VIETNAM NATIONAL UNIVERSITY OF HOCHIMINH CITY
THE INTERNATIONAL UNIVERSITY
SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

WEBSITE PLATFORM FOR ENJOYING PODCASTS AND READING HEALING BOOKS

By
Phạm Đức Đạt
Nguyễn Huỳnh Thảo My

A report submitted to the School of Computer Science and Engineering
in partial fulfillment of the requirements for Web App Development Subject
(IT093IU)

ACKNOWLEGMENTS

I would like to express my sincere gratitude to Associate Professor Dr. Nguyen Van Sinh, who taught the Web App Development course. His guidance and expertise have been invaluable to my learning journey.

I am also thankful to Mr. Nguyen Trung Nghia, MSc, who taught the lab sessions. His dedication and support have greatly enhanced my understanding and practical skills in the field.

TABLE OF CONTENTS

Contents

ACKNOWLEGMENTS 2
TABLE OF CONTENTS 3
LIST OF FIGURES 5
ABSTRACT 6
CHAPTER 1 7
INTRODUCTION 7
1.1. Background 7
1.2. About us 7
1.3. Development process 7
1.4. Development environment 8
CHAPTER 2 10
REQUIREMENT ANALYSIS AND DESIGN 10
2.1. REQUIREMENT ANALYSIS 10
2.1.1. Use case diagram 10
2.1.2. User specification 11
Use Case 1 11
Use Case 2 12
Use Case 3 13
Use Case 4 13
Use Case 5 14
Use Case 6 15
Use Case 7 16
Use Case 8 16
Use Case 9 17
Use Case 10 18
Use Case 11 19
Use Case 12 19
Use Case 13 20
2.2. Functional requirements 21
2.3. Non-Functional Requirements 22
2.4. Design 24
CHAPTER 3 38
IMPLEMENT 38
3.1. Discovery page 38
3.2. Login page 38

3.3. Register page 39
3.4. Listening podcast page 40
3.5. Reading book page 41
CHAPTER 4 43
DISCUSSION AND CONCLUSION 43
4.1. Discussion 43
4.2. Conclusion 43
REFERENCES 44

LIST OF FIGURES

Figure 1 The Scrum Agile methodology used in software development. 8
Figure 2 The Model-View-Controller (MVC) architectural pattern used in software development..... 9
Figure 3 Use case diagram..... 10
Figure 4 User specification for Listen to Podcasts 12
Figure 5 User specification for Listen to Audiobooks 12
Figure 6 User specification for Read Audiobooks 13
Figure 7 User specification for Browse Content..... 14
Figure 8 User specification for Name Search Content 15
Figure 9 User specification for Create Playlists..... 15
Figure 10 User specification for Favorite Content 16
Figure 11 User specfication for Upload Podcasts (Admin) 17
Figure 12 User specification for Upload Audiobooks (Admin) 18
Figure 13 User specification for Edit Content (Admin) 18
Figure 14 User specification for Delete Content (Admin) 19
Figure 15 User specification for Manage Users (Admin)..... 20
Figure 16 User specification for Login the system 21
Figure 17 System architechture model 24
Figure 18 Class Diagram 25
Figure 19 Use case diagram..... 26
Figure 20 Sequence diagram for Listening to Podcasts..... 27
Figure 21 Sequence diagram for Listening to Audiobooks 28
Figure 22 Sequence diagram for Reading Audiobooks..... 29
Figure 23 Sequence diagram for Browsing Content 30
Figure 24 Sequence diagram for Searching Content..... 30
Figure 25 Sequence diagram for Creating Playlists 31
Figure 26 Sequence diagram for Favoriting Content..... 32
Figure 27 Sequence diagram for Uploading Podcasts (Admin) 33
Figure 28 Sequence diagram for Uploading Audiobooks (Admin)..... 33
Figure 29 Sequence diagram for Editing Content (Admin)..... 34
Figure 30 Sequence diagram for Deleting Content (Admin)..... 35
Figure 31 Sequence diagram for Managing Users (Admin) 36
Figure 32 Sequence diagram for login..... 37
Figure 33 Discovery page..... 38
Figure 34 Login page..... 39
Figure 35 Register page..... 39
Figure 36 Podcast page with light mode. 40
Figure 37 Podcast page with dark mode..... 40
Figure 38 List of podcast page. 41
Figure 39 Reading book page with dark mode..... 41
Figure 40 Reading book page with light mode..... 42

ABSTRACT

In the following report, AVO – an online platform created by AVO Team, a firm focusing on web application design and development – is illustrated. In this case, AVO can be considered as a service that helps with healing through both recommended books or podcasts with a focus on individual development or mental health. The platform is easy to use since it offers guide interface and an easy navigation on both desktop and mobile versions and is also available in multiple languages.

Its features include listening to podcasts and books, adjusting font color, saving current reading position, creating lists, as well as receiving suggested listening. The admin tool offers the features of adding, editing and deleting the content within the site, as well as the creation, modification and deletion of the users that exist in the site; this is to ensure that the site remains effective in delivering its content and remain relevant to the users. Thus, it can be concluded that not only does AVO help with the individual transformations and growth but also stands as the example of up-to-scratch web development practices when it comes to providing the user with deep and immersive online experiences.

This brief outlines how the AVO project has been developed, the aims of the development as well as the method that has been employed in the development of the project, and finally the technical architecture and usability of the ideas put in the project.

CHAPTER 1

INTRODUCTION

1.1. Background

AVO is an enriching online platform designed to offer healing through the power of reading and listening. This unique website provides users with a vast collection of books and podcasts focused on healing, personal growth, and mental wellness. The platform is meticulously crafted to help individuals find solace and inspiration through content that nurtures the soul and fosters a peaceful mind.

The user experience on AVO is streamlined and intuitive, making it easy for anyone to navigate and access the healing resources they need. Whether on a desktop or a mobile device, users can effortlessly browse through an expansive library of books and podcasts. Each piece of content is carefully selected to ensure it brings value and support to the listener or reader. Furthermore, AVO is committed to accessibility and inclusivity, offering content in various languages to accommodate a global audience. This feature ensures that anyone seeking healing and personal growth can find helpful resources, no matter where they are in the world.

In essence, AVO is more than just a website; it is a sanctuary for those seeking to heal and grow. It provides a safe, supportive environment where individuals can explore powerful content designed to aid in personal development and mental health.

1.2. About us

One business that specializes in software development is AVO Team. We are a team within the company that specializes in developing web applications. In order to satisfy our customers, we constantly search for the best Web-based solution in terms of performance, design, and efficiency. Small and medium-sized businesses, well-known software development partners, and celebrities are among our possible clients.

Company name	AVO Solution
Team name	AVO Team
Business	Design and develop Web applications
Customer	AIoT Lab VN
Contact	Khu phố 6, Đ. Võ Trường Toản, Phường Linh Trung, Thủ Đức, Thành phố Hồ Chí Minh
Email	contact@avo-solution.com
Phone number	(+84) 90 285 0103

1.3. Development process

Our team is using the Agile methodology [1], specifically the [2], to guide our project development.

Why Scrum?

- Flexibility: Allows for quick adaptation to changing requirements.

- Incremental Development: Delivers the product piece by piece, focusing on completing each function required by customers.
- Customer-Centric Approach: Ensures each feature meets customer expectations.

Scrum Process

- Sprints: Iterations of 2-4 weeks delivering shippable product increments.
- Sprint Planning: Identifies priority tasks and sets objectives.
- Daily Stand-Ups: Brief meetings to discuss progress and address blockers.
- Sprint Review: Demonstrates completed functionalities to stakeholders and gathers feedback.
- Sprint Retrospective: Reflects on the sprint to identify improvements.

Continuous Improvement

- Enhanced User Experience: Continuously improving the user interface.
- Value Delivery: Adding and improving features with each iteration.
- Feedback Response: Incorporating stakeholder feedback to refine the product.

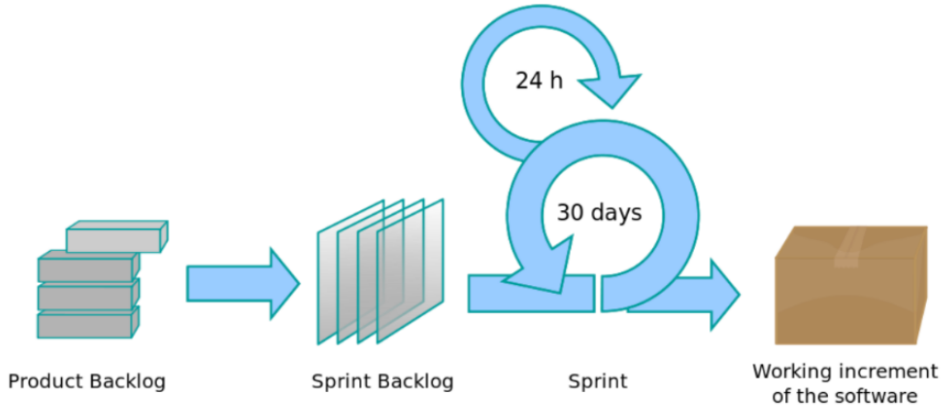

Figure 1 The Scrum Agile methodology used in software development.

1.4. Development environment

Since this is a web-based product, web design and programming languages are used in the project under the MVC model. [3]

The following Programming Languages are used within our system:

- ReactJS [4]: Used to create user interfaces for websites. ReactJS helps build dynamic and optimized interface components, enhancing interactivity and user experience across different devices.
- ExpressJS [5]: As a backend framework running on Node.js, ExpressJS is used to build the server part of the web application. It supports HTTP request handling, routing, and middleware functions required for API development.
- MongoDB [6]: A NoSQL database, used to store website data. MongoDB provides highly scalable and flexible data storage capabilities, suitable for modern web applications that require fast data retrieval speeds and the ability to process large amounts of data.
- Digital Ocean [7]: A cloud hosting platform, used to deploy and host web applications. Digital Ocean provides virtual servers (Droplets) that are scalable and easy to manage, helping to ensure website performance and stability.

