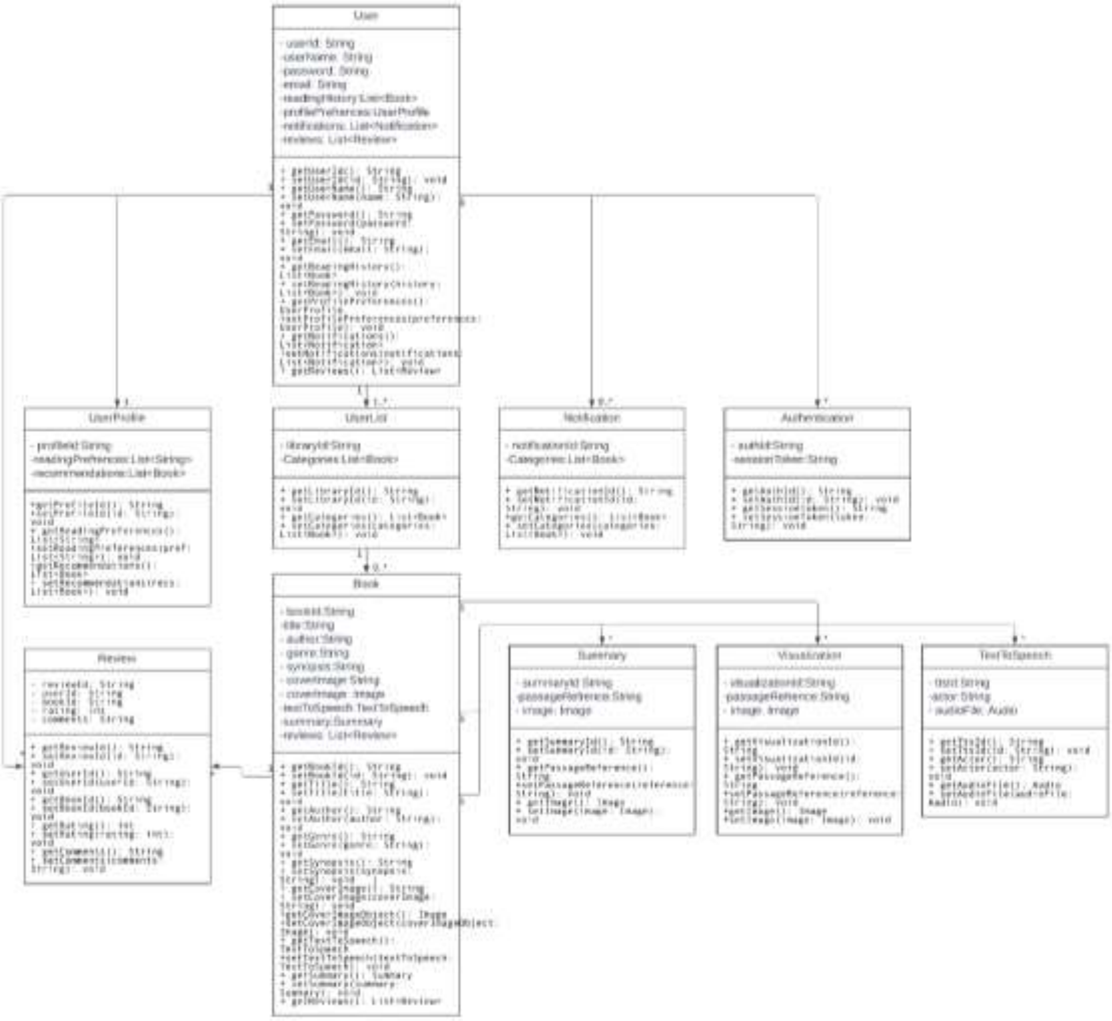
While apps like Blinkist and Speechify offer some similar functions, Muqin is the first solution specialized for Arabic content with AI optimized for the language.
Muqin interacts with publishing partners to obtain licensing and access to Arabic books. It leverages cloud platforms to run the AI models and store generated assets.



2.2 Product Functions

Core functions Muqin provides to users
Muqin enriches the reading experience for users by providing customizable book summaries, engaging audiobooks, AI-generated visuals, and reading progress tracking. It offers seamless cross-device reading continuity, allowing users to enjoy their books wherever they go. These features make reading more efficient, enjoyable, and flexible, motivating users to explore and engage with a diverse range of content. • Browse and read summarized versions of books. • Listen to computer-generated audiobooks. • View AI-generated visualizations like illustrations, and graphics accompanying content. • Track and visualize reading progress.

2.2.1 Class Diagram



2.3 User Characteristics

The intended users of Muqin are diverse, aiming to cater to a wide audience interested in reading Arabic and non-Arabic books. The user base includes:

- **Readers**: Individuals with varying educational levels and reading habits, seeking an enhanced and entertaining reading experience through the app's AI-driven features such as voice acting and content visualization.
- **Students**: Users who are in educational institutions, looking to access valuable educational resources and book summarizations to aid in their studies and expand their knowledge.
- **Authors**: Individuals who have written books and are looking to reach a broader readership through the platform, especially within the Arabic-speaking community.
- **Publishing Institutions**: Organizations that publish books and are interested in collaborating with Muqin to make literature more accessible and promote reading culture.

2.4 Constraints

The development and functionality of Muqin are subject to several constraints, including:

- **Language Support**: The app needs to support both Arabic and other languages, which may impose additional development and translation efforts.
- **AI Integration**: The integration of advanced AI technologies for voice acting and content visualization requires specialized expertise and may limit the choice of development tools and platforms.
- **Cloud Services**: The app relies on cloud services providers for various functionalities, which may impose limitations on data storage, processing, and security.
- **Content Acquisition**: Collaborations with publishing institutions are essential for acquiring book content, which may impose legal and contractual constraints.
- **Accessibility**: The app aims to make literature accessible to a wider audience, including those with disabilities, necessitating adherence to accessibility standards and guidelines.
- **Technological Advancement**: The app aims to introduce revolutionary AI technologies to the Arab world, which may require adherence to specific technological standards and protocols.

These constraints will guide the design, implementation, and deployment of Muqin, ensuring its reliability, usability, and alignment with the project's objectives.

3. Specific Requirements

3.1 Detailed Requirements

1. User Authentication and Authorization:
   1.1. Users should be able to create accounts.
   1.2. Users should be able to log in securely.

2. Book Catalog:
   2.1. The system shall display a catalog of available books.
   2.2. Allow users to search for books by title, author, genre, etc.
   2.3. Provide book details such as title, author, genre, synopsis, and cover image.
   2.4. Adding books will happen through a ticketing system utilizing Google Forms.

3. User Profile:
   3.1. Users should have profiles to track their reading history.
   3.2. Display user-specific recommendations based on reading history.

4. Reading Features:
   4.1. Users should be able to open and read books.
   4.2. Implement basic reading controls (e.g., page flipping, font size adjustment, theme change).
   4.3. Support multiple file formats (e.g., PDF, EPUB).
   4.4. Allow users to bookmark pages for later reference.
   4.5. Ther user should be able to download books for offline reading.

5. Library Management:
   5.1. Users should be able to add/remove books from their personal library.
   5.2. Users should be able to organize books into categories (e.g., "To Read," "Read," "Favorites").
   5.3. Users should be able to create lists to organize books.

6. Text-To-Speech:
   6.1. The app shall be able to perform text-to-speech to text.

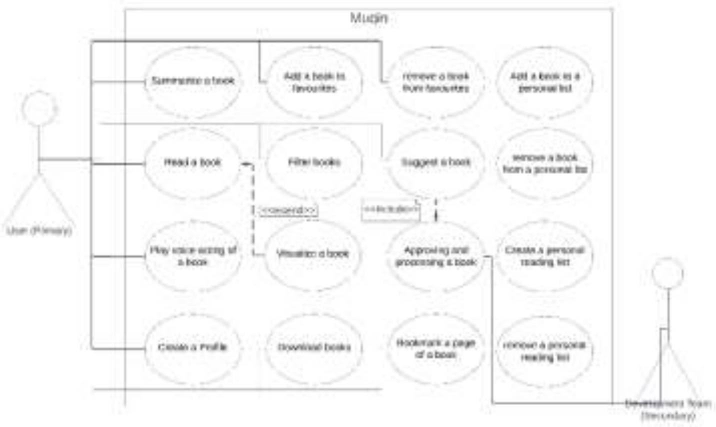
7. Visualization:
   7.1. The app shall be able to generate a picture from a passage depicting the events in the passage.

8. Summarization:
   8.1. The app shall be able to generate a summary of a passage.

9. Notifications:
   9.1. The system should notify users about new releases, or updates related to their favorite genres.
   9.2. Send reminders for unread books or incomplete readings.

10. Cross-Platform Compatibility:
    10.1. The app should support both android and IOS platforms.
11. Security:
    11.1. Implement secure data transmission and storage.

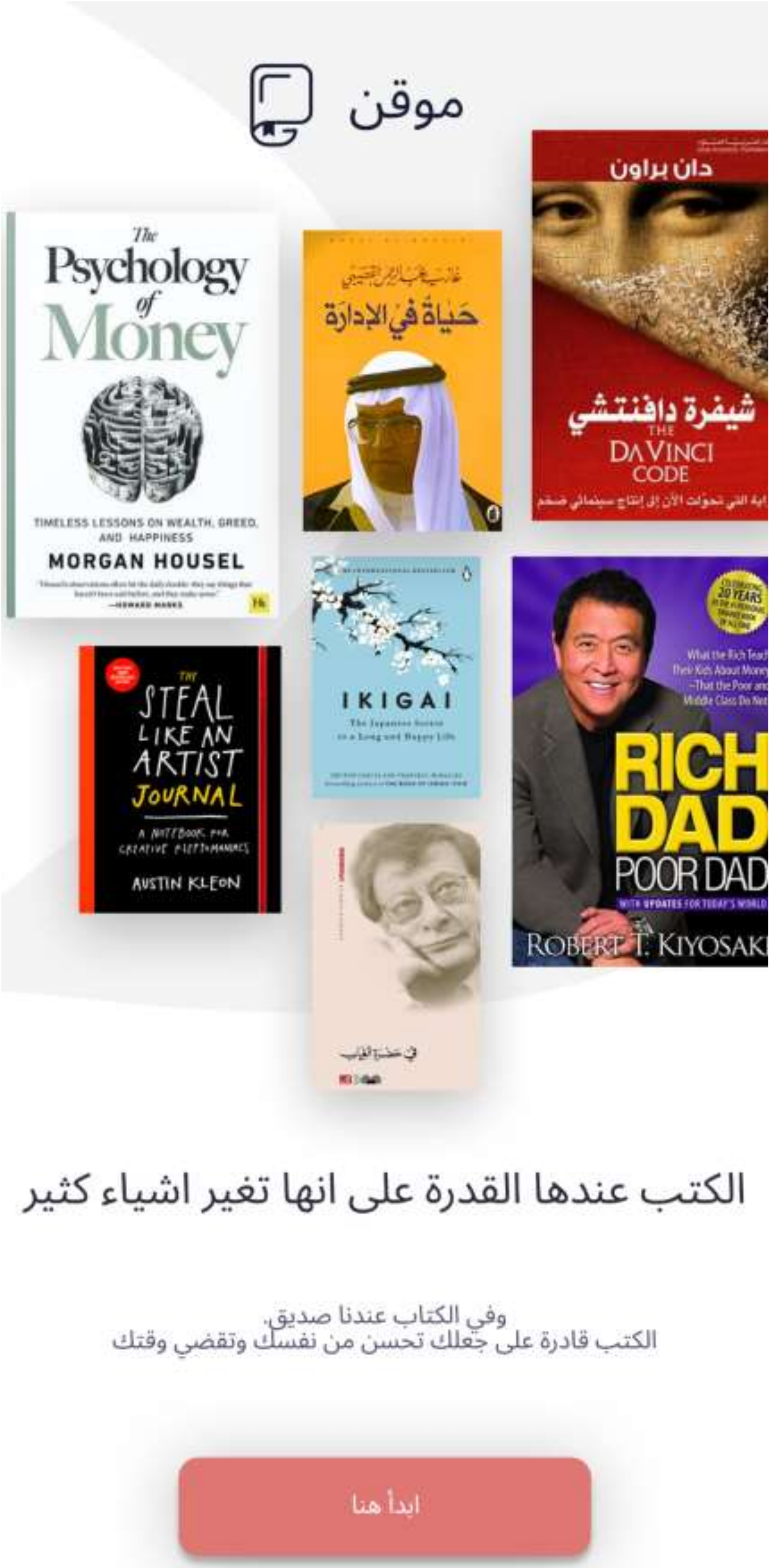
3.2 Use case Diagram.


Mudjin
Summarize a book
Add a book to favourites
remove a book from favourites
Add a book to a personal list
Read a book
Filter books
Suggest a book
remove a book from a personal list
<<extend>>
<<include>>
User (Primary)
Play voice acting of a book
Visualize a book
Approving and processing a book
Create a personal reading list
Create a Profile
Download books
Bookmark a page of a book
remove a personal reading list
Development Team (Secondary)

3.3 Interface Requirements

3.3.1 User Interfaces

موقن



موقن

الكتب عندها القدرة على انها تغير اشياء كثير

وفي الكتاب عندنا صديق.
الكتب قادرة على جعلك تحسن من نفسك وتقضي وقتك

← تخطي

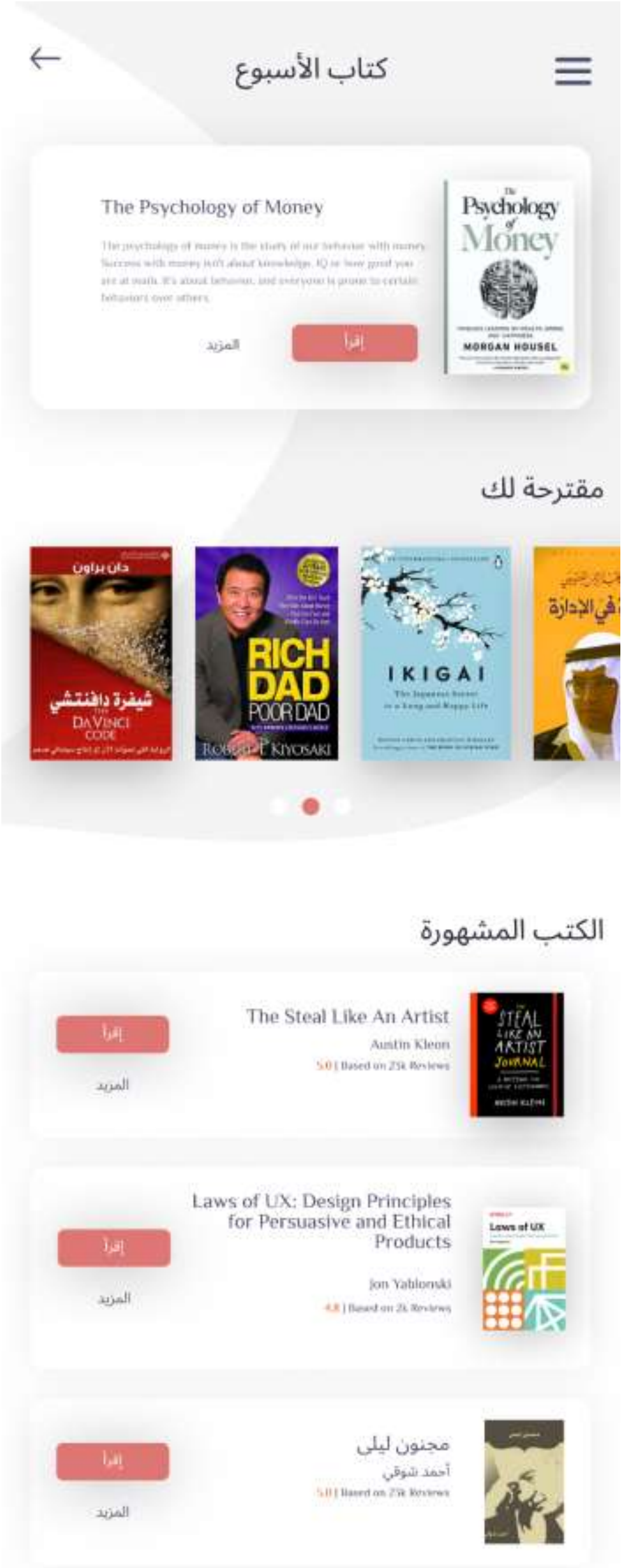
بالكتب نزداد علما

الكتب تساعدك على زيادة معرفتك وتؤهلك للنجاح

← تخطي

تعلم بذكاء

حنا في عام 2024 وجا وقت انك تتعلم بذكاء.
سهلناها عليك وخلينا الكتب عندنا والتلخيص عندنا وحتى
القراءة عندنا باقي عليك انك تفتح مداركك وتتعلم



كتاب الأسبوع

The Psychology of Money

The psychology of money is the study of our behavior with money. Success with money isn't about knowledge, IQ or how good you are at math. It's about behavior, and everyone is prone to certain behaviors over others.