To enhance the website's capabilities and efficiency, the project uses the following additional technologies:

- Redux [8]: Is an application state management library, often used with React to manage global state. Redux helps process data consistently and efficiently, especially in complex applications.
- JWT (JSON Web Tokens) [9]: For added security, JWT can be used for user authentication and authorization in your web application.

The project are implemented according to the MVC model:

- Model: the system's data that is kept in an external system and database
- View: the user interface
- Control: the reasoning and algorithms applied to create a dynamic website with the necessary features

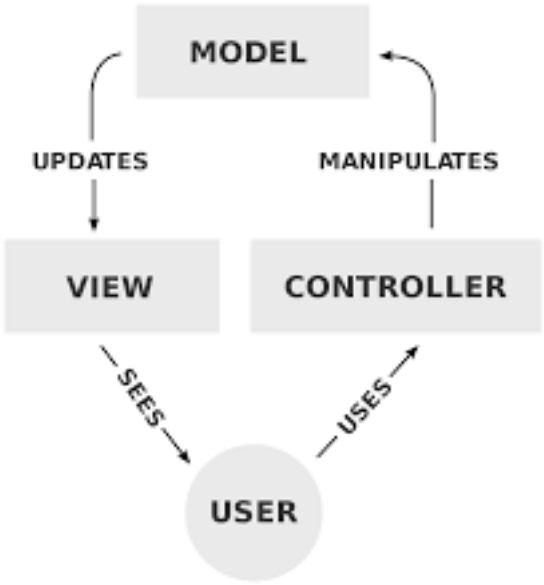

Figure 2 The Model-View-Controller (MVC) architectural pattern used in software development.

CHAPTER 2

REQUIREMENT ANALYSIS AND DESIGN

2.1. REQUIREMENT ANALYSIS

2.1.1. Use case diagram

Actors:

- Vision: A user who visits the website
- Registered User: A user who has an account registered on the website
- Admin: A user with administrative privileges to upload the podcast and book script.

Figure 3 Use case diagram.

Description:

- **Listen to Podcasts:** Users can stream or download podcast episodes, control playback (play, pause, skip), and manage their listening experience.
- **Listen to Audiobooks:** Users can stream or download audiobooks, similar to podcasts, with playback control and session management.

- **Read Audiobooks:** Users can read the text version of audiobooks, with options to adjust the font size, bookmarks, and reading modes (day/night).

- **Browse Content:** Users can explore available content, including podcasts and audiobooks, with filters for categories, genres, and popularity.
- **Search Content:** Users can search for specific podcasts or audiobooks using keywords, titles, authors, or genres.
- **Create Playlists:** Users can create and manage playlists of their favorite podcasts and audiobooks for easy access.
- **Favorite Content:** Users can mark specific podcasts or audiobooks as favorites for quick access and personalized recommendations.
- **Upload Podcasts (Admin):** Admins can upload new podcast episodes, including metadata such as title, description, and category.
- **Upload Audiobooks (Admin):** Admins can upload new audiobooks, managing both audio and text versions, with relevant metadata.
- **Edit Content (Admin):** Admins can modify existing content, update titles, descriptions, and metadata, or replace files.
- **Delete Content (Admin):** Admins can remove outdated or irrelevant content from the platform.
- **Manage Users (Admin):** Admins can manage user accounts, roles, permissions, and account settings.
- **Login:** Required for users and admins to access features like creating playlists, favoriting content, uploading, editing, deleting content, and managing users.

2.1.2. User specification

Use Case 1

Name	**Listen to Podcasts**
Inputs	User selection (podcast episode)
Outputs	1. [If success] Podcast playback starts 2. [If fail] Error Message
Actor	User (Listener) System 1. Open the podcast page -> Display the podcast list 2. Select a podcast episode -> Stream the podcast • [if successful] Start playback • [if fail] Display an error message
Precondition	The user is logged in and has a stable internet connection

Postcondition	None
User story	**As** a podcast enthusiast, I want to listen to my favorite podcasts hands-free while multitasking or relaxing

Figure 4 User specification for Listen to Podcasts

Use Case 2

Name	**Listen to Audiobooks**
Inputs	User selection (audiobook)
Outputs	1. [If success] Audiobook playback starts 2. [If fail] Error message
Actor	User (Listener) System 1. Open the audiobook page -> Display the audiobook list 2. Select an audiobook -> Stream the audiobook - [if successful] start playback - [if fail] display an error message
Precondition	User is logged in and has a stable internet connection
Postcondition	None
User story	As a book lover, I want to listen to audiobooks hands-free while multitasking or relaxing

Figure 5 User specification for Listen to Audiobooks

Use Case 3

Name	**Read Audiobooks**
Inputs	User selection (audiobook)
Outputs	1. [If success] Audiobook text is displayed 2. [If fail] Error message
Actor	User (Reader) System 1. Open the audiobook page -> Display the audiobook list 2. Select an audiobook to read -> Display the audiobook text • [if successful] show the text • [if fail] display an error message
Precondition	User is logged in and has a stable internet connection
Postcondition	None
User story	As a book lover, I want to read audiobooks on my device with options to adjust the reading interface.

Figure 6 User specification for Read Audiobooks

Use Case 4

Name	**Browse Content**
Inputs	User action (browsing categories/genres)
Outputs	1. [If success] Content list is displayed 2. [If fail] Error message

Actor	User (Listener/Reader) System 1. Open the content page -> Display categories/genres 2. Select a category/genre -> Display the content list • [if successful] show the list • [if fail] display an error message
Precondition	User is logged in and has a stable internet connection
Postcondition	None
User story	As a user, I want to browse available podcasts and audiobooks by categories and genres.

Figure 7 User specification for Browse Content

Use Case 5

Name	**Search Content**
Inputs	User input (search query)
Outputs	1. [If success] Search results are displayed 2. [If fail] Error message
Actor	User (Listener/Reader) System 1. Enter a search query -> Process the query 2. Display the search results • [if successful] show the results • [if fail] display an error message
Precondition	User is logged in and has a stable internet connection

Postcondition	None
User story	As a user, I want to search for specific podcasts or audiobooks using keywords.

Figure 8 User specification for Name Search Content

Use Case 6

Name	**Create Playlists**
Inputs	User input (playlist details)
Outputs	1. [If success] Playlist is created 2. [If fail] Error message
Actor	User (Listener/Reader) System 1. Open the playlist creation page -> Display the form 2. Enter playlist details -> Save the playlist • [if successful] confirm creation • [if fail] display an error message
Precondition	User is logged in and has a stable internet connection
Postcondition	None
User story	As a user, I want to create playlists of my favorite podcasts and audiobooks for easy access.

Figure 9 User specification for Create Playlists

Use Case 7

Name	**Favorite Content**
Inputs	User action (favoriting content)
Outputs	1. [If success] Content is marked as favorite 2. [If fail] Error message
Actor	User (Listener/Reader) System 1. Select content to favorite -> Mark as favorite • [if successful] confirm favoriting • [if fail] display an error message
Precondition	User is logged in and has a stable internet connection
Postcondition	None
User story	As a user, I want to mark content as a favorite for quick access later.

Figure 10 User specification for Favorite Content

Use Case 8

Name	**Upload Podcasts (Admin)**
Inputs	Admin input (podcast details, file)
Outputs	1. [If success] Podcast is uploaded 2. [If fail] Error message

Actor	Admin System 1. Open the podcast upload page -> Display the form 2. Enter podcast details and upload file -> Save the podcast • [if successful] confirm upload • [if fail] display an error message
Precondition	Admin is logged in and has a stable internet connection
Postcondition	None
User story	As an admin, I want to upload new podcast episodes to the platform.

Figure 11 User specfication for Upload Podcasts (Admin)

Use Case 9

Name	**Upload Audiobooks (Admin)**
Inputs	Admin input (audiobook details, file)
Outputs	1. [If success] Audiobook is uploaded 2. [If fail] Error message
Actor	Admin System 1. Open the audiobook upload page -> Display the form 2. Enter audiobook details and upload file -> Save the audiobook • [if successful] confirm upload • [if fail] display an error message
Precondition	Admin is logged in and has a stable internet

Postcondition	None
User story	As an admin, I want to upload new audiobooks to the platform.

Figure 12 User specification for Upload Audiobooks (Admin)

Use Case 10

Name	**Edit Content (Admin)**
Inputs	Admin input (content details)
Outputs	1. [If success] Content is edited 2. [If fail] Error message
Actor	Admin System 1. Search for content to edit -> Display the content 2. Edit content details -> Save changes • [if successful] confirm changes • [if fail] display an error message
Precondition	Admin is logged in and has a stable internet connection
Postcondition	None
User story	As an admin, I want to edit existing content on the platform.

Figure 13 User specification for Edit Content (Admin)

Use Case 11

Name	**Delete Content (Admin)**
Inputs	Admin action (delete content)
Outputs	1. [If success] Content is deleted 2. [If fail] Error message
Actor	Admin System 1. Search for content to delete -> Display the content 2. Delete the content • [if successful] confirm deletion • [if fail] display an error message
Precondition	Admin is logged in and has a stable internet connection
Postcondition	None
User story	As an admin, I want to delete outdated or irrelevant content from the platform.

Figure 14 User specification for Delete Content (Admin)

Use Case 12

Name	**Manage Users (Admin)**
Inputs	Admin action (manage user accounts)
Outputs	1. [If success] User accounts are managed 2. [If fail] Error message

Actor	Admin System 1. Search for user accounts to manage -> Display the user list 2. Edit user permissions/details -> Save changes • [if successful] confirm changes • [if fail] display an error message
Precondition	Admin is logged in and has a stable internet
Postcondition	None
User story	As an admin, I want to manage user accounts, including roles and permissions.

Figure 15 User specification for Manage Users (Admin)

Use Case 13

Name	**Login the system**
Inputs	User name Password
Outputs	1. [If success] The home page with the user's authorization 2. [If fail] The login page
Actor	User (Listener/ Reader) System 1. Open the login page -> Display the login page 2. Enter the username and password 3. Submit -> Check the user's info. • [if successful] return the home page • [if fail] return the login page
Precondition	The user has a registered account of the online store that was created earlier (user name, password)

Postcondition	None
User story	As a podcast enthusiast and book lover, I want a website where can enjoy my world listening to my favorite podcasts and books that are allowed to me. So that I can engage audio content and literature hands-free while multitasking or relaxing.

Figure 16 User specification for Login the system

2.2. Functional requirements

User Registration and Authentication: Users should be able to create accounts using their email address and a password. They should also have the ability to log in and log out securely, ensuring their personal information is protected. Additionally, admins should have a separate login with elevated permissions to manage the platform effectively.

User Profile Management: Users should be able to update their profile information as needed. Additionally, they should have the ability to reset their passwords to maintain account security and access.

Content Browsing and Searching: Users should be able to browse the available podcasts and audiobooks easily, navigating through various categories and genres to discover new content. Additionally, they should have the capability to search for specific content using keywords, enabling them to quickly find the podcasts or audiobooks they are interested in. The search functionality should be intuitive and efficient, providing relevant results based on the user's input.

Audio Playback: Users should be able to stream both podcasts and audiobooks seamlessly, ensuring a smooth listening experience. They should also have full control over playback, including the ability to play, pause, and skip content as desired. This functionality will allow users to manage their listening sessions conveniently and tailor their experience to their preferences.

Reading Audiobooks: Users should be able to access and read the text version of audiobooks directly on the platform. They should have various options to enhance their reading experience, including the ability to adjust the font size for better readability, use bookmarks to easily return to specific sections, and switch between day and night reading modes to suit their lighting conditions and personal preferences.

Playlist Management: Users should be able to create, edit, and delete playlists. And they should be able to add and remove content from playlists.

Favoriting Content: Users should be able to mark content as a favorite for easy access later.

Admin Content Management: Admins should be able to upload, edit, and delete both podcast and audiobook content on the platform. They should also have the capability to manage metadata for each piece of content, ensuring that information such as titles, descriptions, authors, and categories are accurate and up-to-date. This functionality allows admins to maintain a well-organized and easily navigable content library for users.

User Management: Admins should have the capability to view and manage user accounts comprehensively. Additionally, they should be able to assign roles and permissions to users, ensuring that each user has the appropriate level of access and functionality based on their role within the platform.

Feedback and Rating System: Users should be able to rate and leave feedback on content and admins should be able to view and moderate feedback.

2.3. Non-Functional Requirements

2.3.1. Performance Requirements

Response Time: The web application's pages should be loaded and it should be able to provide responses to user interaction in less than 2 seconds depending on conditions.

Efficiency: Audio and video streaming should load in 'You wait 3 seconds' time after the user clicks and accessibility should be allowed up to 10,000 users on the same file.

2.3.2. Humanity Requirements

Most system users have little to no experience with computers. The development team must produce comprehensive documentation so that, after reading it, a user can quickly grasp the system. The system's graphical user interfaces also need to be well-designed to make it simple to use and easy to learn. It shouldn't take the user a long time to become accustomed to the system interfaces. In order to improve user experience, the interface must also be responsive and aesthetically pleasing.

2.3.3. Security Requirements

Data Encryption: Employ robust methods of data encrypting of both storage facilities and the transfers as a means of safeguarding the data of users.

Authentication and Authorization: Utilize secure authentication protocols like OAuth or JWT to safeguard the application and establish the necessary permissions if users are to access any specific parts of the application.

Security Audits: Some of the measures include security audits that would be conducted from time to time and updating the security to reduce or avoid these issues.

2.3.4. Legal and Regulatory Requirements

Data Protection: Make clear privacy policies like those of GDPR for European members or CCPA for those in California are being followed.

Copyright Laws: Comply with international or regional copyright laws on the use of content on the site, to make sure that all content is either produced by the company or used with permission from the copyright owner.

Audit Requirements: Make provision to record all critical activities and alterations whereby audits can be done in order to monitor and /or ascertain compliance to regulatory standards.

2.3.5. Stakeholder Requirements

Users: Access to a wide range of podcasts and audiobooks. A user-friendly interface for browsing and listening to content.

Admins: Efficient tools for content and user management. Detailed analytics and reporting on user interactions and content performance.

Business Owners:

- A scalable and maintainable system to support business growth.
- Insights into user behavior and preferences to drive business decisions.
- By addressing these requirements, the website will provide a comprehensive and user-friendly platform for both podcast enthusiasts and book lovers to enjoy their favorite audio content.

2.3.6. Support Requirements

The people who use systems the least are those with little experience maintaining technology, particularly store managers. Thus, in addition to creating the system, the development team should have a better method so that the user can get assistance from the team as soon as possible.

Provide a hotline for customers to contact the store manager with feedback in the event of an issue. Offer a remote maintenance tool (such as TeamViewer, Computer Remote Control, etc.) to assist the manager directly.

Regularly verify and preserve the system's stability (every three or two months).

2.3.7. Interface Requirements

User Interface (UI) Design

- Consistency: It is important to have a unified interface and design throughout the website and considered the use of mobile devices. This includes such features as color, fonts, icons, and layouts, and their structures that are provided on the website.
- Responsiveness: Additionally, the GUI should adapt to the sizes and orientation of the devices through which they are accessed, whether computers/laptops, tablets, or smart mobile phones.
- Navigation: It should have a navigation hierarchy that is easy to implement and understand via menus, breadcrumbs and search bar to enable the users to access the content of the platform with ease.

Content Presentation

- Readable Text: The text in each article is comprehensible and images have proper font size, line spacing and contrast. There should be provisions for changing the size of the text on the screen since this is important for users with impaired vision.
- Multimedia Controls: The controls for the playback, pause, rewind, forward, and changing volume of the configured audio and video streams should also be simple to use.
- Reading Modes: Save different versions of reading mode (for example, day theme and night theme) to make users feel comfortable during the process of reading.

Customization

- User Profiles: It must be possible for users to configure their profile and this must include settings such as what kind of content to recommend to the users, how the content should play and what notifications should be sent to the users.
- Playlist Management: Playlists should be manageable and be easy to create, modify, or even delete, and the Playlist Items belonging to the Playlist should be manageable as well in terms of adding or removing a Playlist Item from a Playlist.

Administrative Interface

- Content Management: The interface provided to the admins should allow ease of image upload, creation, editing, and deletion and should support features such as multiple upload and metadata operations.
- User Management: It should be easy for admins to be able to dictate the roles that any end user has, the kind of permissions that they enjoy amongst other things that we shall see as we proceed with the text.
- Analytics and Reporting: Give facility to admins that they can monitor their usage statistics, user’s activities or other info etc.