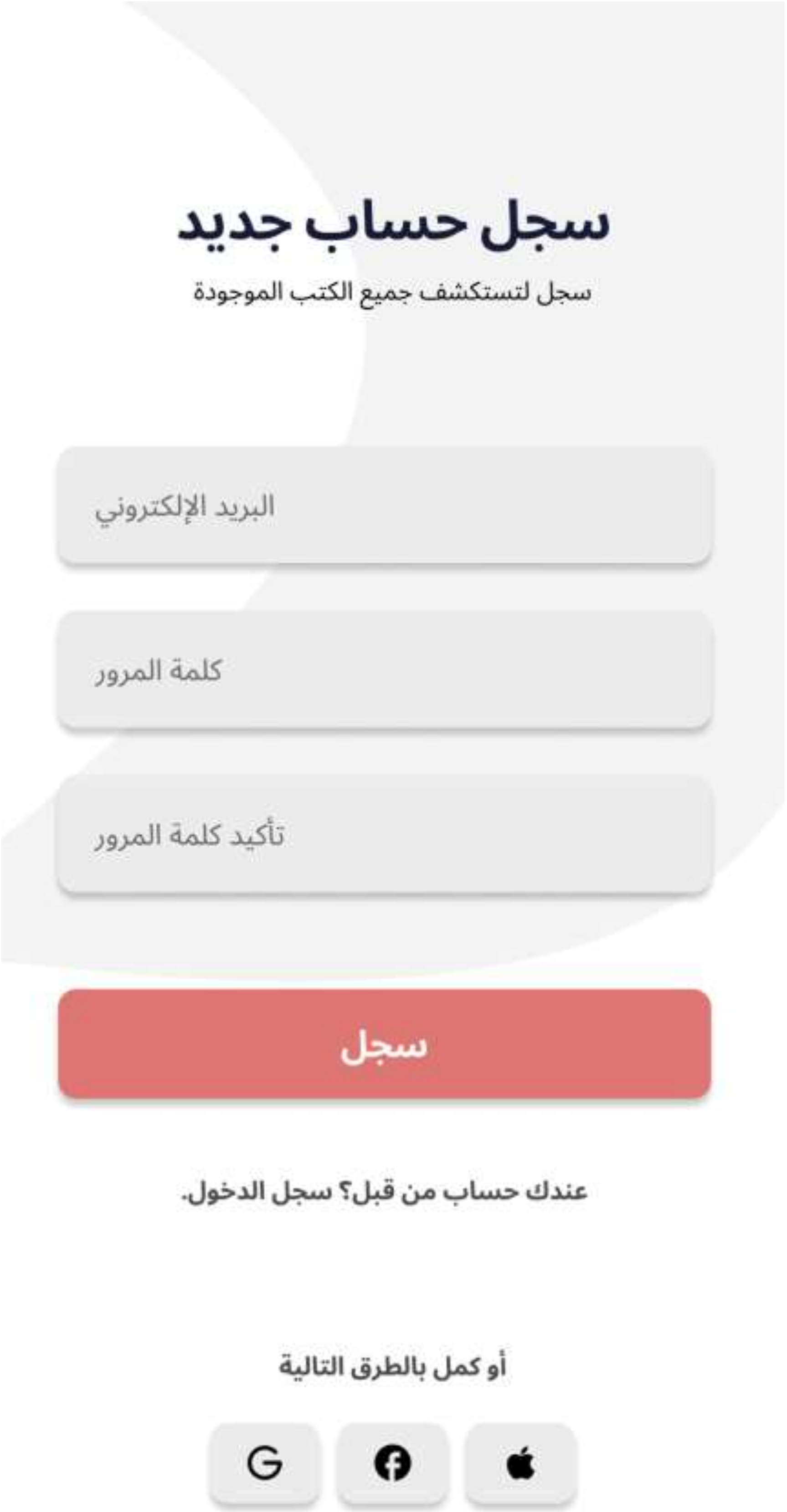
مقترحة لك

الكتب المشهورة

The Steal Like An Artist
Austin Kleon
5.0 | Based on 25k Reviews

Laws of UX: Design Principles for Persuasive and Ethical Products
Jon Yablonski
4.8 | Based on 2k Reviews

مجنون ليلى
أحمد شوقي
5.0 | Based on 25k Reviews

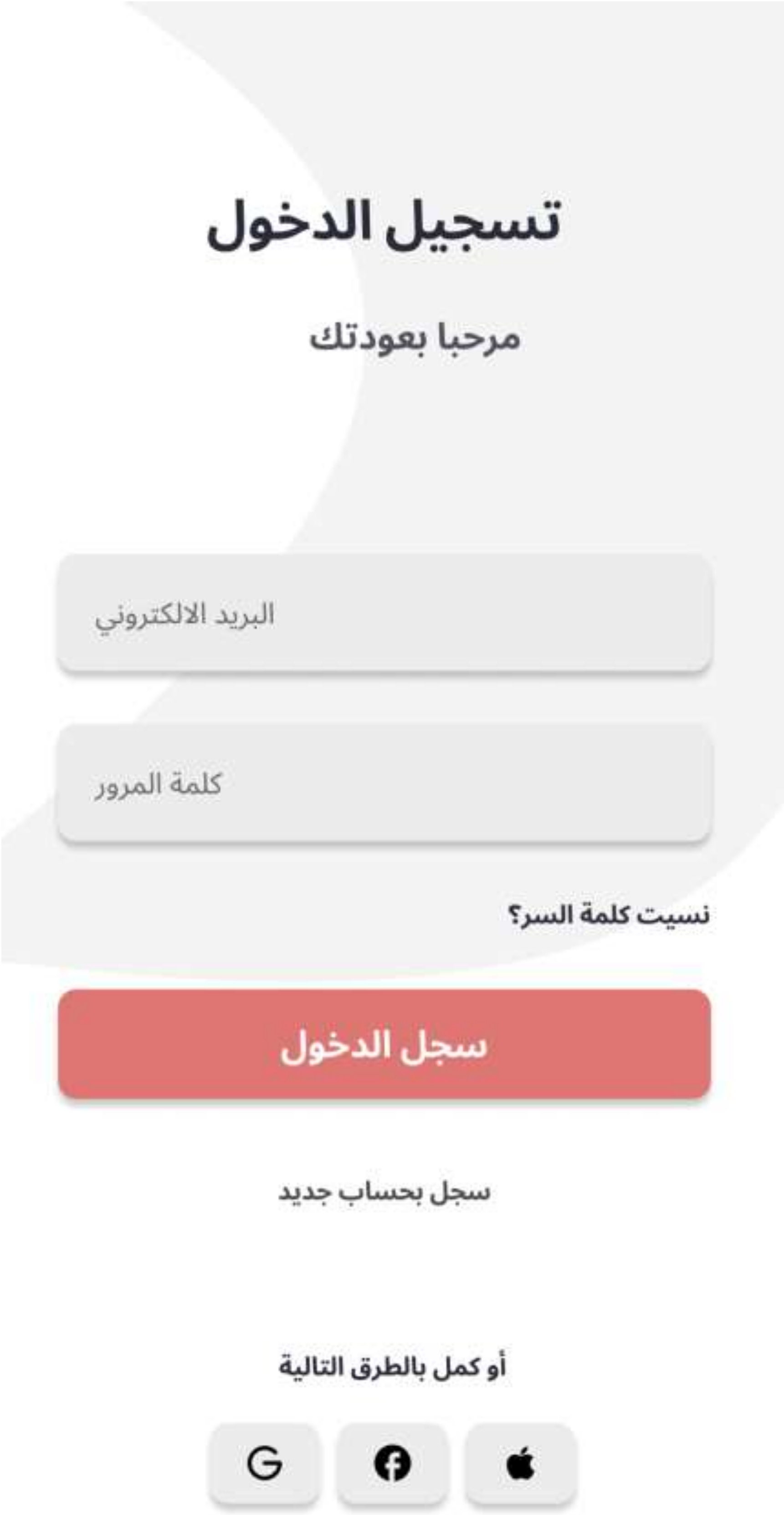


سجل حساب جديد

سجل لتستكشف جميع الكتب الموجودة

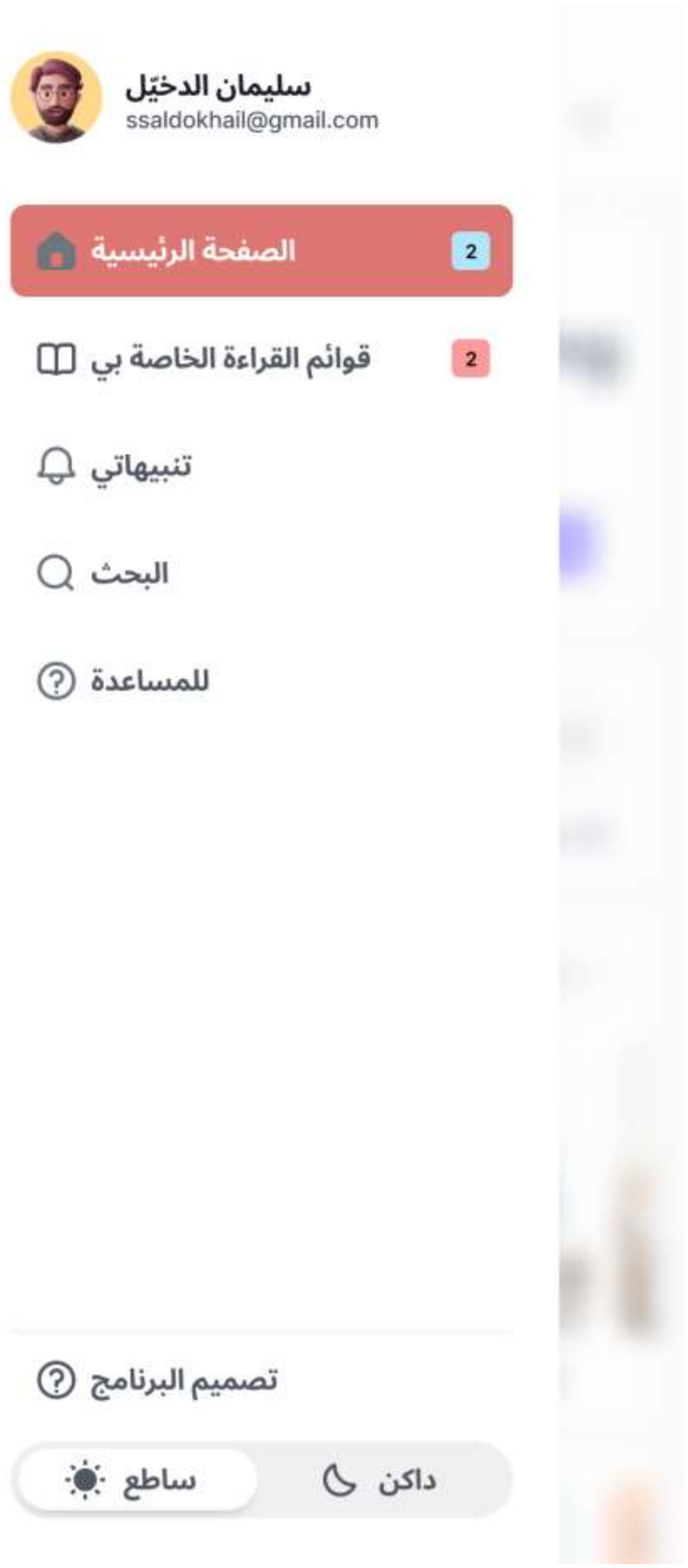
عندك حساب من قبل؟ سجل الدخول.

أو كمل بالطرق التالية

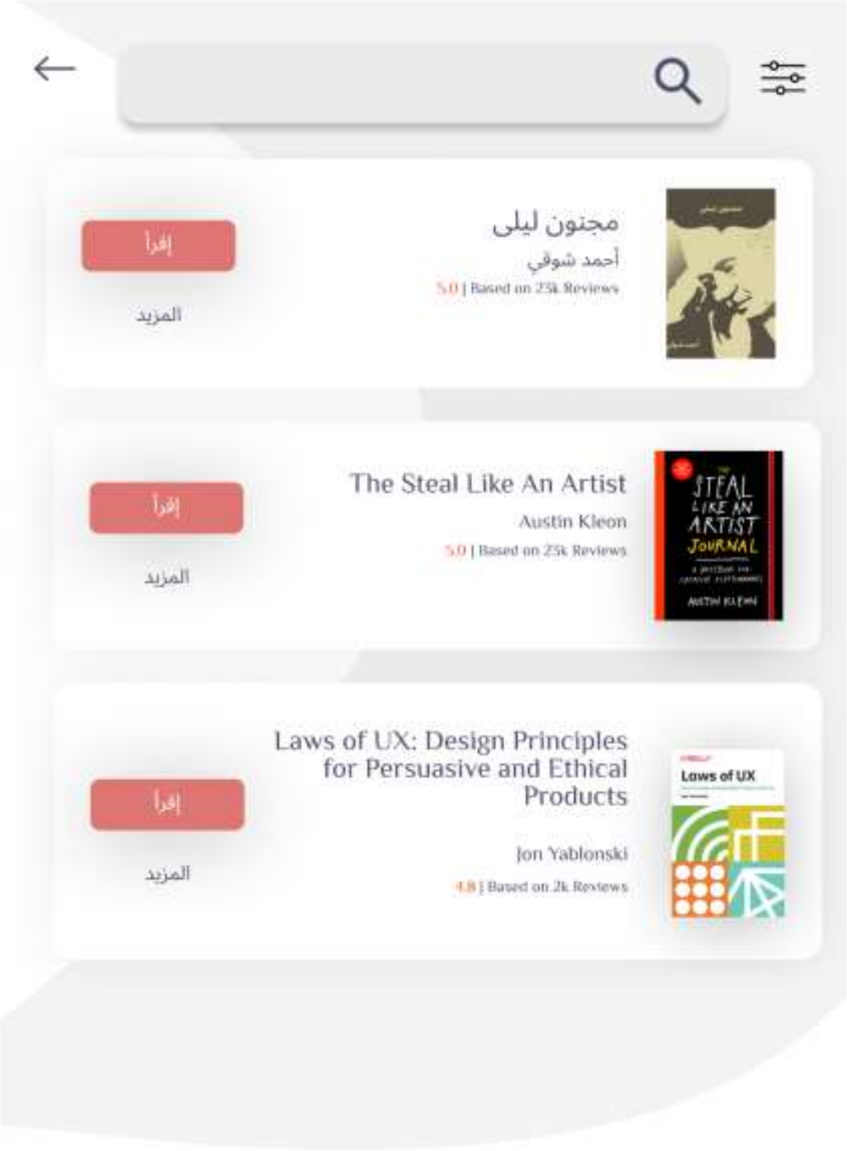


تسجيل الدخول

مرحبا بعودتك



سليمان الدخيّل
ssaldokhail@gmail.com



مجنون ليلى

أحمد شوقي

5.0 | Based on 25k Reviews

The Steal Like An Artist

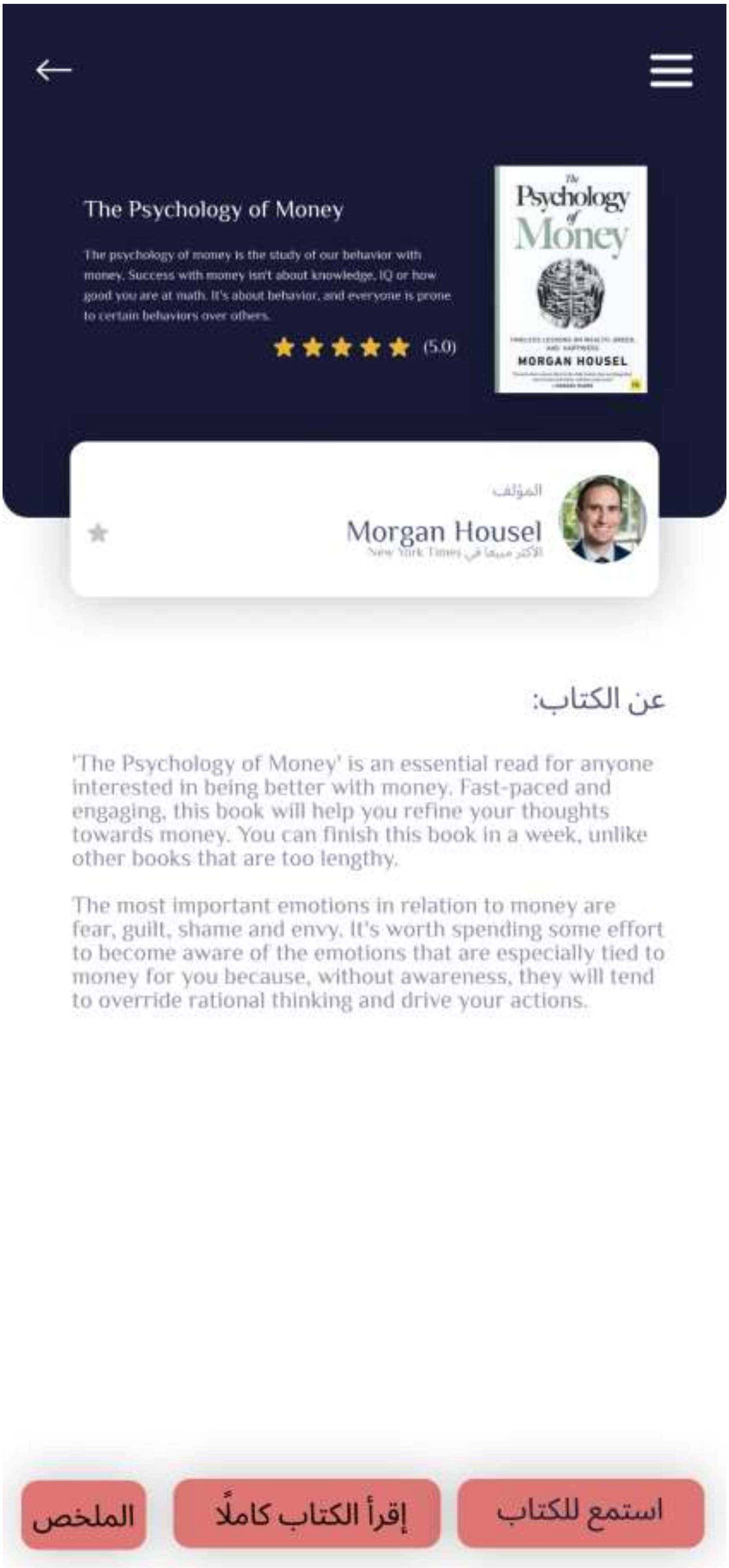
Austin Kleon

5.0 | Based on 25k Reviews

Laws of UX: Design Principles for Persuasive and Ethical Products

Jon Yablonski

4.8 | Based on 2k Reviews



The Psychology of Money

The psychology of money is the study of our behavior with money. Success with money isn't about knowledge, IQ or how good you are at math. It's about behavior, and everyone is prone to certain behaviors over others.