2.4. Design

2.4.1. Interface Requirements

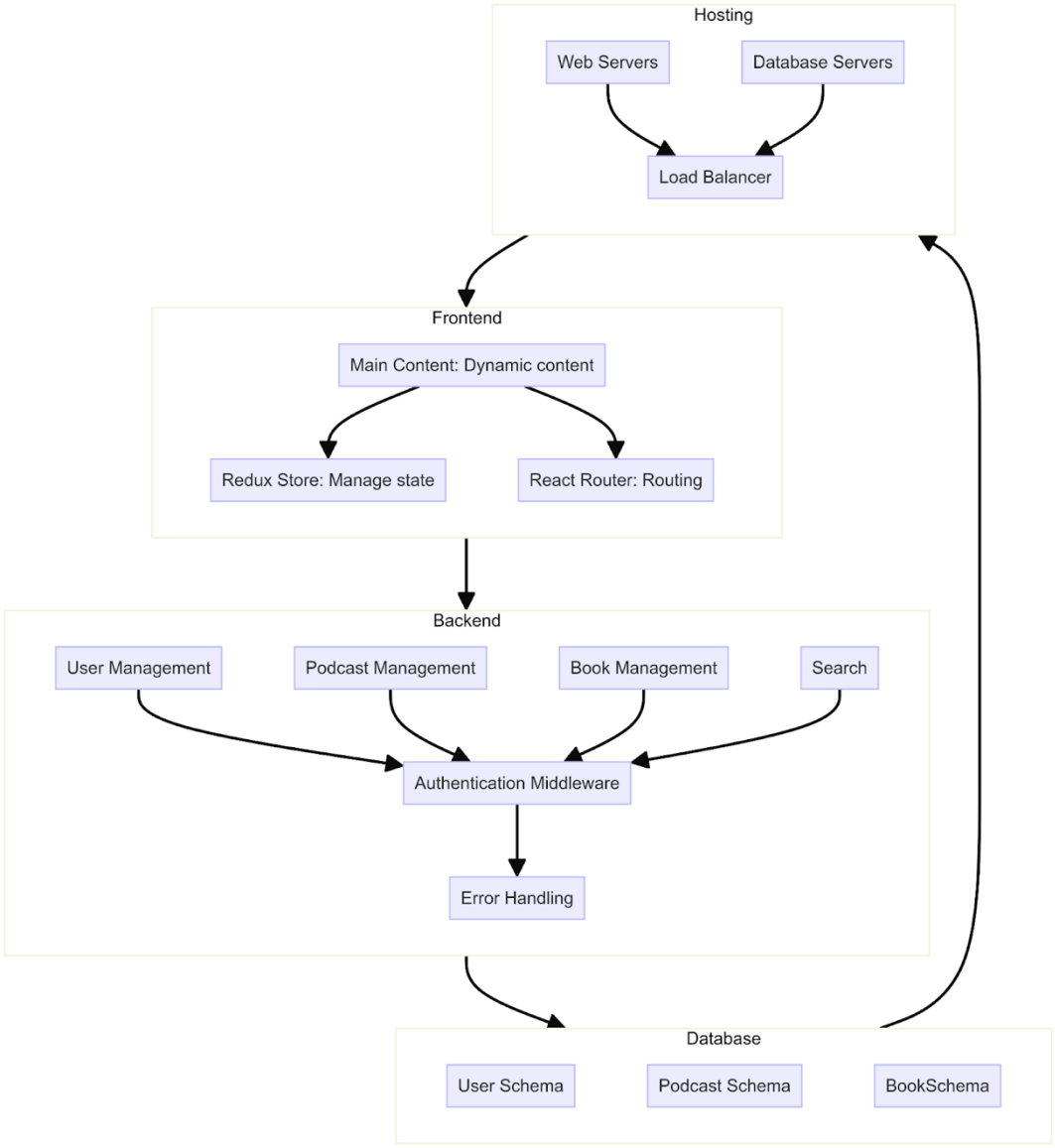

Figure 17 System architechture model

The architecture of the AVO platform includes such components and aims at scalability and security of the data as well as effective operation of the system. The architecture is divided into several layers and components, each with specific responsibilities:The architecture is divided into several layers and components, each with specific responsibilities:

Hosting

- Web Servers and Database Servers: The platform used is Digital Ocean which provides virtual servers known as Droplets and the web application is hosted on a different droplet as the database. A load balancer is used for clients to be split in fairly between servers and help in ensuring that they are quality servers.

Frontend

- Main Content: The frontend of this web application is created with REACTJS, and it means that the layout and content are rendered dynamically to provide the best user experience across multiple display sizes.

- Redux Store: Controls state of the application at the global level, thus stabilizing its work and orchestrating data flow between parts of the program.
- React Router: Helps users to switch easily from one page or one section of the application to another without having to carry out a difficult process of searching for the right page.

Backend

- User Management, Podcast Management, Book Management, Search: This is mainly built using ExpressJS where operations like user management, podcast and book content, and Search functionalities are encapsulated.
- Authentication Middleware: The user login and approval process is implemented using JSON Web Token (JWT), thus limiting user access to specified areas of the application.
- Error Handling: To ensure the application is protected from a multiple point failure issue a centralized error management is done so that errors can be captured and handled effectively.

Database

- Schemas: A MongoDB is used for storing the data and it has the appropriate schemes for users, podcasts, and the books section. This is a NoSQL database that offers more openness and simplicity of data storage and retrieval for any sized data set.

2.4.2. Class diagram

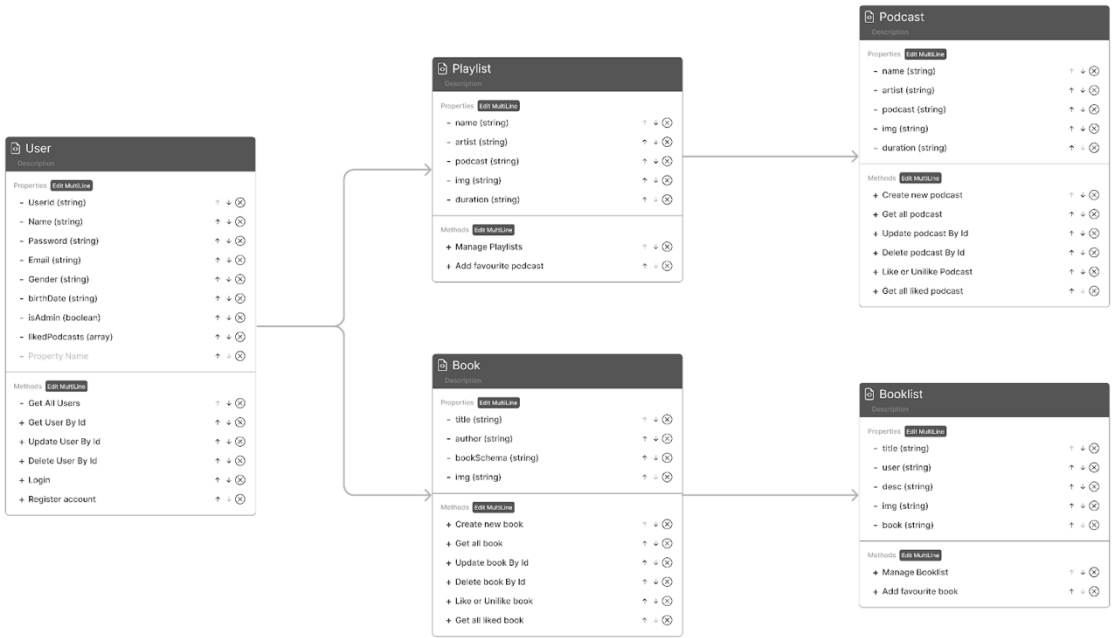

Figure 18 Class Diagram

2.4.3. Use case diagram

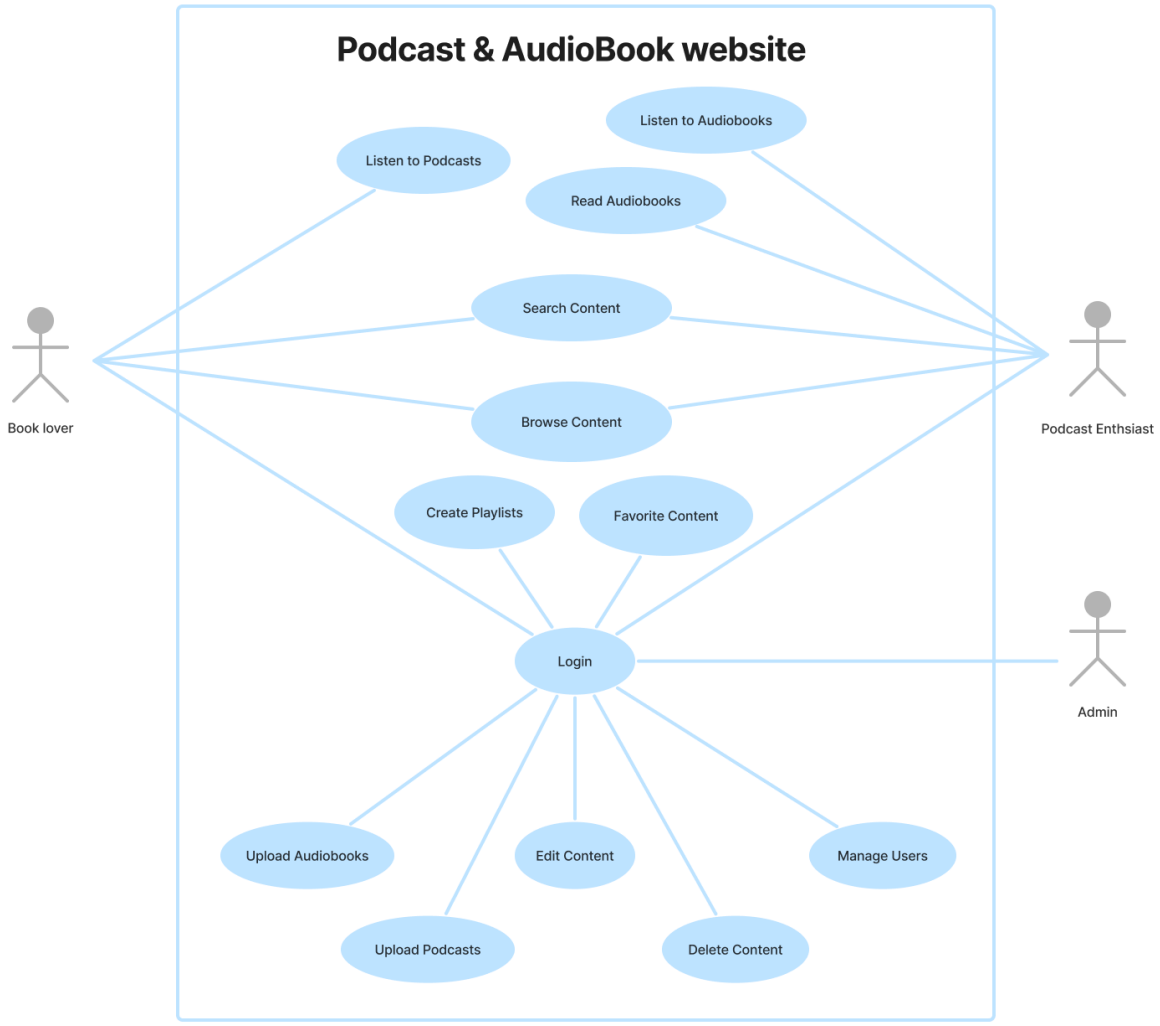

Figure 19 Use case diagram

2.4.4. Sequence diagram

Use case 1: Listening to Podcasts
Description:

- User opens the Audio Content Platform.
- User searches or browses for a podcast.
- Audio Content Platform retrieves the list of podcasts.
- User selects a podcast episode to play.
- Audio Content Platform streams the podcast to the user.

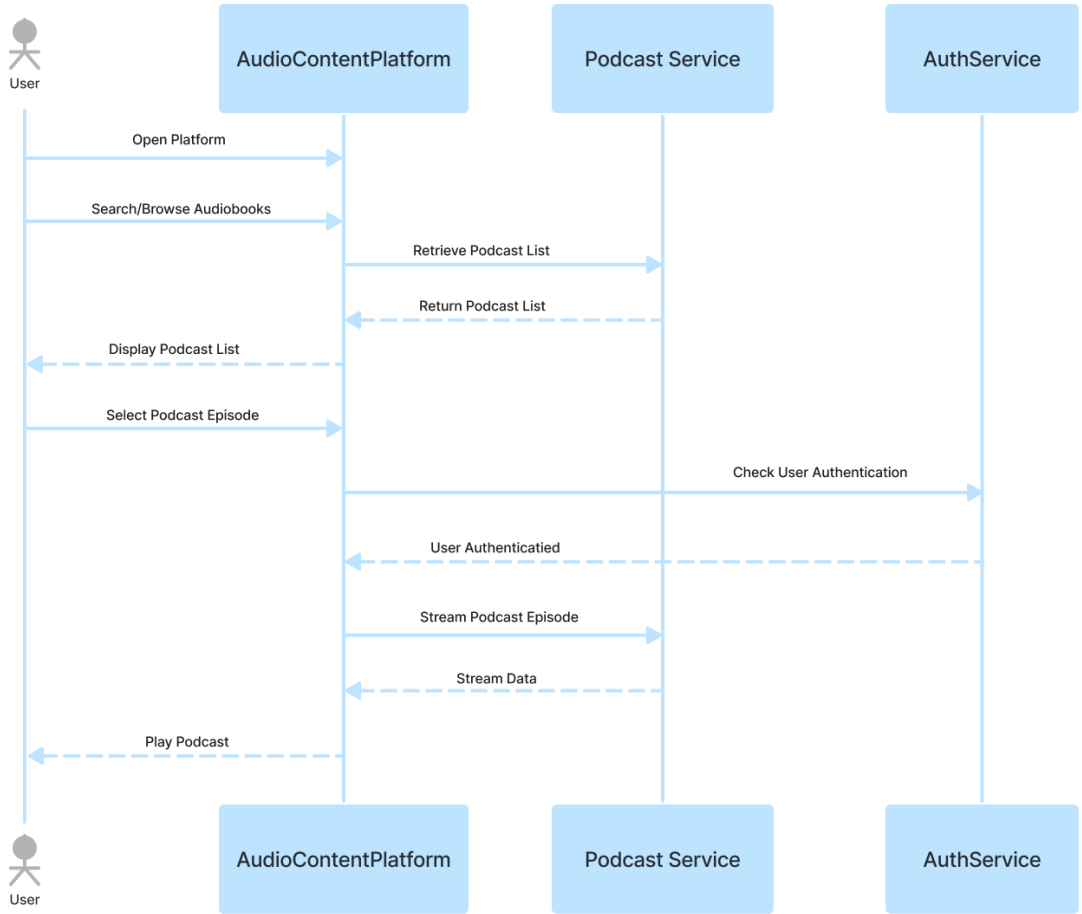

Figure 20 Sequence diagram for Listening to Podcasts

Use case 2: Listening to Audiobooks

Description:

- User opens the Audio Content Platform.
- User searches or browses for an audiobook.
- Audio Content Platform retrieves the list of audiobooks.
- User selects an audiobook to play.
- Audio Content Platform streams the audiobook to the user.

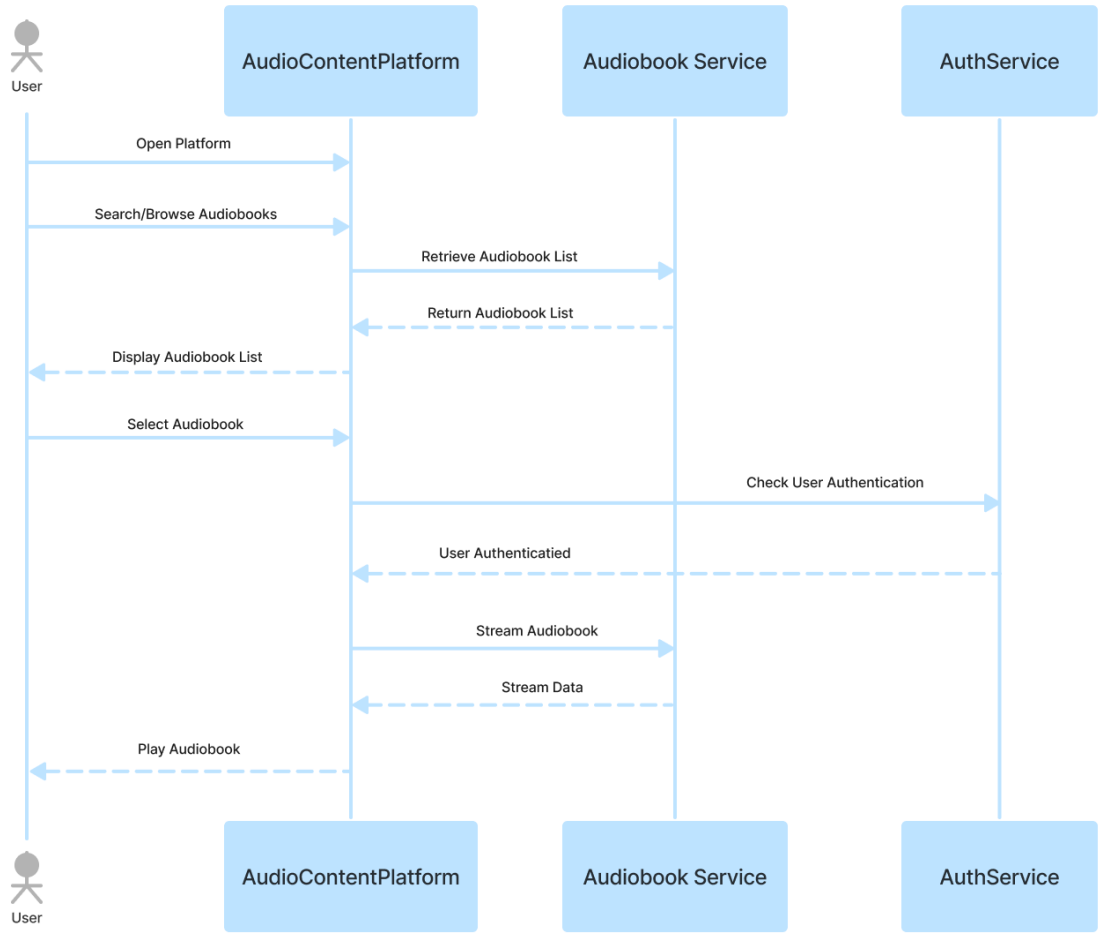

Figure 21 Sequence diagram for Listening to Audiobooks

Use case 3: Reading Audiobooks

Description:

- User opens the Audio Content Platform.
- User searches or browses for an audiobook.
- Audio Content Platform retrieves the list of audiobooks.
- User selects an audiobook to read.
- Audio Content Platform displays the text version of the audiobook.

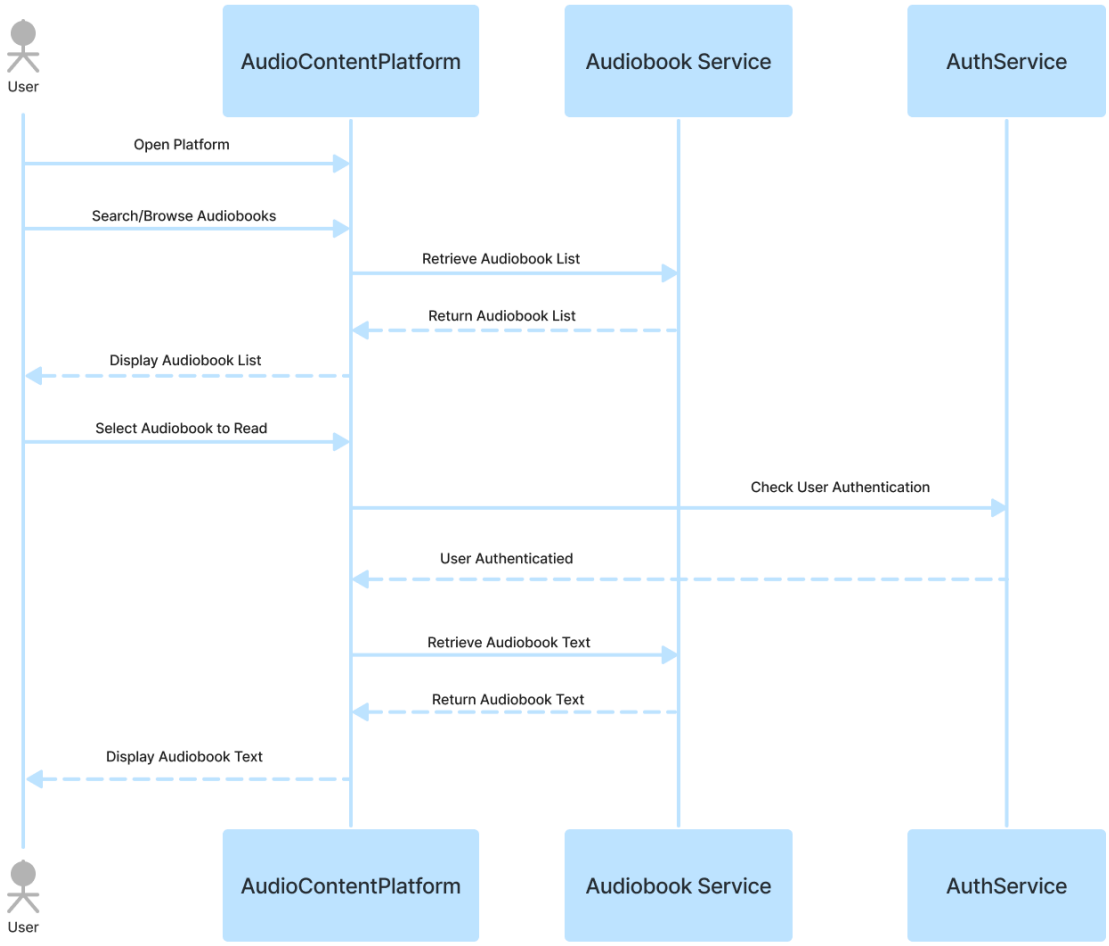

Figure 22 Sequence diagram for Reading Audiobooks

Use case 4: Browsing Content

Description:

- User opens the Audio Content Platform.
- User browses through the categories or genres.
- Audio Content Platform retrieves and displays content based on the selected category or genre.