★★★★★ (5.0)

المؤلف

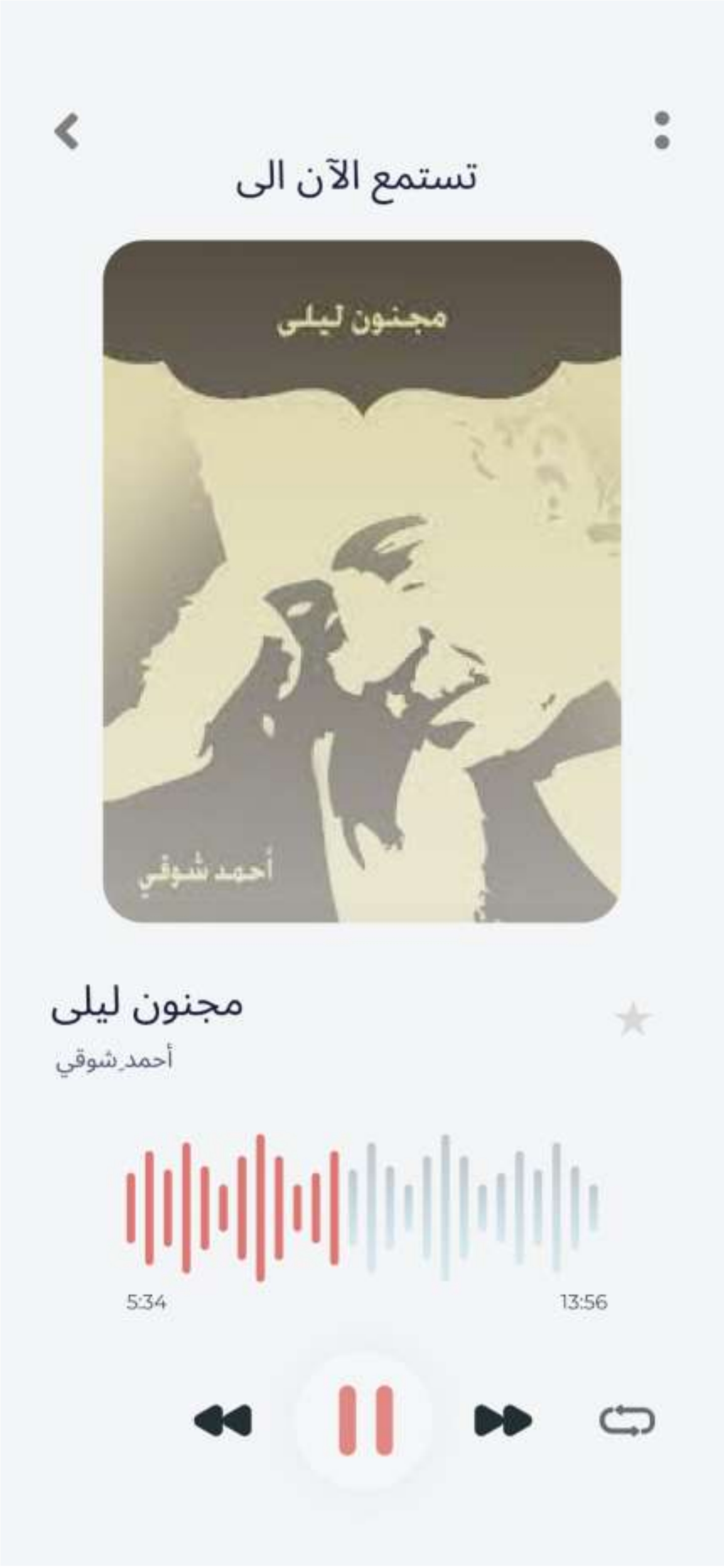
Morgan Housel

الأكثر مبيعا في New York Times

عن الكتاب:

'The Psychology of Money' is an essential read for anyone interested in being better with money. Fast-paced and engaging, this book will help you refine your thoughts towards money. You can finish this book in a week, unlike other books that are too lengthy.

The most important emotions in relation to money are fear, guilt, shame and envy. It's worth spending some effort to become aware of the emotions that are especially tied to money for you because, without awareness, they will tend to override rational thinking and drive your actions.



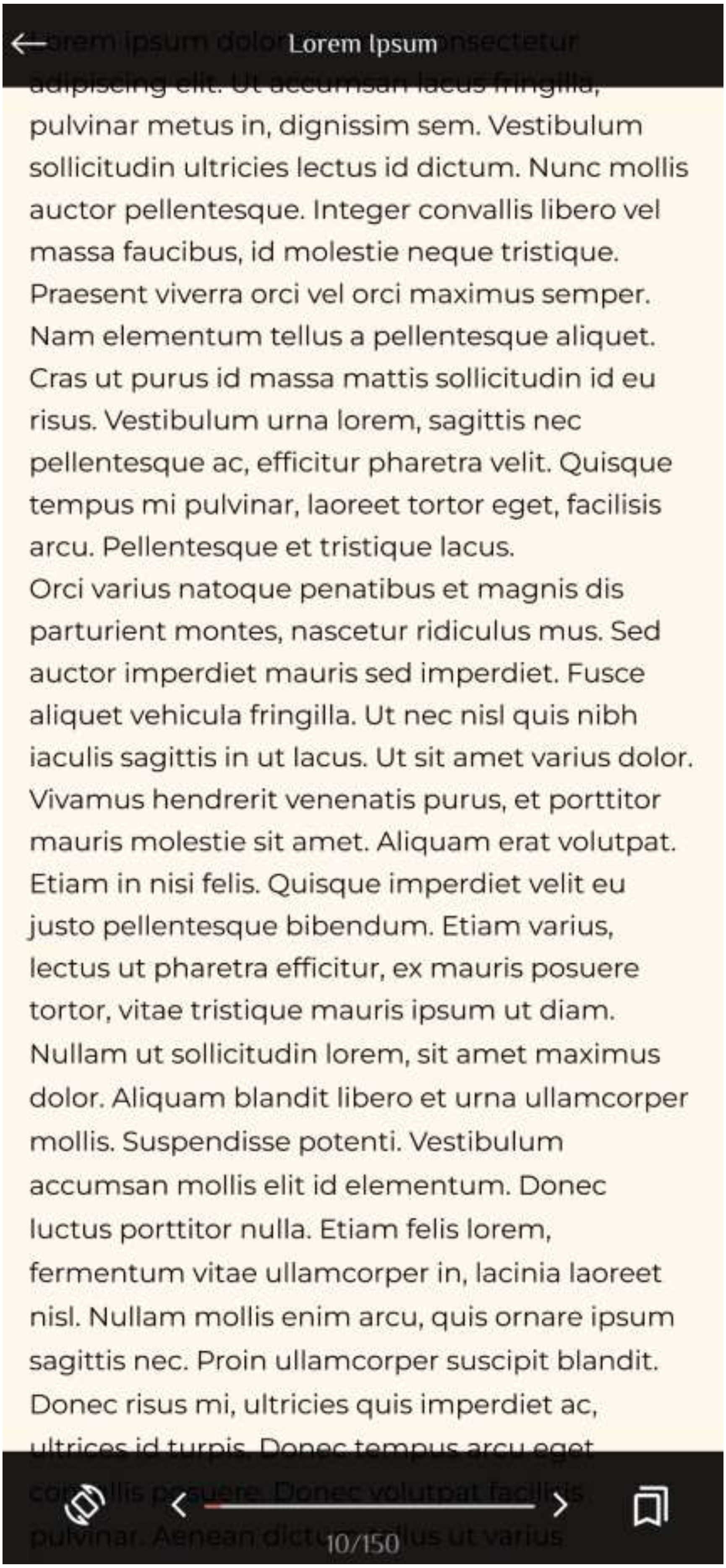
تستمع الآن الى

مجنون ليلى

أحمد شوقي

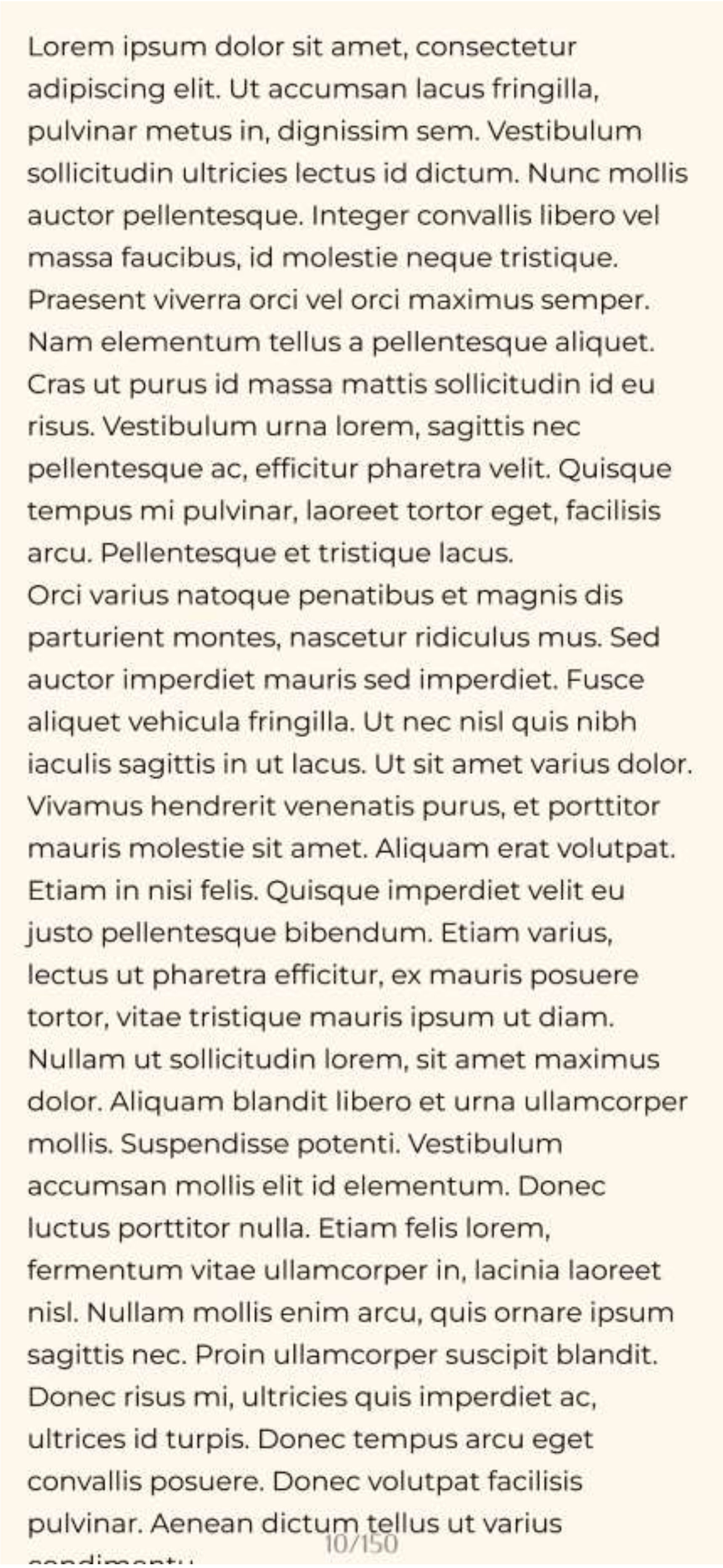
5:34

13:56



Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut accumsan lacus fringilla, pulvinar metus in, dignissim sem. Vestibulum sollicitudin ultricies lectus id dictum. Nunc mollis auctor pellentesque. Integer convallis libero vel massa faucibus, id molestie neque tristique. Praesent viverra orci vel orci maximus semper. Nam elementum tellus a pellentesque aliquet. Cras ut purus id massa mattis sollicitudin id eu risus. Vestibulum urna lorem, sagittis nec pellentesque ac, efficitur pharetra velit. Quisque tempus mi pulvinar, laoreet tortor eget, facilisis arcu. Pellentesque et tristique lacus.

Orci varius natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Sed auctor imperdiet mauris sed imperdiet. Fusce aliquet vehicula fringilla. Ut nec nisl quis nibh iaculis sagittis in ut lacus. Ut sit amet varius dolor. Vivamus hendrerit venenatis purus, et porttitor mauris molestie sit amet. Aliquam erat volutpat. Etiam in nisi felis. Quisque imperdiet velit eu justo pellentesque bibendum. Etiam varius, lectus ut pharetra efficitur, ex mauris posuere tortor, vitae tristique mauris ipsum ut diam. Nullam ut sollicitudin lorem, sit amet maximus dolor. Aliquam blandit libero et urna ullamcorper mollis. Suspendisse potenti. Vestibulum accumsan mollis elit id elementum. Donec luctus porttitor nulla. Etiam felis lorem, fermentum vitae ullamcorper in, lacinia laoreet nisl. Nullam mollis enim arcu, quis ornare ipsum sagittis nec. Proin ullamcorper suscipit blandit. Donec risus mi, ultricies quis imperdiet ac, ultrices id turpis. Donec tempus arcu eget convallis posuere. Donec volutpat facilisis pulvinar. Aenean dictum tellus ut varius



Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut accumsan lacus fringilla, pulvinar metus in, dignissim sem. Vestibulum sollicitudin ultricies lectus id dictum. Nunc mollis auctor pellentesque. Integer convallis libero vel massa faucibus, id molestie neque tristique. Praesent viverra orci vel orci maximus semper. Nam elementum tellus a pellentesque aliquet. Cras ut purus id massa mattis sollicitudin id eu risus. Vestibulum urna lorem, sagittis nec pellentesque ac, efficitur pharetra velit. Quisque tempus mi pulvinar, laoreet tortor eget, facilisis arcu. Pellentesque et tristique lacus.

Orci varius natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Sed auctor imperdiet mauris sed imperdiet. Fusce aliquet vehicula fringilla. Ut nec nisl quis nibh iaculis sagittis in ut lacus. Ut sit amet varius dolor. Vivamus hendrerit venenatis purus, et porttitor mauris molestie sit amet. Aliquam erat volutpat. Etiam in nisi felis. Quisque imperdiet velit eu justo pellentesque bibendum. Etiam varius, lectus ut pharetra efficitur, ex mauris posuere tortor, vitae tristique mauris ipsum ut diam. Nullam ut sollicitudin lorem, sit amet maximus dolor. Aliquam blandit libero et urna ullamcorper mollis. Suspendisse potenti. Vestibulum accumsan mollis elit id elementum. Donec luctus porttitor nulla. Etiam felis lorem, fermentum vitae ullamcorper in, lacinia laoreet nisl. Nullam mollis enim arcu, quis ornare ipsum sagittis nec. Proin ullamcorper suscipit blandit. Donec risus mi, ultricies quis imperdiet ac, ultrices id turpis. Donec tempus arcu eget convallis posuere. Donec volutpat facilisis pulvinar. Aenean dictum tellus ut varius condimentu

3.3.2 Hardware Interfaces

Based on the Software Requirements Specification for the Muqin application, the system does not necessitate any specialized hardware interfaces.

3.3.3 Software Interfaces

Name	TensorFlow
Mnemonic	TF
Version number	2.0 or higher
Source	https://www.tensorflow.org/
Purpose of Interfacing	framework to develop AI models for summarization, speech synthesis, etc.

.

Name	Flutter
Mnemonic	Flutter
Version number	Latest stable
Source	https://flutter.dev/
Purpose of Interfacing	Cross-platform mobile app development framework for building iOS and Android apps.

Name	Firebase
Mnemonic	Firebase
Version number	Latest
Source	firebase.google.com/
Purpose of Interfacing	Backend development including database, storage, hosting, analytics, etc.