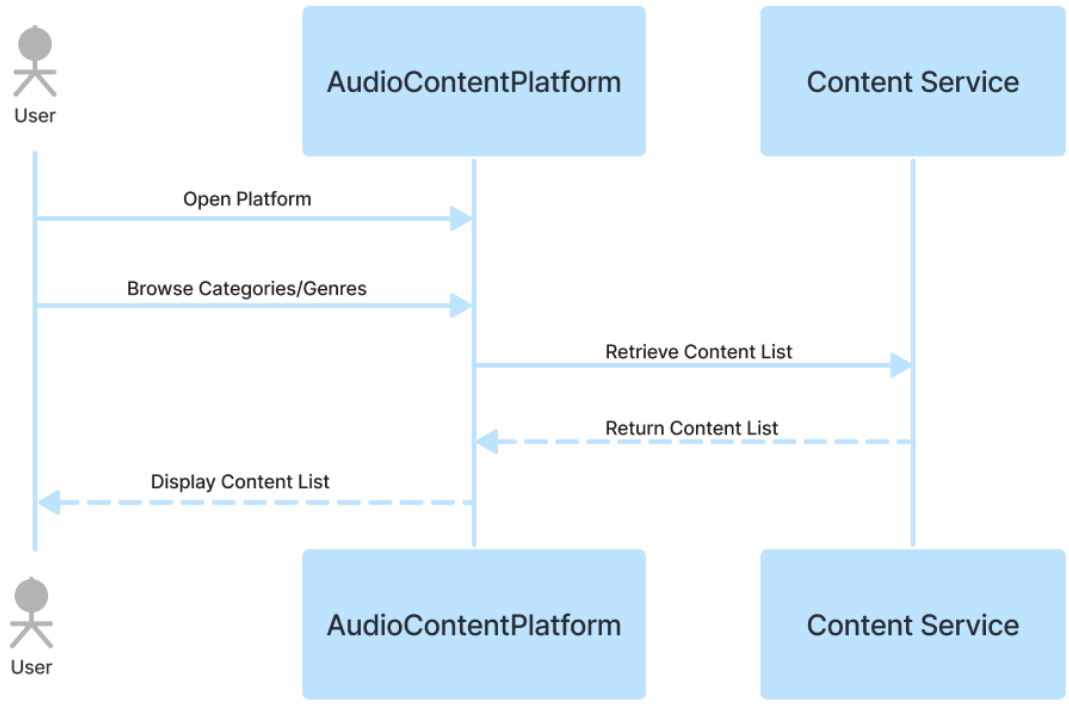

Figure 23 Sequence diagram for Browsing Content

Use case 5: Searching Content

Description:

- User enters a search query in the Audio Content Platform.
- Audio Content Platform processes the search query.
- Audio Content Platform retrieves and displays search results.

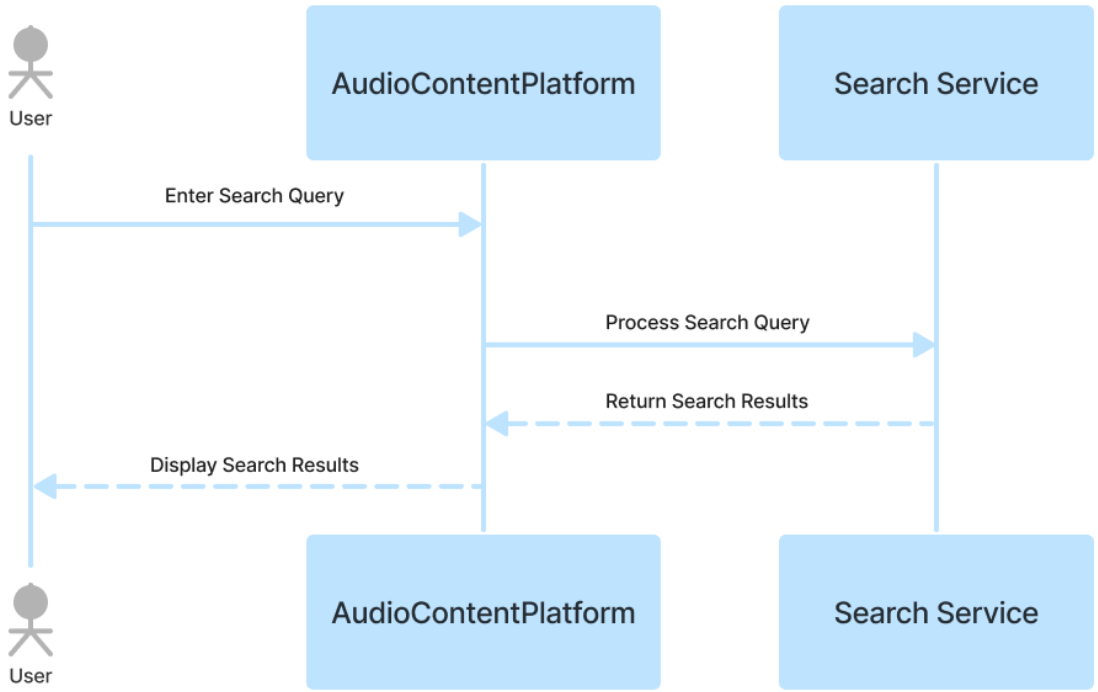

Figure 24 Sequence diagram for Searching Content

Use case 6: Creating Playlists
Description:

- User logs into the Audio Content Platform.
- User searches or browses for content to add to the playlist.
- User creates a new playlist or selects an existing one.
- User adds content to the playlist.
- Audio Content Platform saves the playlist.

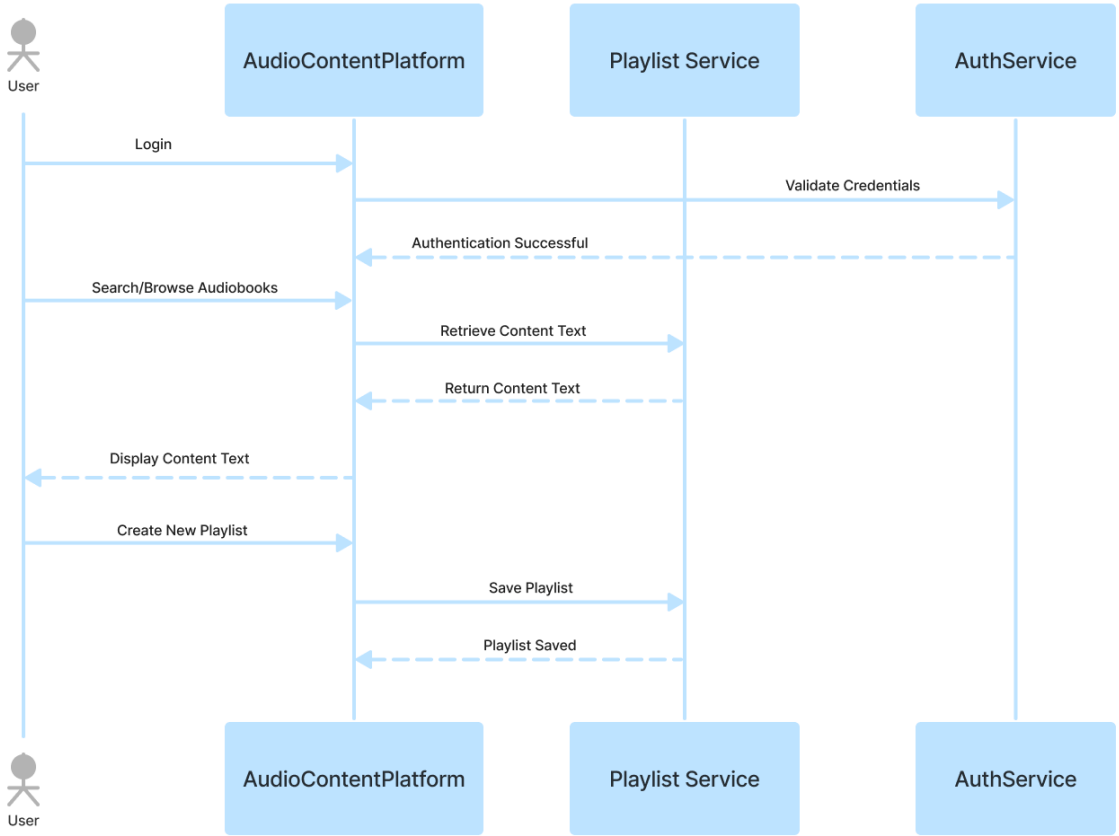

Figure 25 Sequence diagram for Creating Playlists

Use case 7: Favoriting Content
Description:

- User logs into the Audio Content Platform.
- User searches or browses for content.
- User marks content as favorite.
- Audio Content Platform saves the favorite content.