Name	Figma
Mnemonic	Figma
Version number	Latest
Source	www.figma.com/
Purpose of Interfacing	Collaborative UI/UX design and prototyping.

Name	GitHub & Git
Mnemonic	GitHub
Version number	N/A
Source	Github.com
Purpose of Interfacing	Version control, code hosting, project management.

Name	Hugging Face Transformers
Mnemonic	HuggingFace
Version number	N/A
Source	https://huggingface.co
Purpose of Interfacing	State-of-the-art NLP models like BERT, GPT-2, etc.

Name	Anthropic Claude API
Mnemonic	Claude
Version number	2.0
Source	https://www.anthropic.com
Purpose of Interfacing	Conversational AI assistant with summarization capabilities.

3.4 Functional Requirements Description

F1: User Authentication and Authorization	
Priority	High
Effort	2 person-months
Risk	Medium
Use Case(s)	UC5: Create a profile
Description	Allows users to create accounts and log in securely.
Notes	Ensure encryption for passwords.

F2: Book Catalog	
Priority	High
Effort	3 person-months
Risk	Medium
Use Case(s)	UC1: Read a book. UC2: Filter Books.
Description	Display a catalog of books, allow search, and provide book details.
Notes	

F3: User Profile	
Priority	High
Effort	2 person-months
Risk	Low
Use Case(s)	UC3: Suggest a book. UC5: Create a profile.
Description	Users have profiles to track reading history and get recommendations.
Notes	

F4: Reading Features	
Priority	High
Effort	3 person-months
Risk	Medium
Use Case(s)	UC1: Read a book, UC4: Bookmark a page, UC6: Download a book
Description	Users can read, bookmark, and download books with reading controls.
Notes	

F5: Library Management	
Priority	Medium
Effort	2 person-months
Risk	Low
Use Case(s)	UC7: Add a book to a reading list.

	UC8: Remove a reading list. UC9: Add a book to favorites. UC10: Remove a book from favorites. UC11: Create a reading list. UC12: Remove a book from a personal list.
Description	Users can manage their personal library, organize books, and create reading lists.
Notes	

F6: Text-To-Speech	
Priority	Medium
Effort	1 person-month
Risk	Medium
Use Case(s)	UC14: Play voice acting of a book
Description	The app can convert text to speech.
Notes	

F7: Visualization	
Priority	Medium
Effort	2 person-months
Risk	Medium
Use Case(s)	UC15: Visualize a book
Description	Generate visuals from book passages.
Notes	

F8: Summarization	
Priority	Medium
Effort	2 person-months
Risk	Medium
Use Case(s)	UC13: Summarize a book
Description	Generate summaries of book passages.
Notes	

F9: Notifications	
Priority	Medium
Effort	1 person-month
Risk	Low
Use Case(s)	Not directly related to a specific use case.
Description	Notify users about updates, releases, and reading reminders.
Notes	Ensure timely and relevant notifications.

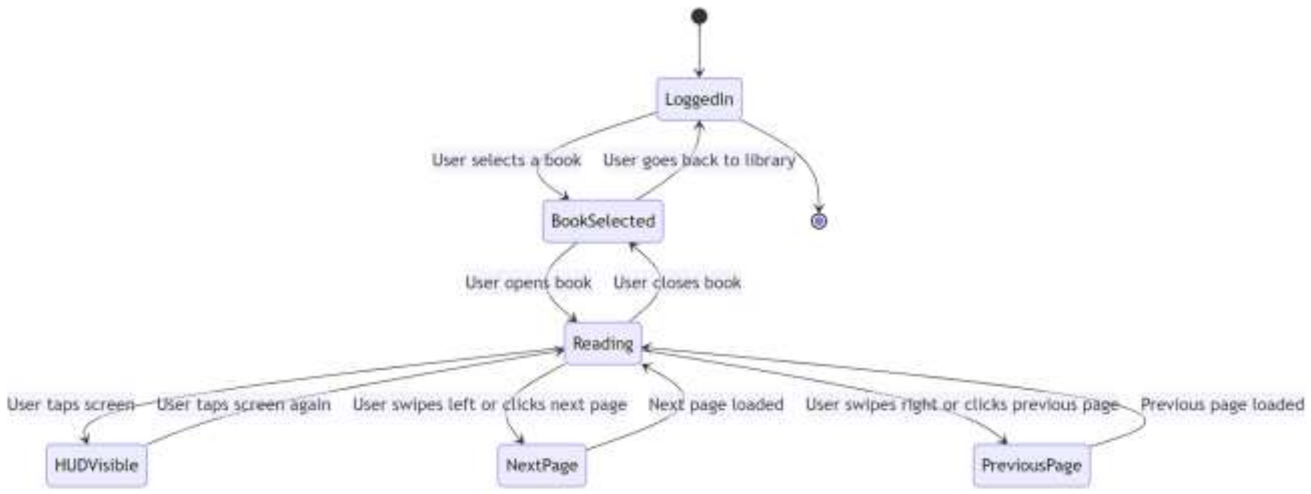
F10: Cross-Platform Compatibility	
Priority	High
Effort	3 person-months
Risk	Medium
Use Case(s)	Not directly related to a specific use case.
Description	The app should work on both Android and iOS platforms.
Notes	

F11: Security	
Priority	High
Effort	2 person-months

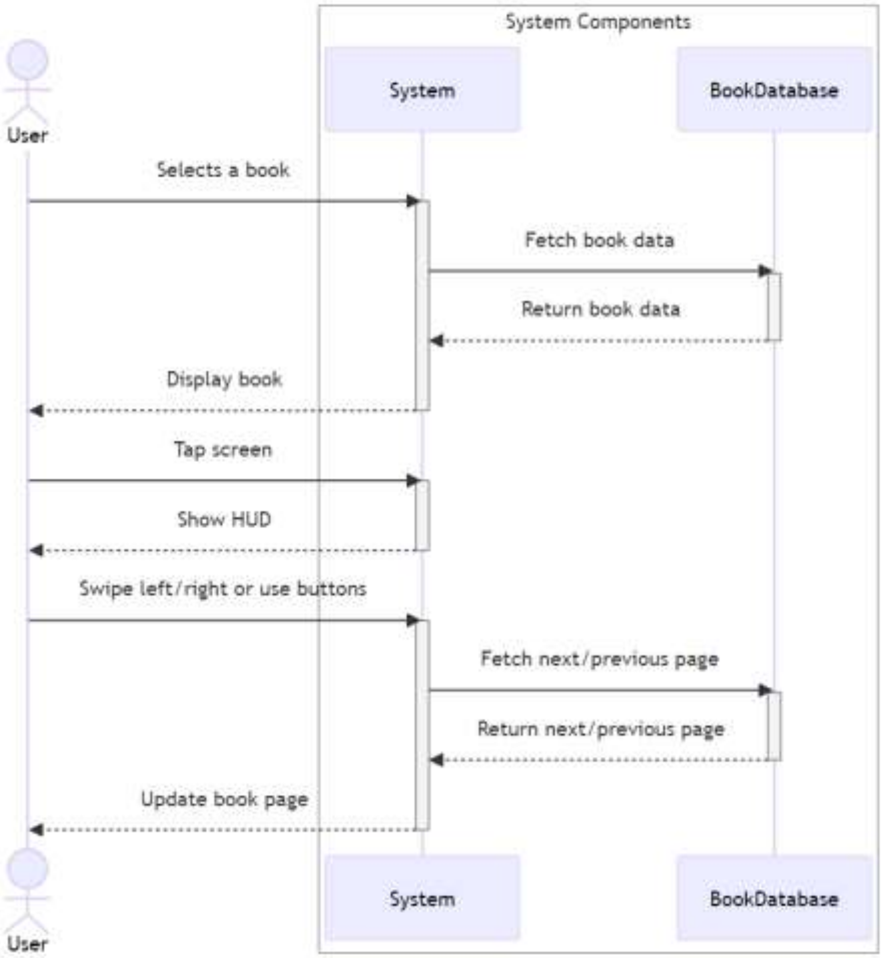
Risk	High
Use Case(s)	UC5: Create a profile
Description	Secure data transmission and storage.
Notes	

Use Case Details	**UC1: Read a Book**
Description	Allows the user to read a selected book in an environment, with options to navigate pages.
Actors	User (Reader)
Pre-Condition(s)	• User is logged in. • The book is available in the user's library or the main catalog.
Post-Condition(s)	The user has read or navigated through the book.
Main Flow	1. User selects a book from the library or catalog. 2. The system opens the book to the last read page (or the first page if never read). 3. The user can read the book with no HUD or buttons displayed. 4. Upon tapping the screen, a horizontal bar appears at the bottom with navigation buttons. 5. The user can navigate to the next or previous page using the buttons or by swiping left or right. 6. The user can hide the HUD by tapping the screen again.
Alternative(s)	If the book fails to load, display an error message to the user.
Dependency	Requires integration with the book database.

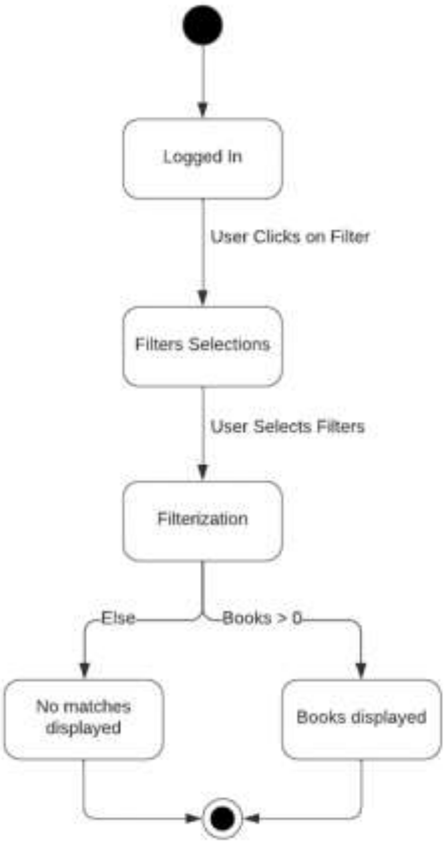
UC1 State Diagram

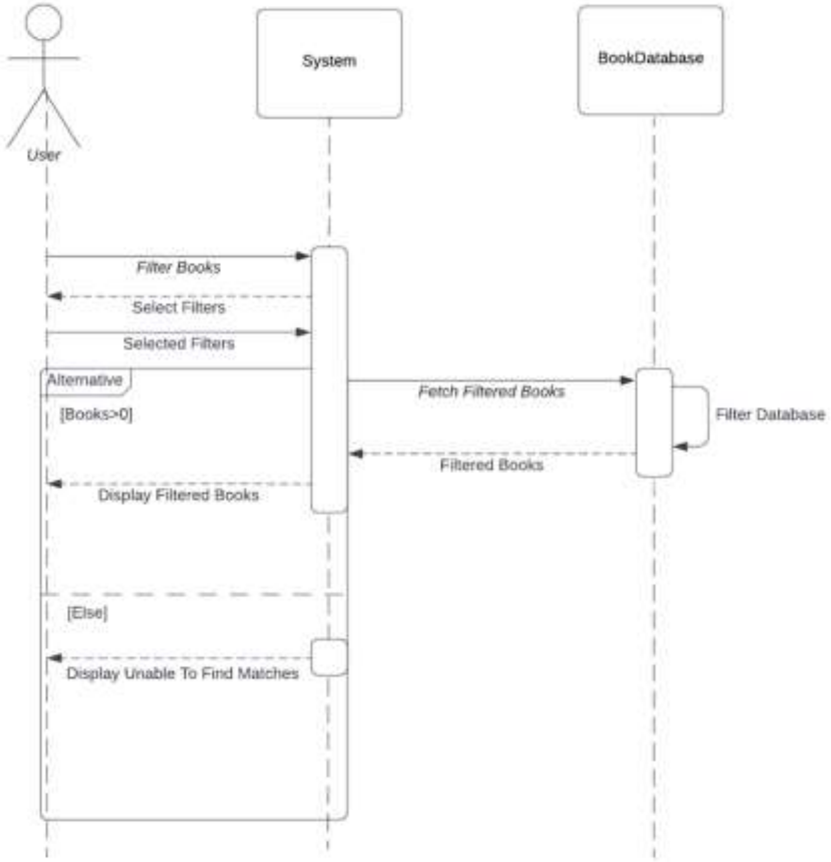


UC1 Sequence Diagram



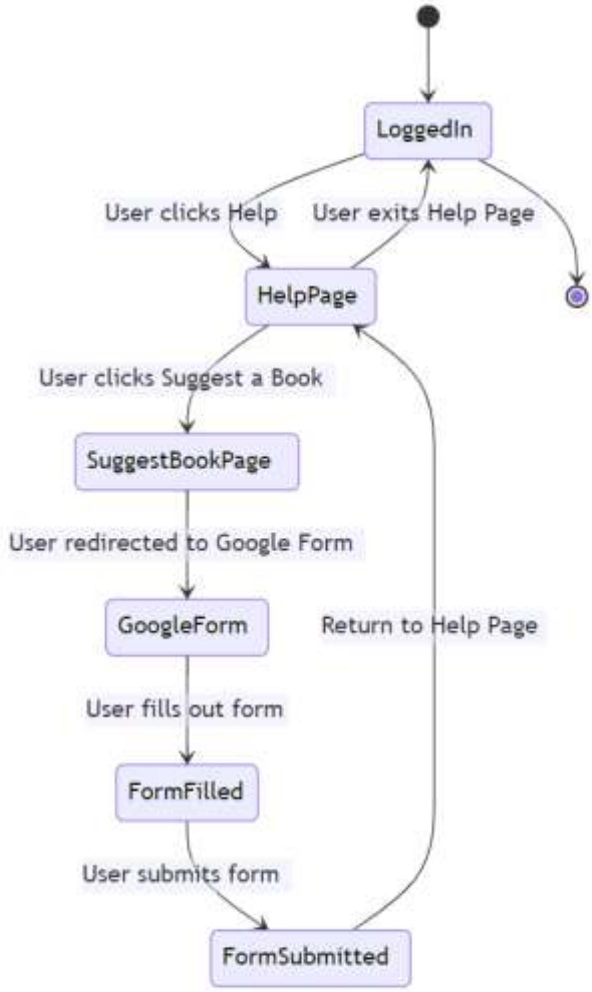
Use Case Details	**UC2: Filter Books**
Description	Allows the user to filter books in the catalog
Actors	User (Reader)
Pre-Condition(s)	- User is logged in.
Post-Condition(s)	A list of books matching the filters is displayed
Main Flow	1.User clicks ‘Filter Books’. 2. System asks for the filters. 3. User selects filters. 4. The system processes the request 5. The system displays the filtered books list.
Alternative(s)	At step 4 if the list size is 0 display no matches message.
Dependency	Requires integration with the book database.


Logged In
User Clicks on Filter
Filters Selections
User Selects Filters
Filterization
Else
Books > 0
No matches displayed
Books displayed

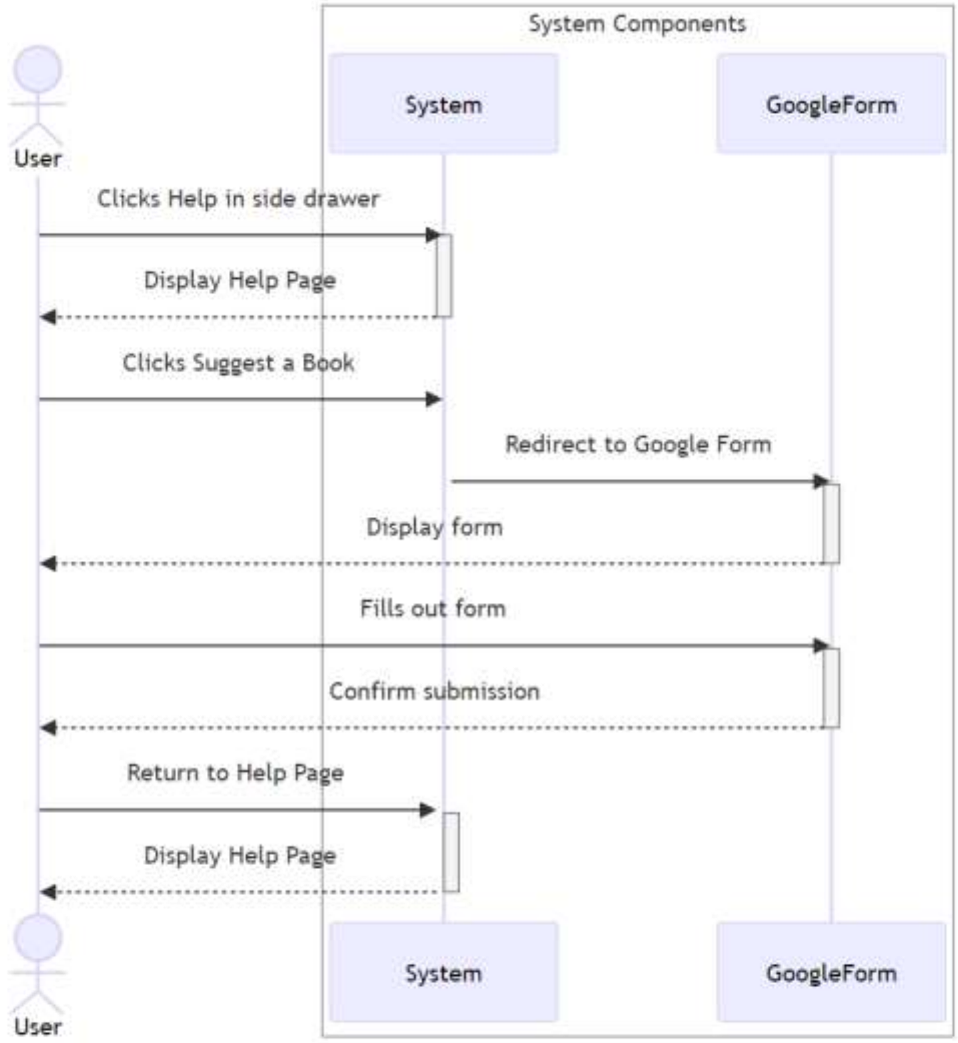


Use Case Details	**UC3: Suggest a Book**
Description	Allows the user to suggest a new book to be added to the app's catalog by filling out a Google form.
Actors	User
Pre-Condition(s)	• User is logged in. • User has accessed the side drawer.
Post-Condition(s)	The user has successfully submitted a book suggestion via a Google form.
Main Flow	1. User clicks on the Help button in the side drawer menu. 2. The system displays the Help page. 3. User clicks on the "Suggest a Book" button within the Help page. 4. The system redirects the user to a Google form. 5. User fills out the Google form with book details. 6. User submits the form.
Alternative(s)	• If the Google form fails to load, display an error message to the user. • If the user exits the form without submitting, no suggestion is made.
Dependency	Requires external dependency on Google Forms.