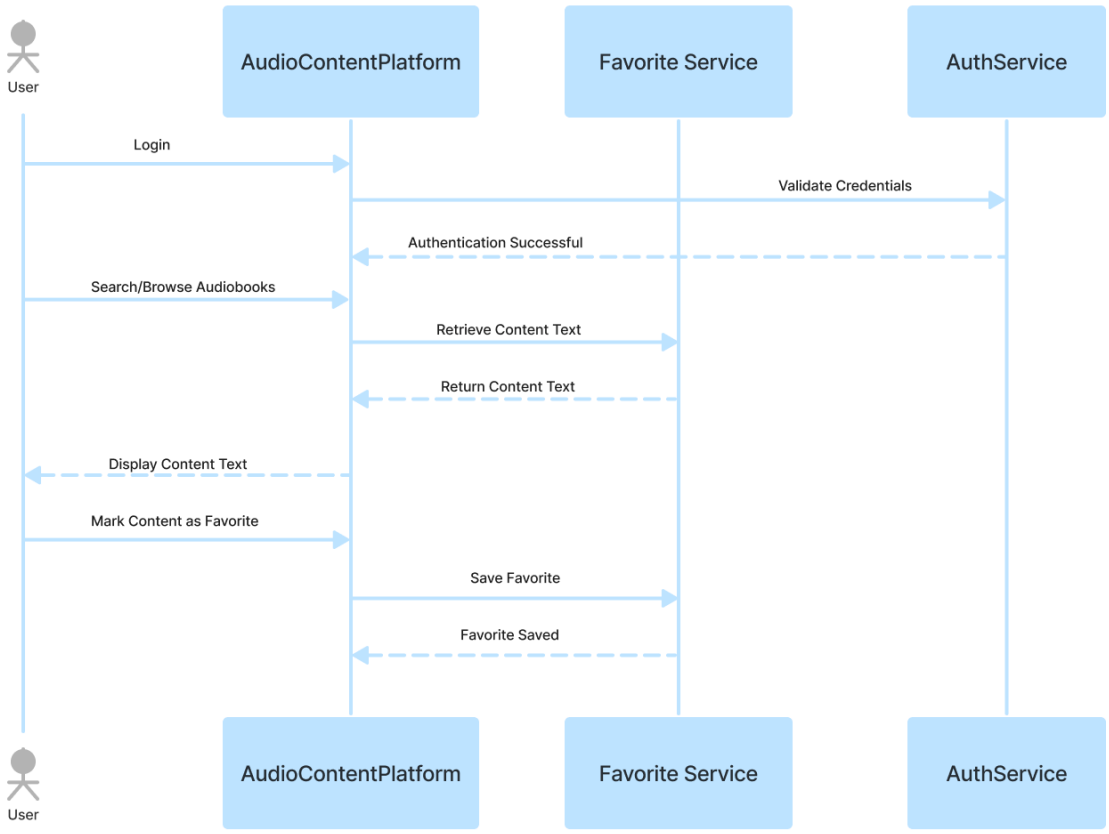

Figure 26 Sequence diagram for Favoriting Content

Use case 8: Uploading Podcasts (Admin)

Description:

- Admin logs into the Audio Content Platform.
- Admin selects the upload option for podcasts.
- Admin uploads podcast file and enters metadata.
- Audio Content Platform saves the podcast and metadata.

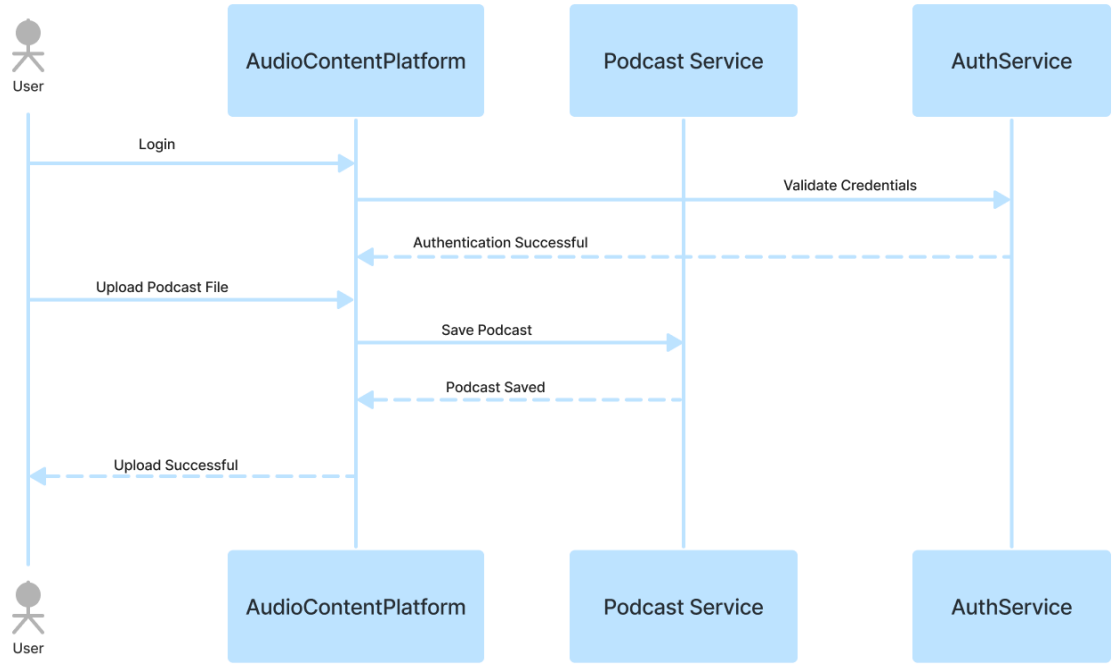

Figure 27 Sequence diagram for Uploading Podcasts (Admin)

Use case 9: Uploading Audiobooks (Admin)

Description:

- Admin logs into the Audio Content Platform.
- Admin selects the upload option for audiobooks.
- Admin uploads audiobook file and enters metadata.
- Audio Content Platform saves the audiobook and metadata.

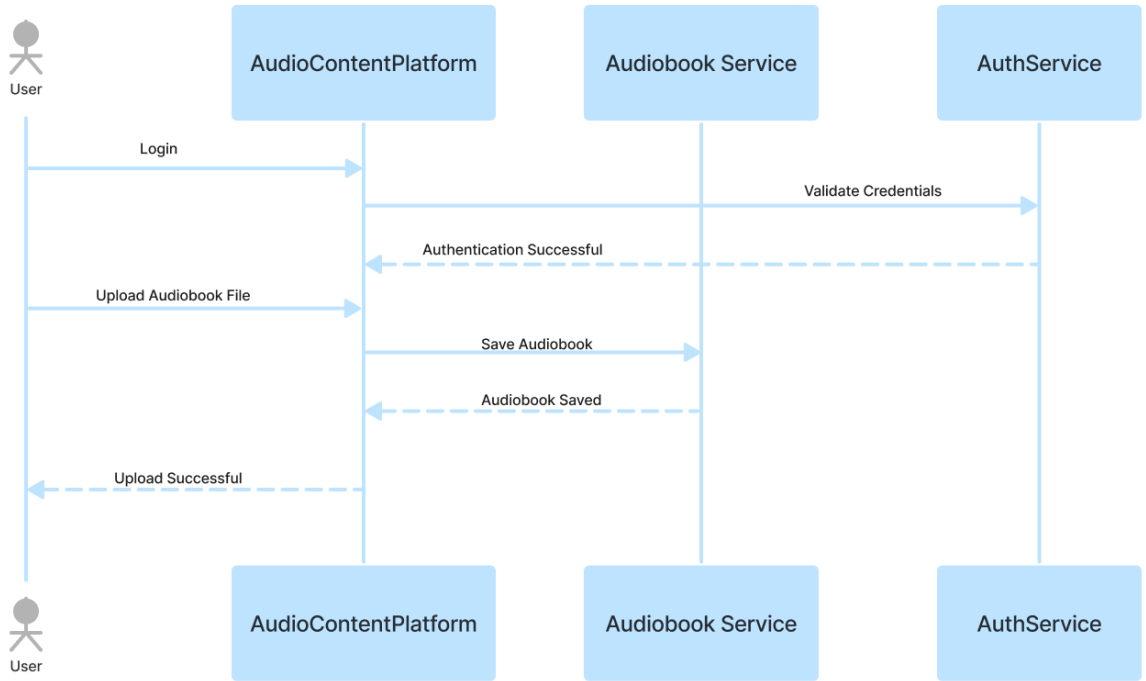

Figure 28 Sequence diagram for Uploading Audiobooks (Admin)

Use case 10: Editing Content (Admin)
Description:

- Admin logs into the Audio Content Platform.
- Admin searches for content to edit.
- Admin makes necessary edits to the content.
- Audio Content Platform updates the content.