UC3 State Diagram

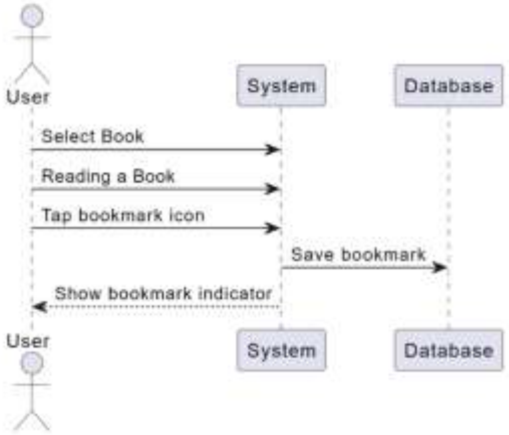


UC3 Sequence Diagram

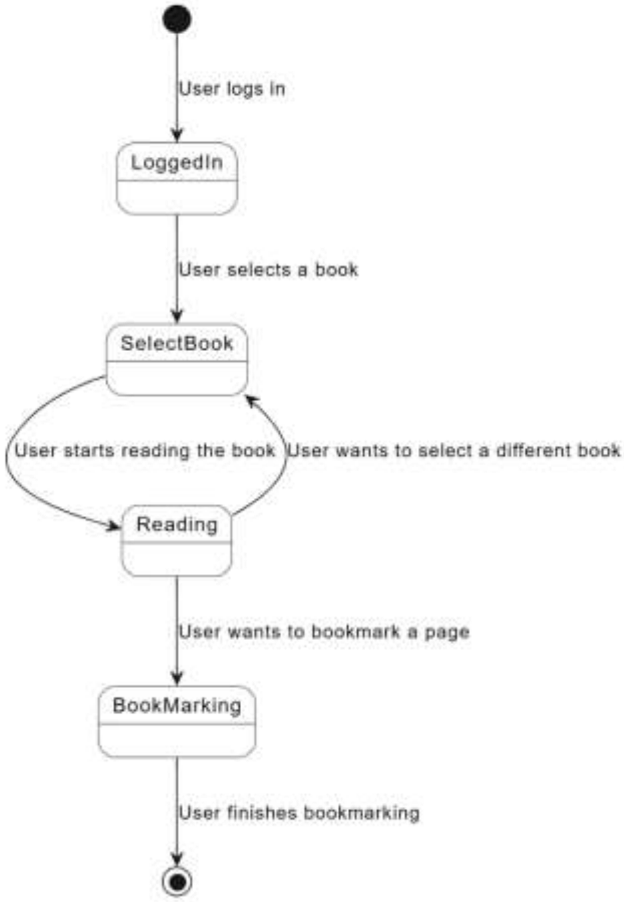


Use Case Details	**UC-04:** Bookmark a Page
Description	Allows user to save a bookmark for a specific page in a book they are reading.
Actors	User (Reader)
Pre-Condition(s)	- User is logged in and reading a book.
Post-Condition(s)	Bookmark is saved for the page.
Main Flow	1. User taps bookmark icon while on a page. 2. System saves bookmark for the page. 3. Bookmark indicator is shown on page.
Alternative(s)	
Dependency	Requires integration with the book database.

Sequence Diagram: "Muqin Mobile Reading App - Bookmark a Page Sequence Diagram"



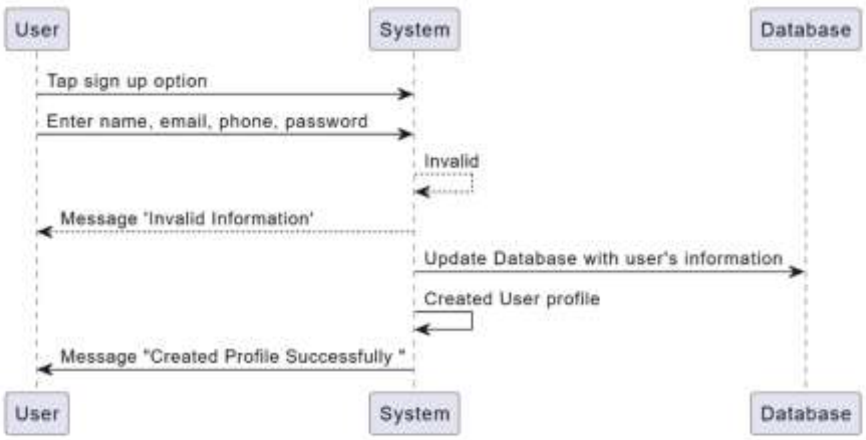
State Diagram: "Muqin Mobile Reading App - Bookmark a Page State Diagram"


User logs in
LoggedIn
User selects a book
SelectBook
User starts reading the book
User wants to select a different book
Reading
User wants to bookmark a page
BookMarking
User finishes bookmarking

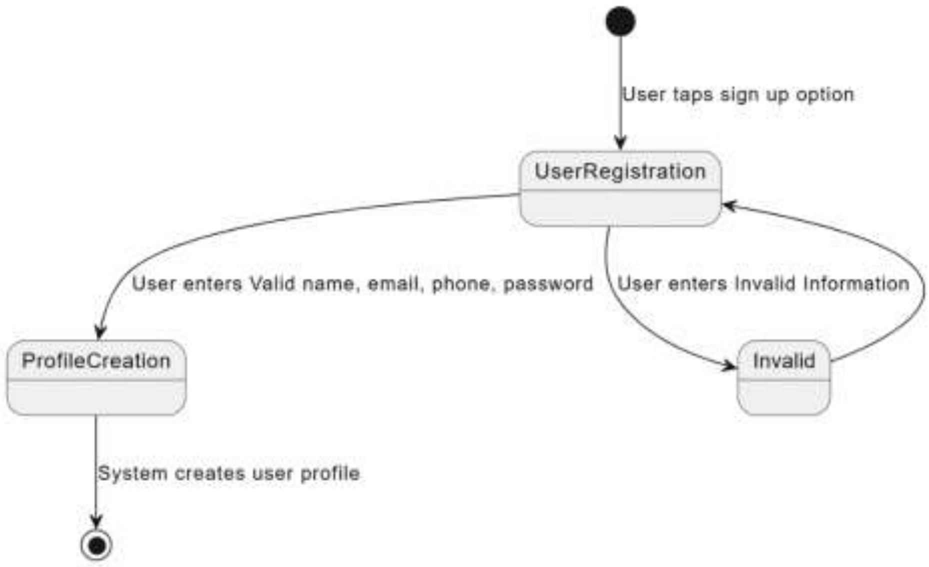
Use Case Details	**UC-05:** Create a Profile

Description	Allows user to create a user profile in the app.
Actors	User (Reader)
Pre-Condition(s)	None
Post-Condition(s)	User profile is created.
Main Flow	1. User taps sign up option. 2. User enters name, email, password. 3. System creates user profile.
Alternative(s)	
Dependency	Requires integration with the book database.

Sequence Diagram: "Muqin Mobile Reading App - Create a Profile Sequence Diagram"


User
System
Database
Tap sign up option
Enter name, email, phone, password
Invalid
Message 'Invalid Information'
Update Database with user's information
Created User profile
Message "Created Profile Successfully "
User
System
Database

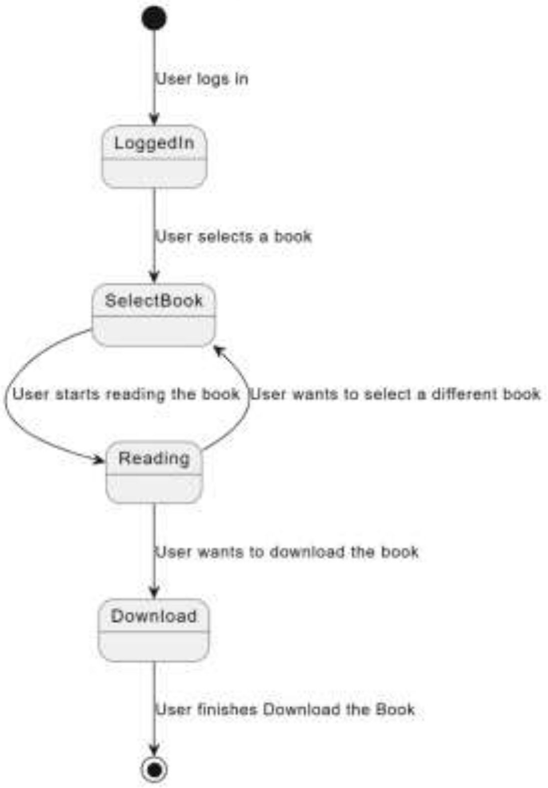
State Diagram: "Muqin Mobile Reading App - Create a Profile State Diagram"


User taps sign up option
UserRegistration
User enters Valid name, email, phone, password
User enters Invalid Information
ProfileCreation
Invalid
System creates user profile

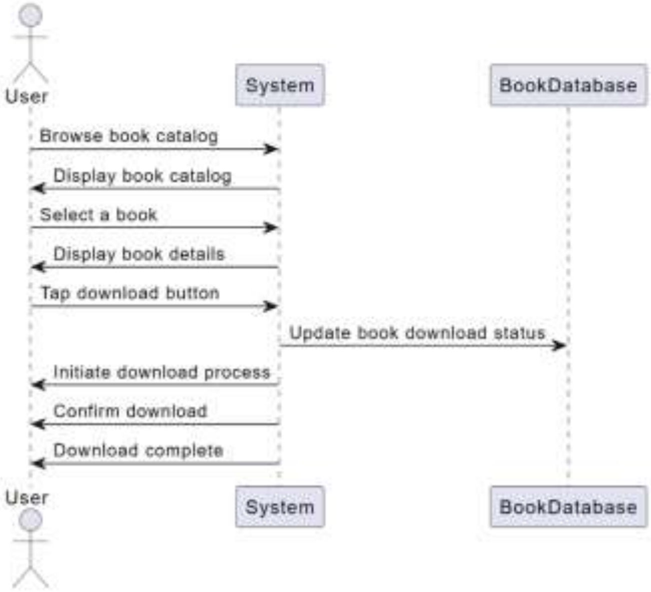
Use Case Details	**UC-06:** Download a Book
Description	Description: Allows user to download a book for offline Reading.
Actors	User (Reader)
Pre-Condition(s)	- User is logged in.
Post-Condition(s)	Book is downloaded locally.
Main Flow	1. User browses book catalog and selects a book.

	2. User taps download button for the book. 3. System downloads book content locally.
Alternative(s)	
Dependency	Requires integration with the book database, and enough free space on the device.

State Diagram: "Muqin Mobile Reading App - Download a Book State Diagram"


User logs in
LoggedIn
User selects a book
SelectBook
User starts reading the book
User wants to select a different book
Reading
User wants to download the book
Download
User finishes Download the Book

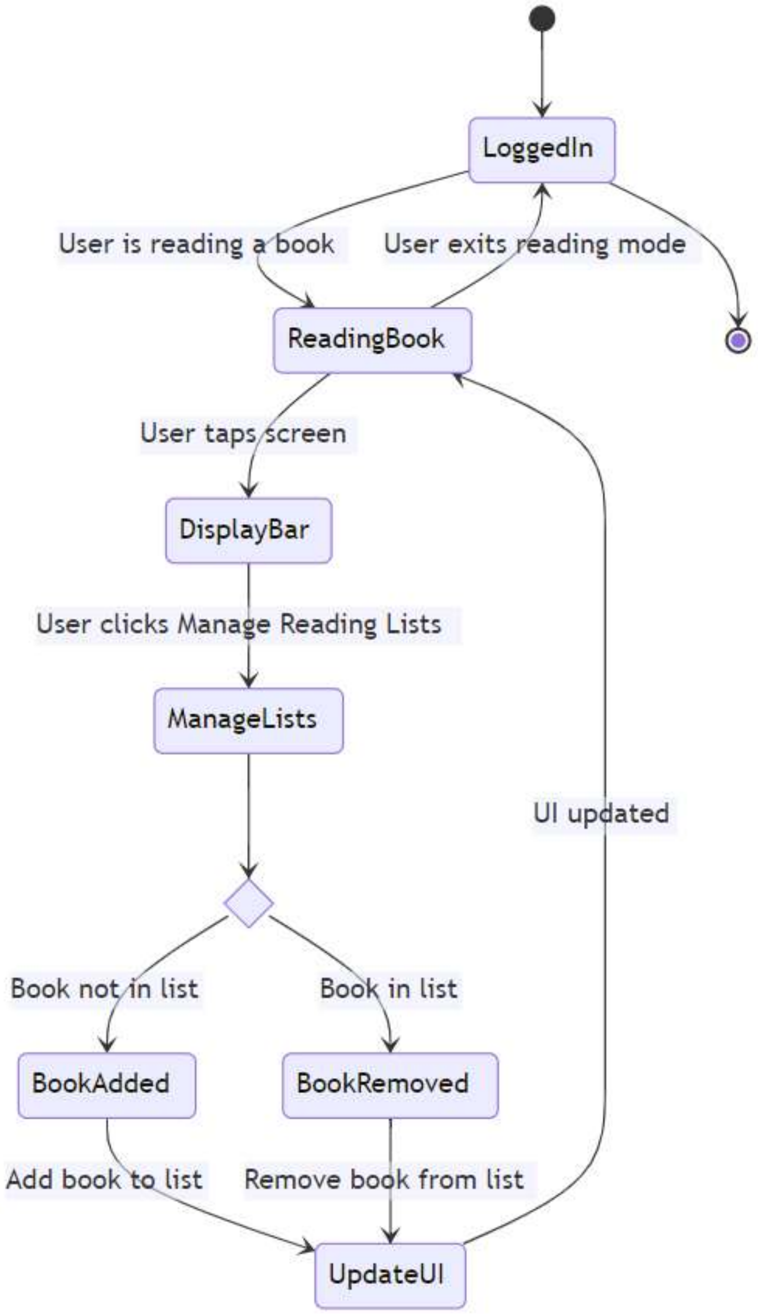
Sequence Diagram: "Muqin Mobile Reading App - Download a Book Sequence Diagram"



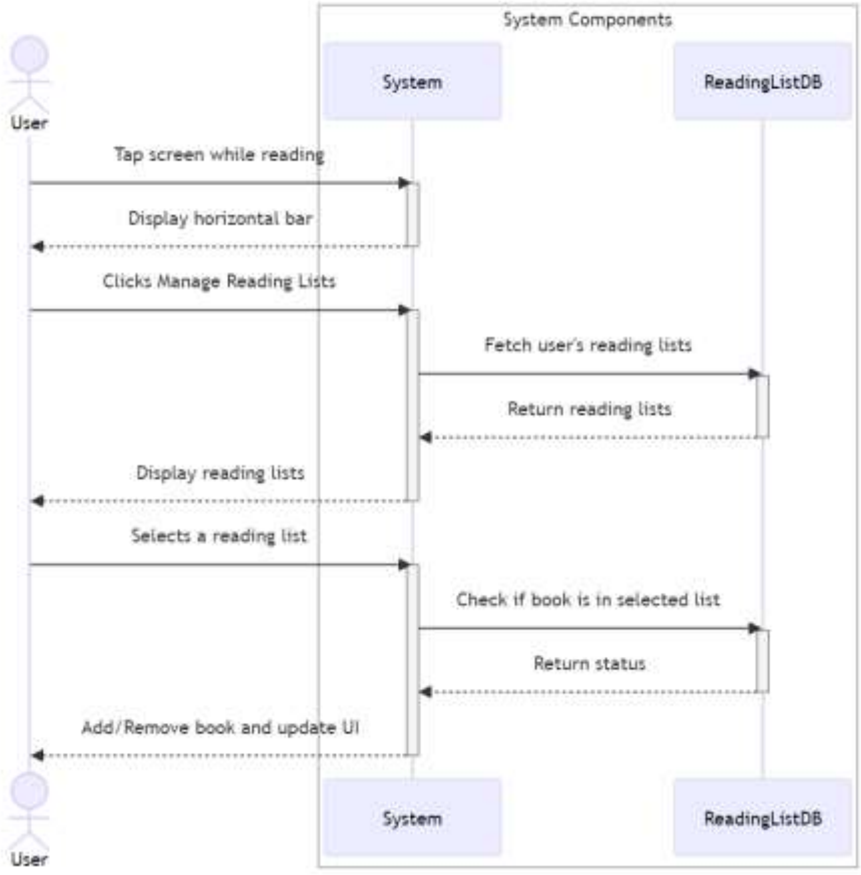
Use Case Details	**UC7-8: Add/Remove a Book from a Reading List.**
Description	Allows the user to manage their reading lists by either adding or removing a selected book while reading.

Actors	User (Reader)
Pre-Condition(s)	- User is logged in. - User is reading a book. - User has at least one reading list in their personal library.
Post-Condition(s)	The selected book is either added to or removed from the chosen reading list.
Main Flow	1. While reading a book, the user taps the screen to display the horizontal bar. 2. User clicks on the "Manage Reading Lists" button in the horizontal bar. 3. The system displays the user's reading lists. 4. User selects a reading list. 5. The system checks if the selected book is already in the chosen reading list. 6. If the book is not in the list, it is added; if it is, it is removed. 7. The UI updates to indicate the current status of the book in the selected reading list.
Alternative(s)	- If the reading list fails to load, display an error message to the user. - If the user has no reading lists, display a message prompting them to create one.
Dependency	Connection with the database for reading lists and the books is established.

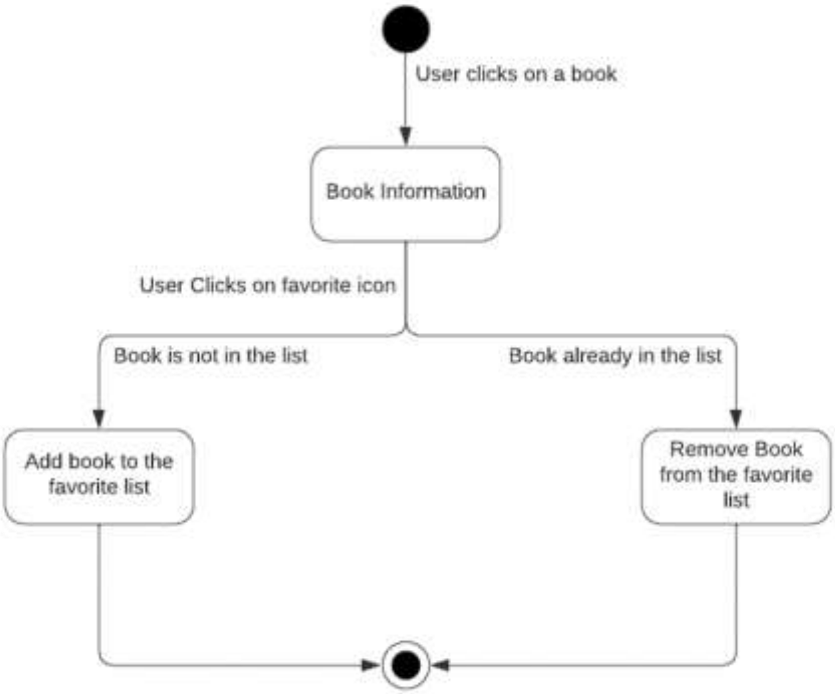
UC7-8 State Diagram

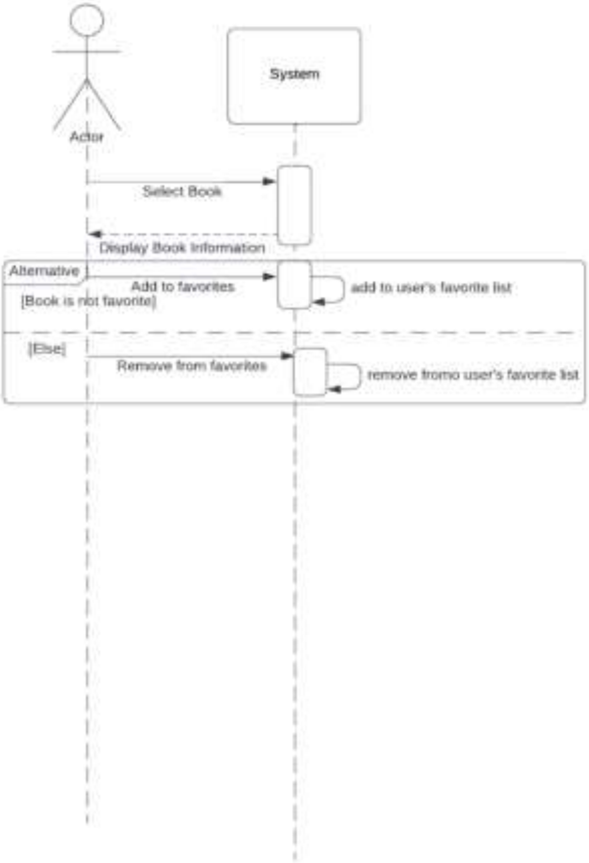


UC7-8 Sequence Diagram



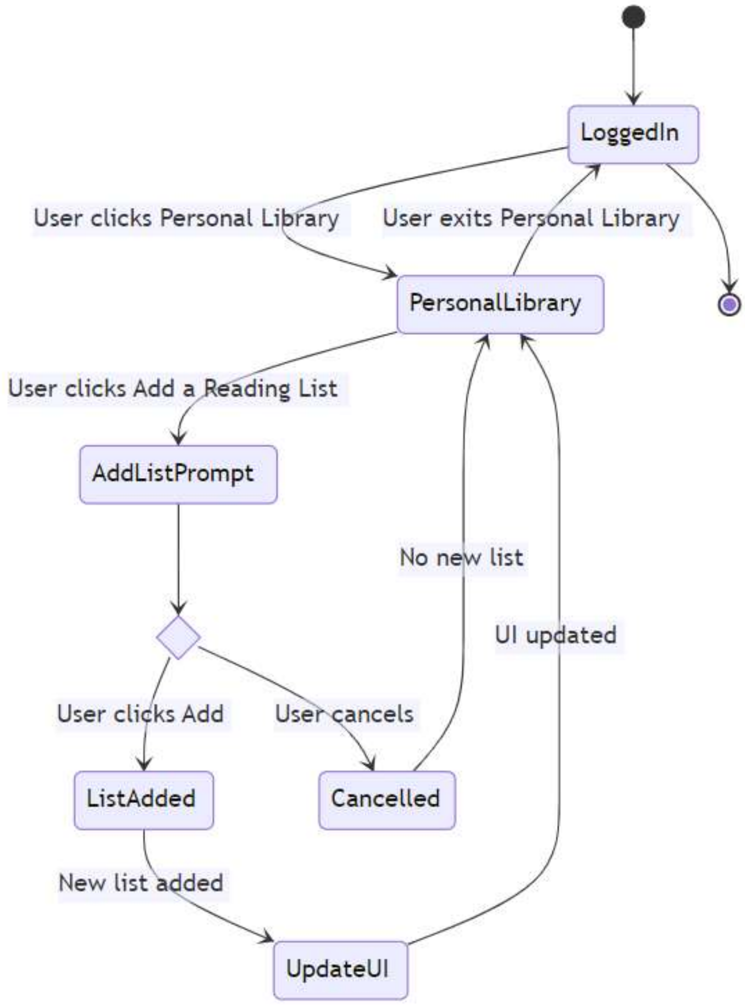
Use Case Details	**UC9-10: Add/remove a book from favorites.**
Description	Adding/removing books to favorites list
Actors	User (Reader)
Pre-Condition(s)	
Post-Condition(s)	The book is added/removed from the favorite list
Main Flow	1. User clicks on a book. 2. User clicks on the favorite icon. 3. System adds the book to the favorite list.
Alternative(s)	On step 3, if the book is already in the favorite list, remove the book from the list
Dependency	


User clicks on a book
Book Information
User Clicks on favorite icon
Book is not in the list
Book already in the list
Add book to the favorite list
Remove Book from the favorite list

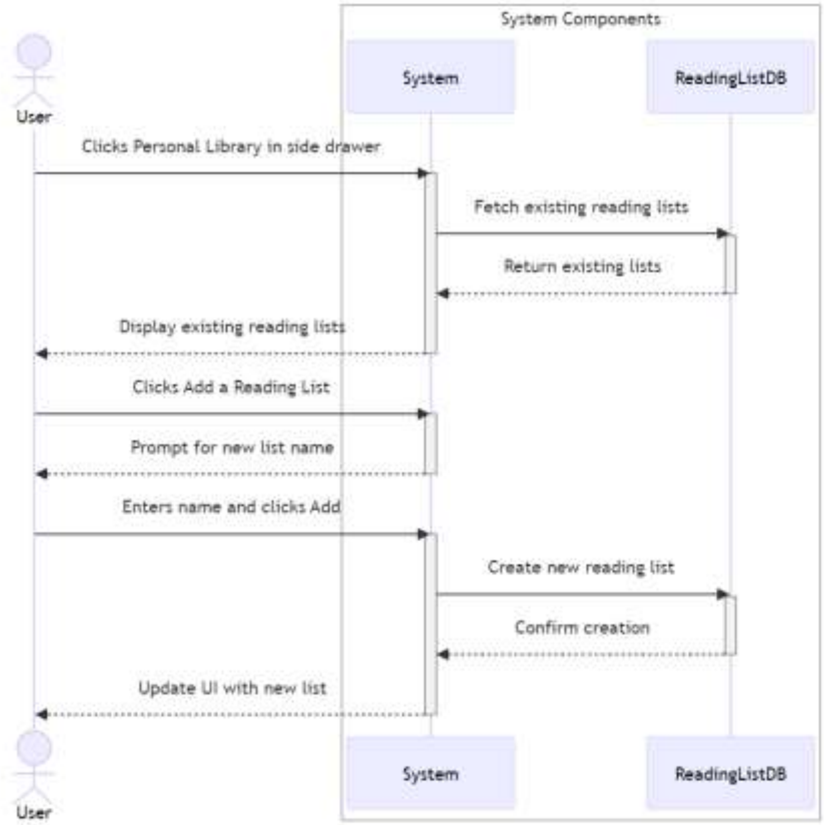

Actor
System
Select Book
Display Book Information
Alternative
Add to favorites
[Book is not favorite]
add to user's favorite list
[Else]
Remove from favorites
remove fromo user's favorite list

Use Case Details	**UC11: Adding a New Reading List.**
Description	Allows the user to create a new reading list within their personal library.
Actors	User (Reader)
Pre-Condition(s)	• User is logged in. • User has accessed the side drawer.
Post-Condition(s)	A new reading list is created and added to the user's personal library.
Main Flow	1. User clicks on the Personal Library button in the side drawer menu. 2. The system displays the user's existing reading lists. 3. User clicks on the "Add a Reading List" button. 4. A prompt appears asking the user to name the new reading list. 5. User enters a name for the new reading list. 6. User clicks the "Add" button. 7. The system creates the new reading list and adds it to the user's personal library. 8. The UI updates to display the newly created reading list.
Alternative(s)	• If the user cancels the operation, no new reading list is created. • If the user tries to create a reading list with a name that already exists, display an error message.
Dependency	Connection with the database for reading lists and the books is established.

UC11 State Diagram

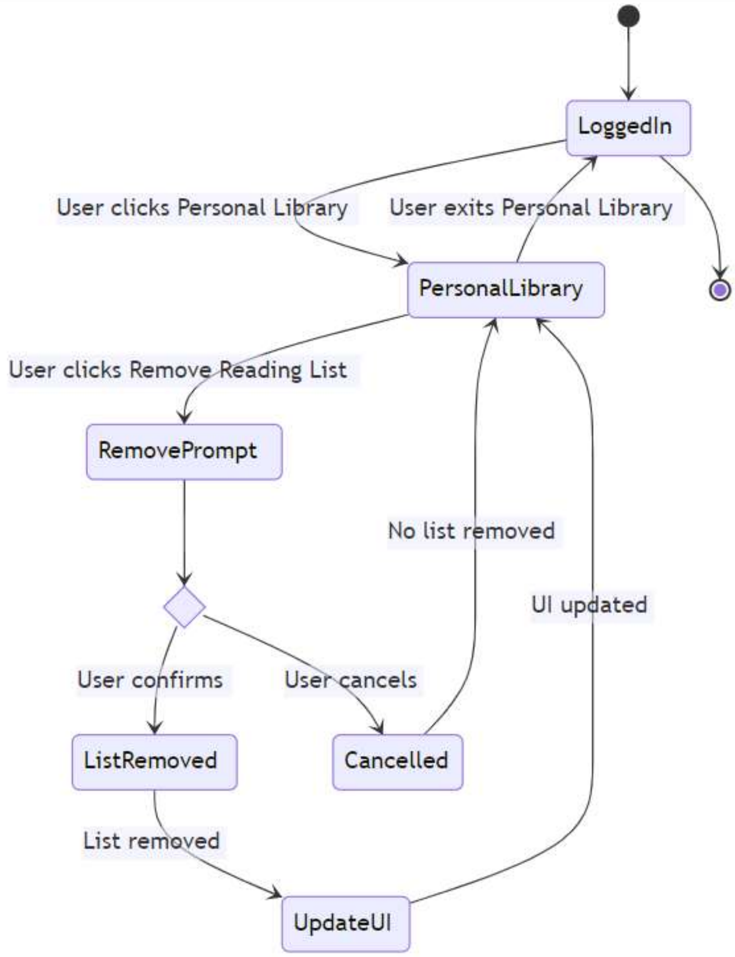


UC11 Sequence Diagram

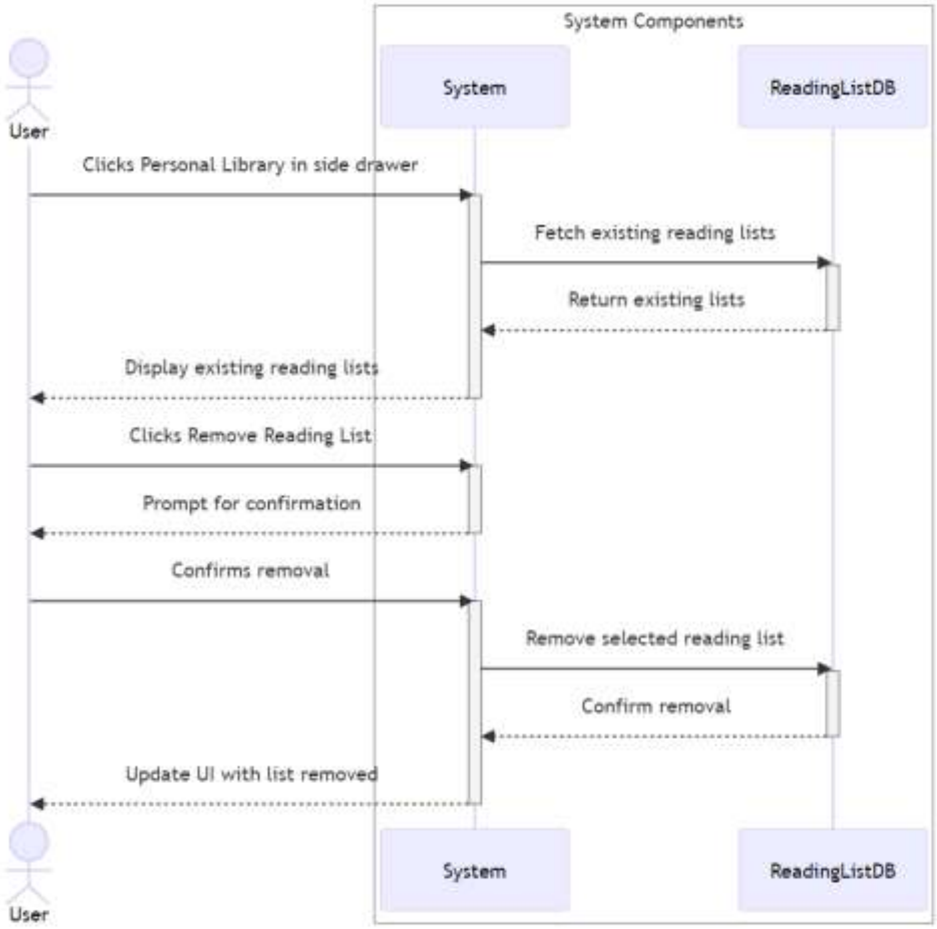


Use Case Details	**UC12: Adding a New Reading List.**
Description	Allows the user to create a new reading list within their personal library.
Actors	User (Reader)
Pre-Condition(s)	• User is logged in. • User has accessed the side drawer menu. • At least one reading list exists in the user's personal library.
Post-Condition(s)	The selected reading list is removed from the user's personal library.
Main Flow	1. User clicks on the Personal Library button in the side drawer menu. 2. The system displays the user's existing reading lists. 3. User clicks on the "Remove Reading List" button next to the desired list. 4. A confirmation prompt appears. 5. User confirms the removal. 6. The system removes the selected reading list from the user's personal library. 7. The UI updates to reflect the removal of the reading list.
Alternative(s)	• If the user cancels the operation, no reading list is removed.
Dependency	Connection with the database for reading lists and the books is established.

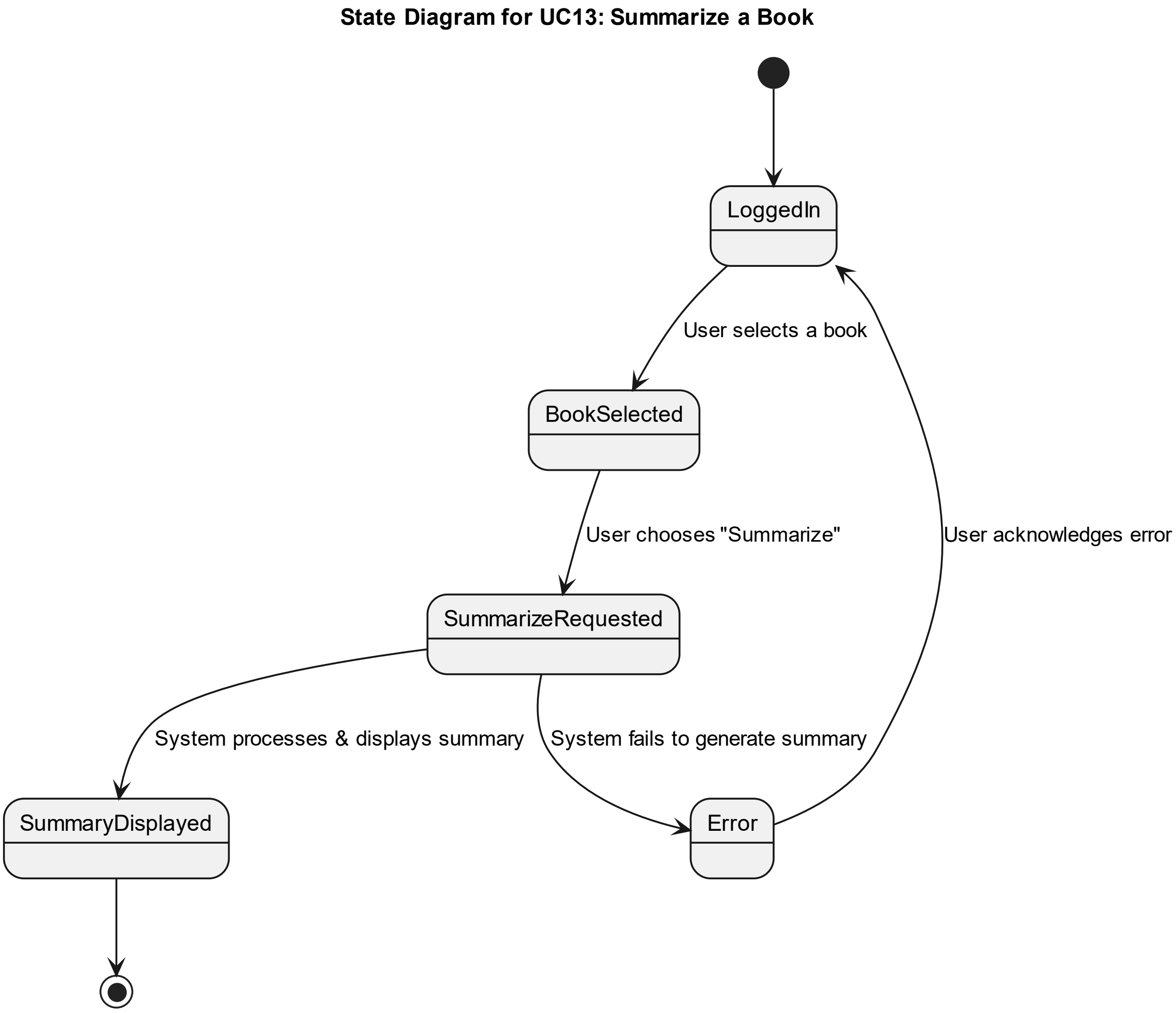
UC12 State Diagram

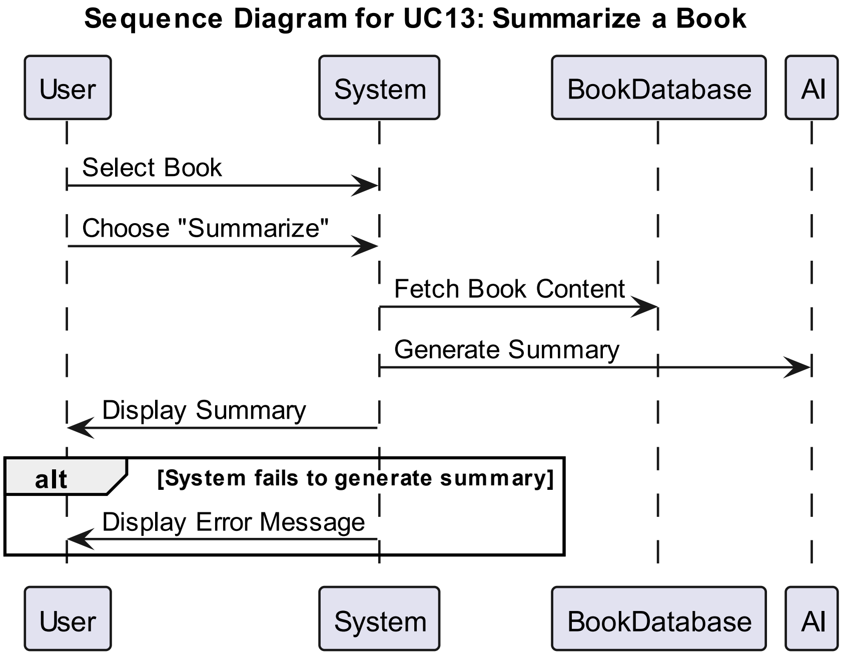


UC12 Sequence Diagram


System Components
User
System
ReadingListDB
Clicks Personal Library in side drawer
Fetch existing reading lists
Return existing lists
Display existing reading lists
Clicks Remove Reading List
Prompt for confirmation
Confirms removal
Remove selected reading list
Confirm removal
Update UI with list removed
User
System
ReadingListDB

Use Case Details	**UC13: Summarize a Book**
Description	Allows the user to generate a concise summary of a selected book.
Actors	User (Reader)
Pre-Condition(s)	- User is logged in. - The book is available in the user's library or the main catalog.
Post-Condition(s)	A summary of the book is displayed to the user.
Main Flow	1. User selects a book. 2. User chooses the "Summarize" option. 3. The system processes the book content. 4. The system displays the summarized content to the user.
Alternative(s)	If the system fails to generate a summary, display an error message to the user.
Dependency	Requires integration with the book database.

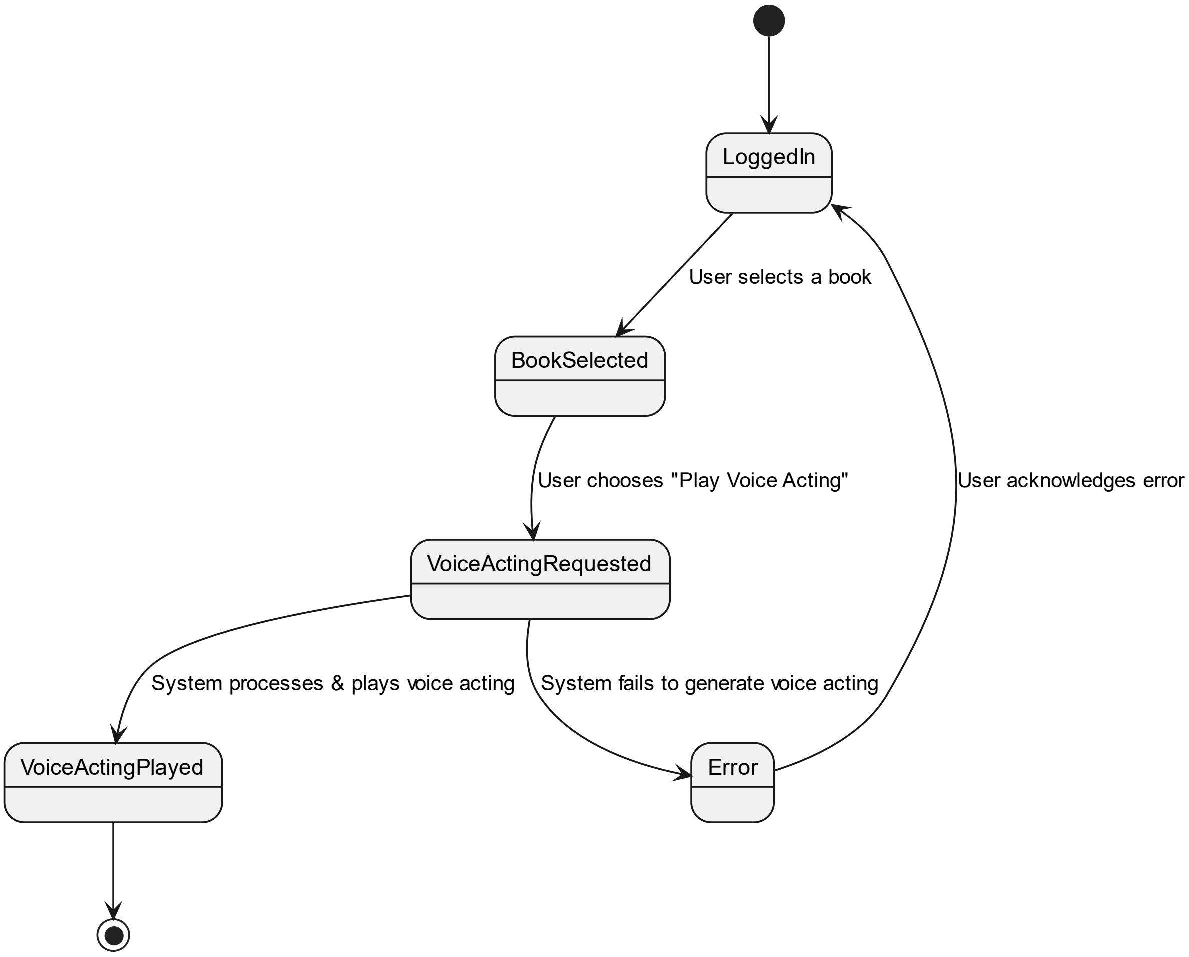




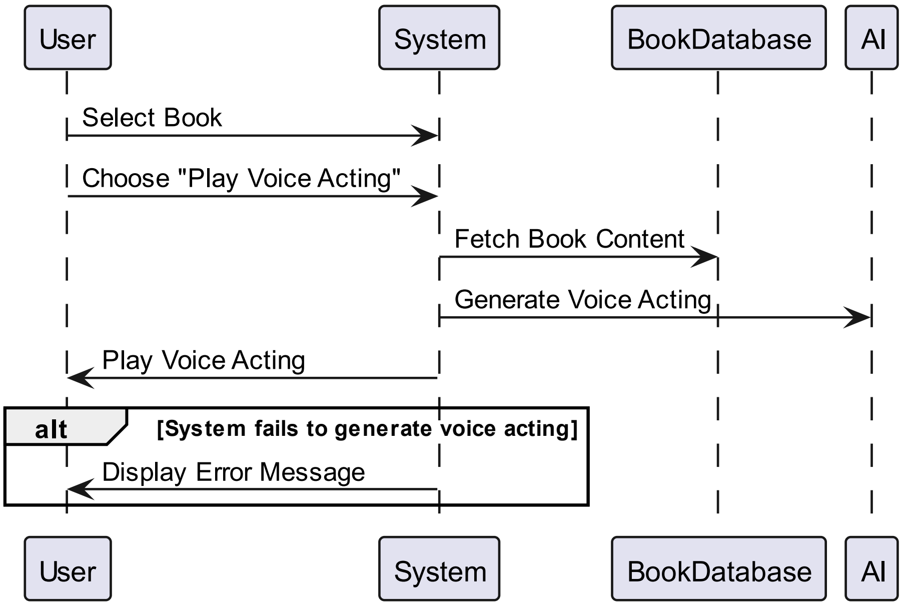
Use Case Details	UC14: Play Voice Acting of a Book

Description	Allows the user to listen to an AI-generated voice acting of the selected book.
Actors	User (Reader)
Pre-Condition(s)	- User is logged in. - The book is available in the user's library or the main catalog.
Post-Condition(s)	The book's content is played audibly to the user.
Main Flow	1. User selects a book. 2. User chooses the "Play Voice Acting" option. 3. The system processes the book content. 4. The system audibly plays the book content to the user.
Alternative(s)	If the system fails to generate voice acting, display an error message to the user.
Dependency	Requires integration with the book database and AI voice acting system.

State Diagram for UC14: Play Voice Acting of a Book

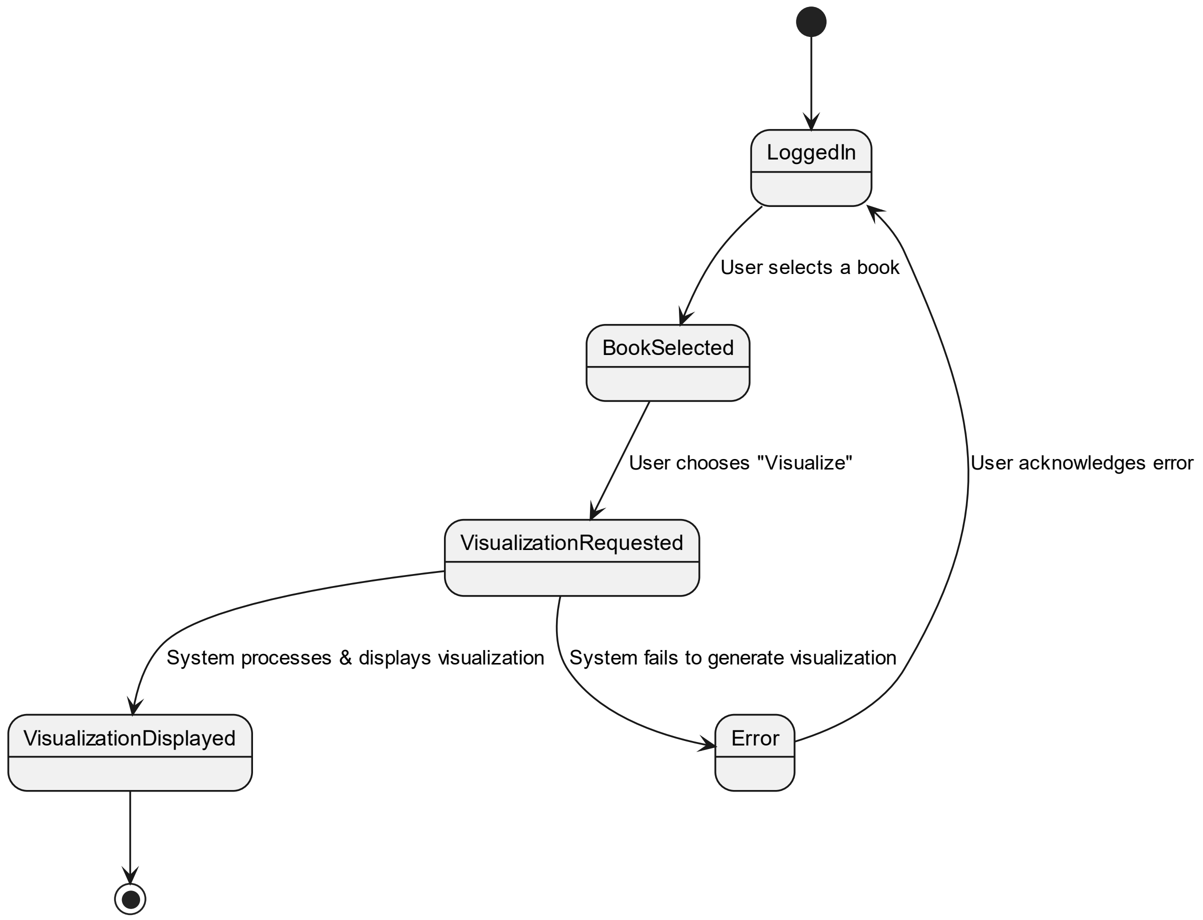

LoggedIn
User selects a book
BookSelected
User chooses "Play Voice Acting"
User acknowledges error
VoiceActingRequested
System processes & plays voice acting
System fails to generate voice acting
VoiceActingPlayed
Error

Sequence Diagram for UC14: Play Voice Acting of a Book

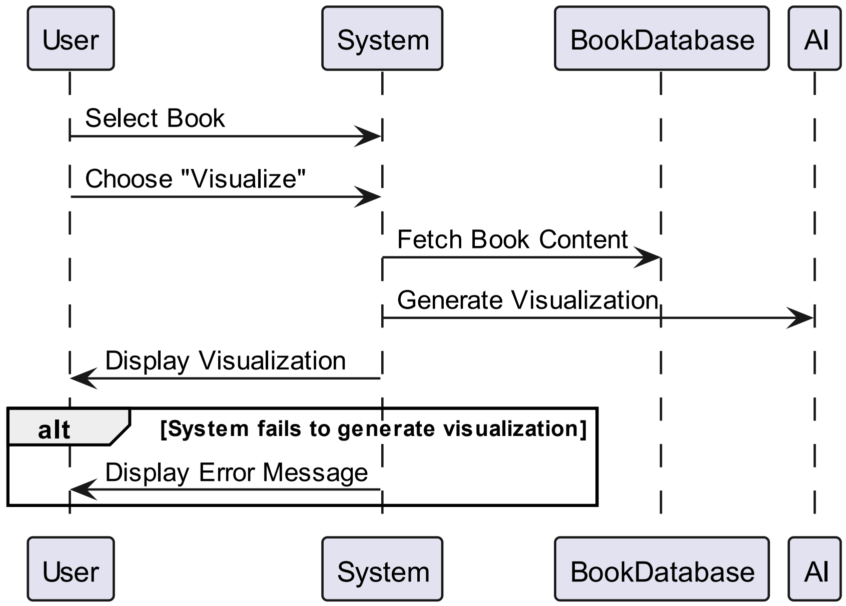


Use Case Details	**UC15: Visualize a Book**
Description	Allows the user to view AI-generated visualizations related to the content of the selected book.
Actors	User (Reader)
Pre-Condition(s)	- User is logged in. - The book is available in the user's library or the main catalog.
Post-Condition(s)	Visualizations related to the book's content are displayed to the user.
Main Flow	1. User selects a book. 2. User chooses the "Visualize" option. 3. The system processes the book content. 4. The system displays visualizations related to the book content to the user.
Alternative(s)	If the system fails to generate visualizations, display an error message to the user.
Dependency	Requires integration with the book database and AI visualization system.

State Diagram for UC15: Visualize a Book



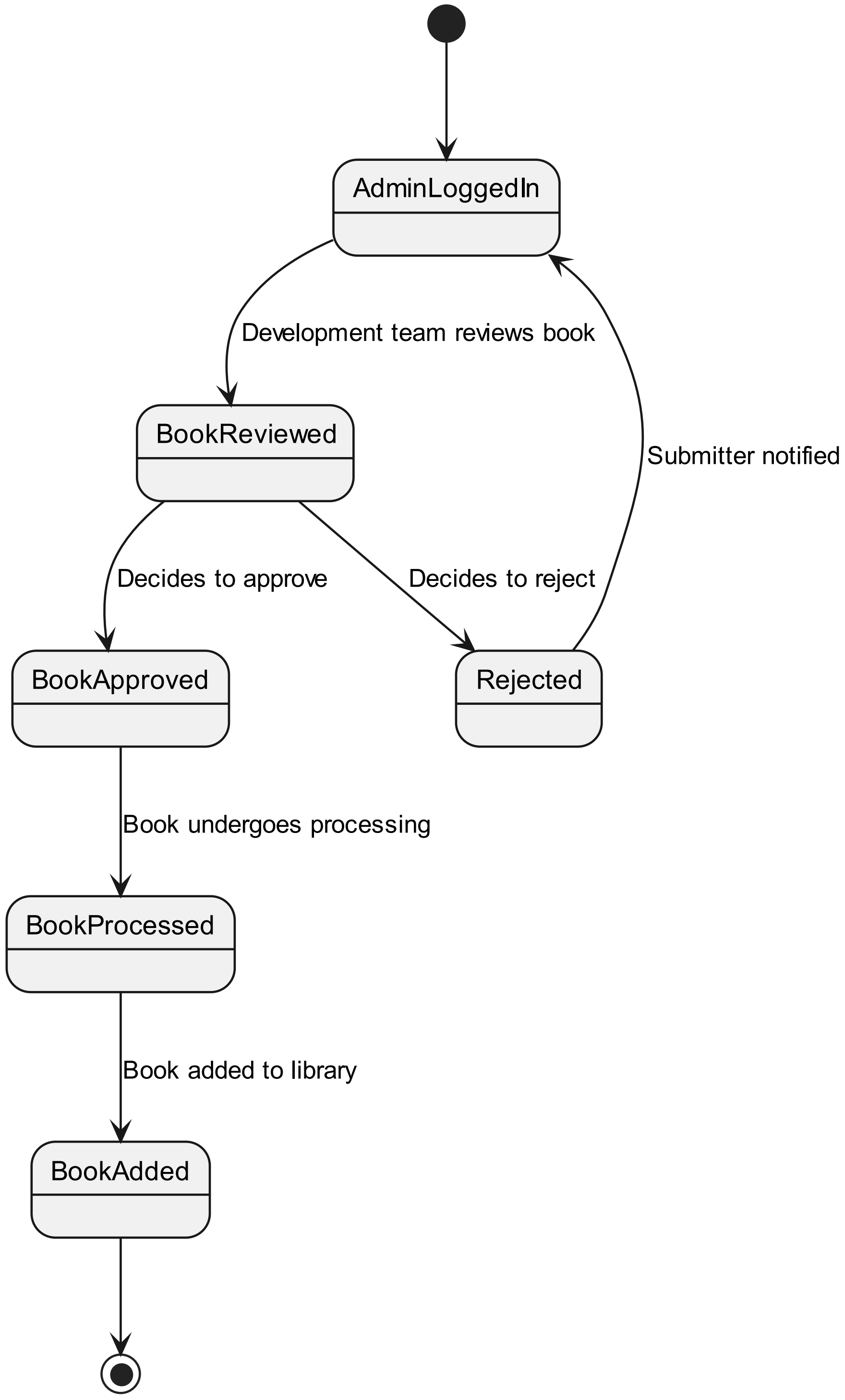
Sequence Diagram for UC15: Visualize a Book

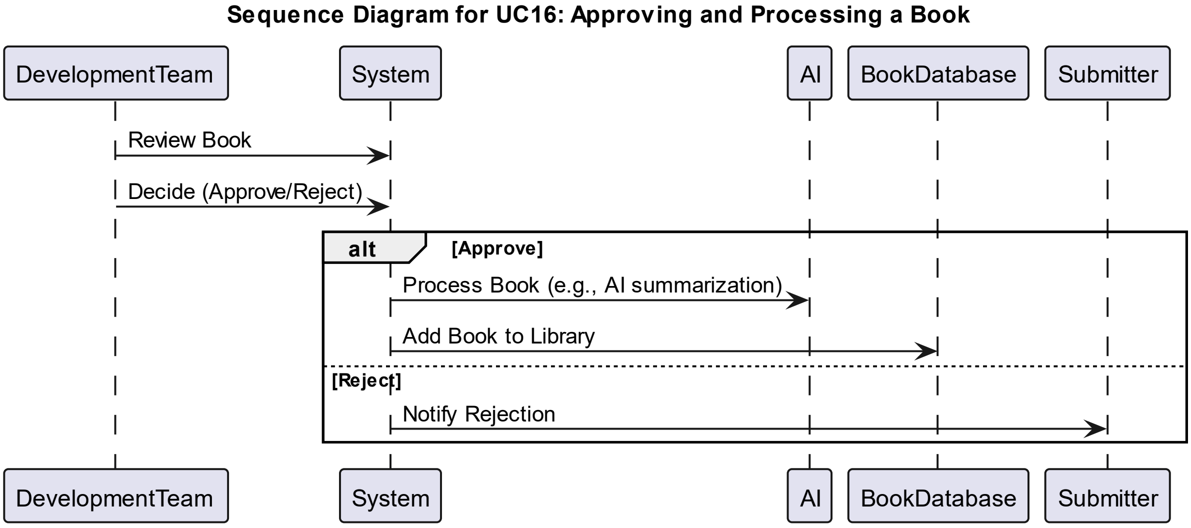


Use Case Details	**UC16: Approving and Processing a Book**
Description	Allows the development team to review, approve, and process books for inclusion in the Muqin library.

Actors	Development Team
Pre-Condition(s)	A member of the development team is logged in with appropriate permissions.
Post-Condition(s)	The book is either approved for inclusion in the library or rejected.
Main Flow	1. Development team member reviews the details and content of a submitted book. 2. Decides to approve or reject the book. 3. If approved, the book undergoes processing (e.g., AI summarization, voice acting preparation). 4. The book is added to the Muqin library and made available to users.
Alternative(s)	If the book is rejected, the submitter is notified of the decision and provided with feedback if necessary.
Dependency	Requires administrative access and integration with the book submission system.

State Diagram for UC16: Approving and Processing a Book




Sequence Diagram for UC16: Approving and Processing a Book
DevelopmentTeam
System
AI
BookDatabase
Submitter
Review Book
Decide (Approve/Reject)
alt
[Approve]
Process Book (e.g., AI summarization)
Add Book to Library
[Reject]
Notify Rejection
DevelopmentTeam
System
AI
BookDatabase
Submitter

3.5 Performance Requirements

This subsection specifies both the static and the dynamic numerical requirements. Static numerical requirements such as the number of simultaneous users to be supported. Dynamic numerical requirements such as the numbers of transactions and tasks and the amount of data to be processed within certain time period for both normal and peak workload conditions.

All of these requirements should be stated in measurable terms. For example,
"95% of the transactions shall be processed in less than 1 second"

1. The system must support a minimum of 100 simultaneous users during peak usage hours.
2. The app should load on a user's device within 6 seconds of launching.
3. User authentication and login should take no longer than 2 seconds.
4. The app must support a library size of at least 50 books per user.
5. User profiles and bookshelves should load in under 5 seconds.
6. 95% of search queries should return results within 4 seconds.
7. The app should be able to handle a minimum of 20 book downloads per minute.
8. Book requests by users should be processed within 3 days of submission.
9. Data backups and maintenance operations should not shut down the system for more than 15 minutes.
10. The app should have an uptime of at least 99.9% over a rolling 30-day period, excluding scheduled maintenance.
11. In case of system failure, the app should recover and resume normal operation within 30 minutes.
12. New book additions to the catalog should be visible to users within 15 minutes of publishing.
13. Updates to book metadata, such as author or genre changes, should be reflected within 1 hour.

3.6 Logical Database Requirements

This section specifies the logical requirements for any information that is to be placed into a database.

- **User Authentication and Authorization:**
- User credentials, including usernames and encrypted passwords, must be securely stored.
- User profiles should include information such as reading history and preferences.
- **Book Catalog:**
- Book details, including title, author, genre, synopsis, and cover image, should be stored.
- User-specific book data, such as reading progress and bookmarks, must be associated with individual profiles.
- **Library Management:**
- User-created lists and categories for book organization should be stored.
- **Text-To-Speech and Visualization:**
- Data related to text-to-speech processing and content visualization should be stored for seamless user experience.

Types of Informatio:	Book Information	User Information	TTS & Ai (AI-Generated Content)

Data Entities and Their Relationships:
Users are related to their downloaded books, bookmarks, and feedback.
Books are related to user feedback and user reading progress.
Downloaded books are linked to users.
AI-Generated Content is associated with specific books and user interactions.

Relationships:

User to Book (e.g., downloaded books, reading progress,Bookmark,etc..)
User to Feedback (e.g., user reviews and ratings)
Book to AI-Generated Content (e.g., TTS scripts, AR visuals)

3.7 Software System Attributes

These are also known as non-functional requirements or quality attributes. It is important that required attributes by specified so that their achievement can be objectively verified. A partial list is added as subsections. You can add and remove these requirements according to your choice.

3.7.1 Security

Specify the factors that would protect the software from accidental or malicious access, use, modification, destruction, or disclosure.

- **Authentication:** The Muqin app will implement strong user authentication mechanisms, ensuring that only authorized users can access their accounts. This may include multi-factor authentication (MFA) for added security.
- **Data Encryption:** All sensitive user data, including personal information and reading history, will be encrypted both in transit (using protocols like TLS) and at rest in the database.
- **Secure APIs:** The app will ensure that all APIs used, whether for fetching book content or for other functionalities, are secure and not susceptible to attacks like SQL injection or cross-site scripting.
- **Permissions:** The app will request only necessary permissions from the user's device (e.g., storage access for downloading books).

3.7.2 Reliability

Specify the factors required to establish the required reliability of the software system at time of delivery.

- **Error Handling:** The Muqin app will have robust error handling to ensure that minor issues do not cause the app to crash. Users will receive clear error messages guiding them on what went wrong and potential steps to resolve it.
- **Backup and Recovery:** User data, including bookmarks, reading lists, and preferences, will be regularly backed up to prevent data loss. In the event of any failures, there will be mechanisms in place for data recovery.
- **Load Balancing:** If the app relies on servers for content delivery or other functionalities, load balancers will be used to ensure that high user volumes do not lead to server crashes or slow performance.
- **Offline Mode:** Recognizing that users might not always have a stable internet connection, the Muqin app will provide an offline mode where users can access downloaded content seamlessly.

3.7.3 Availability

Specify the factors required to guarantee a defined availability level for the entire system such as checkpoint, recovery, and restart.

- **Cloud Infrastructure:** If the Muqin app relies on cloud servers for any functionalities, it will ensure that these servers have high uptime guarantees, ensuring that the app is always available for users.

- **Content Delivery Network (CDN):** To ensure that users can quickly access content, especially if they are spread globally, the app might use a CDN to reduce latency.

- **Redundancy:** Critical components of the app's infrastructure will have redundancies in place. If one component fails, the backup will take over, ensuring uninterrupted service.

- **Monitoring and Alerts:** The Muqin app's technical team will continuously monitor the app's performance and will be alerted to any issues that might affect availability, allowing for quick resolution.

3.7.4 Maintainability

Specify attributes of software that relate to the ease of maintenance of the software itself. There may be some requirement for certain modularity, interfaces, complexity, etc.

- **Modularity:**
- The software shall be designed with a modular structure, allowing individual components to be modified or updated without affecting the entire system.
- Each functional module should have well-defined interfaces, minimizing dependencies.
- **Code Documentation:**
- Comprehensive and well-maintained documentation shall accompany the source code, providing insights into the functionality, purpose, and usage of each module.
- **Version Control:**
- Utilize version control systems (e.g., Git) to track changes in the source code, enabling efficient collaboration among developers and ensuring the ability to roll back to previous versions if needed.
- **Logging and Monitoring:**
- Implement logging mechanisms to record errors, warnings, and user activities for effective troubleshooting and analysis.
- Set up monitoring tools to detect performance issues and potential areas for improvement.
- **Update Procedures:**
- Establish clear procedures for releasing updates and patches, ensuring a smooth and controlled update process.
- Provide mechanisms for notifying users about available updates and guiding them through the update process.

3.7.5 Portability

Specify attributes of software that relate to the ease of porting the software to other host machines and/or operating systems. These may include use of portable language like java, use of well-known compilers or even Operating Systems.

- **Cross-Platform Compatibility:**
- Design the software to be compatible with multiple platforms, including Android and iOS, ensuring a broad user reach.
- **Portable Language:**
- Utilize programming languages that support portability, like Flutter.
- **Well-Known Compilers:**
- Write code adhering to standard coding practices and using well-known compilers, enhancing the software's compatibility with different systems.
- **Operating System Independence:**
- Minimize dependencies on specific operating system features, making the software less reliant on unique functionalities of a particular OS.
- **API Compatibility:**
- Ensure that the software's interactions with external APIs follow industry standards, facilitating integration with various systems and services.
- **Adherence to Standards:**
- Follow industry standards and protocols, promoting compatibility and ease of integration with different hardware and software configurations.

Appendix

The Appendices are not always considered part of the actual requirements specification and are not always necessary. They may include:

- *Supporting or background information that can help the readers of the SRS*
- *A description of the problems to be solved by the software*
- *Special packaging instructions for the code and the media to meet security, export, initial loading, or other requirements*

A. Supporting or Background Information

1. **Focus on Arabic Content**: Muqin is designed specifically to cater to Arabic speakers. It aims to enrich the Arabic digital reading landscape by offering a wide array of literature and articles. English content will be also highly included.
2. **Target Audience**: The app targets a broad audience, ranging from young readers to adults, as well as students and language learners. This inclusivity aims to address the different reading needs and preferences of the Arabic-speaking community.
3. **Core Objectives**: Muqin's primary goal is to address the issue of low readership among Arabic speakers. The application aims to make reading more accessible and engaging for this demographic.

B. Description of Problems to be Solved

1. **Low Readership**: One of the significant issues Muqin aims to solve is the low engagement with literature among Arabic speakers. This has far-reaching implications, including a lack of exposure to diverse ideas and knowledge.
2. **Limited Platforms for Arabic Content**: Existing platforms may not cater adequately to Arabic content, creating a barrier for potential readers. Muqin aims to fill this gap by offering a dedicated platform for Arabic literature and articles.

C. Special Packaging Instructions

1. **Export Requirements**: If the application is to be made available in different countries, compliance with local and international data regulations should be considered while packaging the app.
2. **Initial Loading**: A user-friendly onboarding process should be packaged along with the main application, guiding new users through the features and functionalities of Muqin.
3. **Updates and Patches**: The packaging should allow for seamless updates and patches without requiring complete reinstallation.