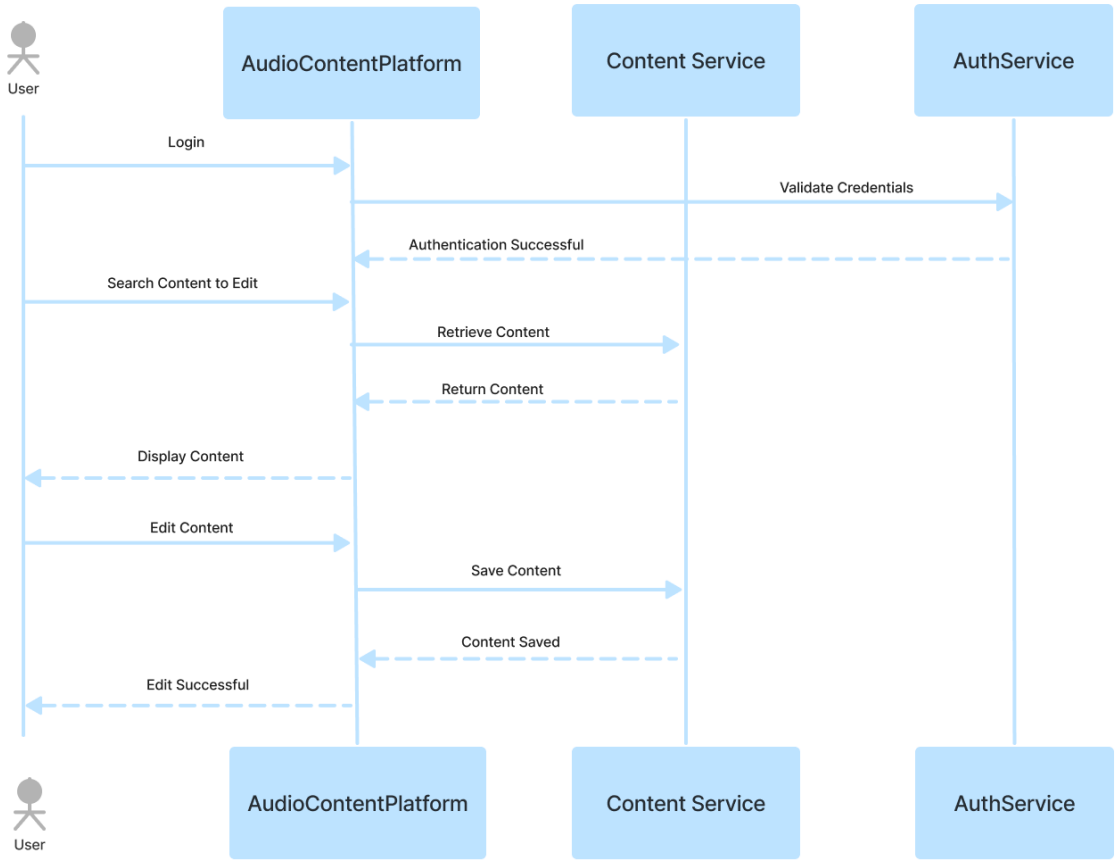

Figure 29 Sequence diagram for Editing Content (Admin)

Use case 11: Deleting Content (Admin)
Description:

- Admin logs into the Audio Content Platform.
- Admin searches for content to delete.
- Admin deletes the selected content.
- Audio Content Platform removes the content.

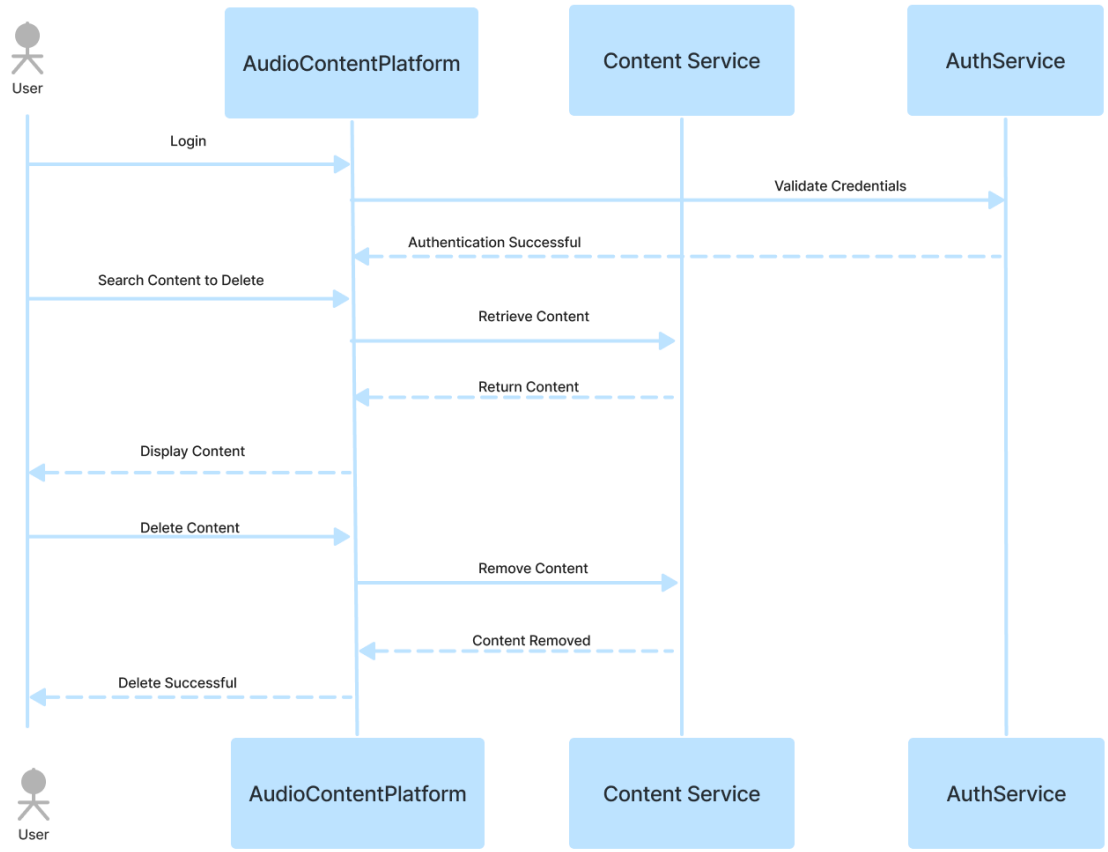

Figure 30 Sequence diagram for Deleting Content (Admin)

Use case 12: Managing Users (Admin)

Description:

- Admin logs into the Audio Content Platform.
- Admin searches for user accounts to manage.
- Admin updates user account details or permissions.
- Audio Content Platform saves the updates.

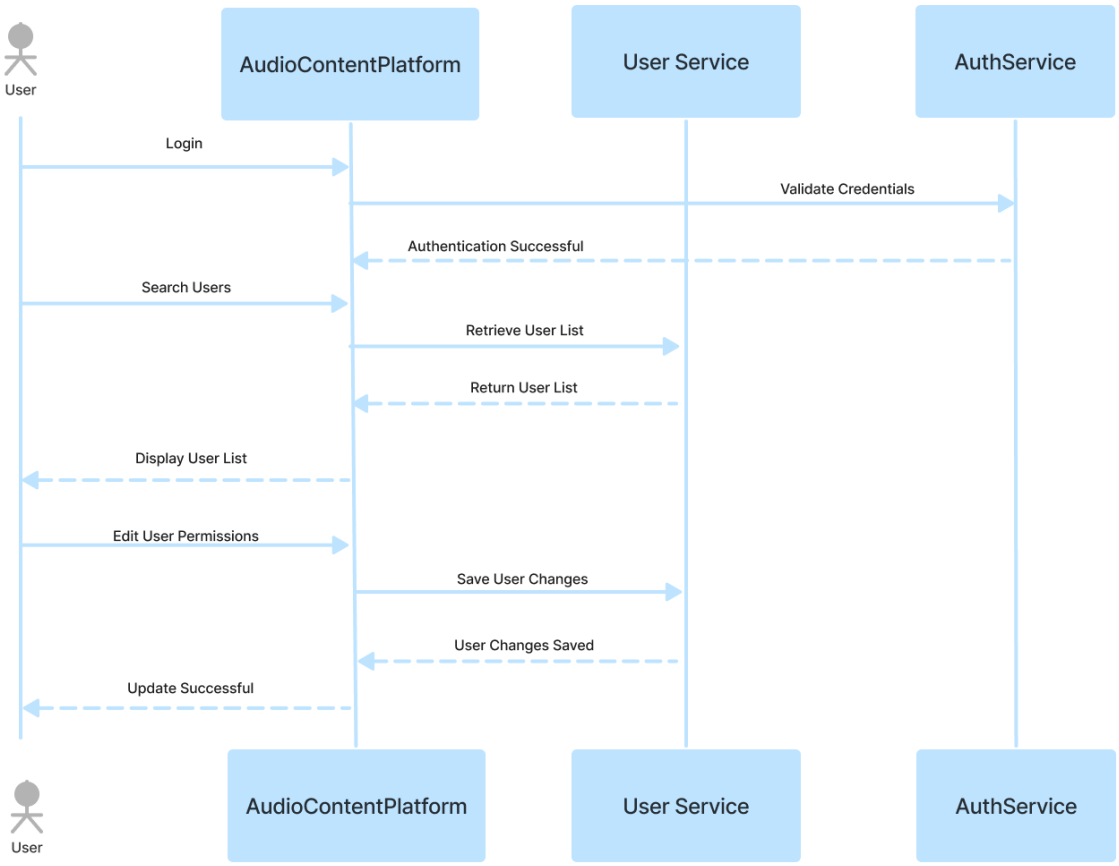

Figure 31 Sequence diagram for Managing Users (Admin)

Use case 12: Login

Description:

- User or Admin opens the Audio Content Platform.
- User or Admin enters login credentials.
- Audio Content Platform validates the credentials.
- User or Admin is granted access to the platform.

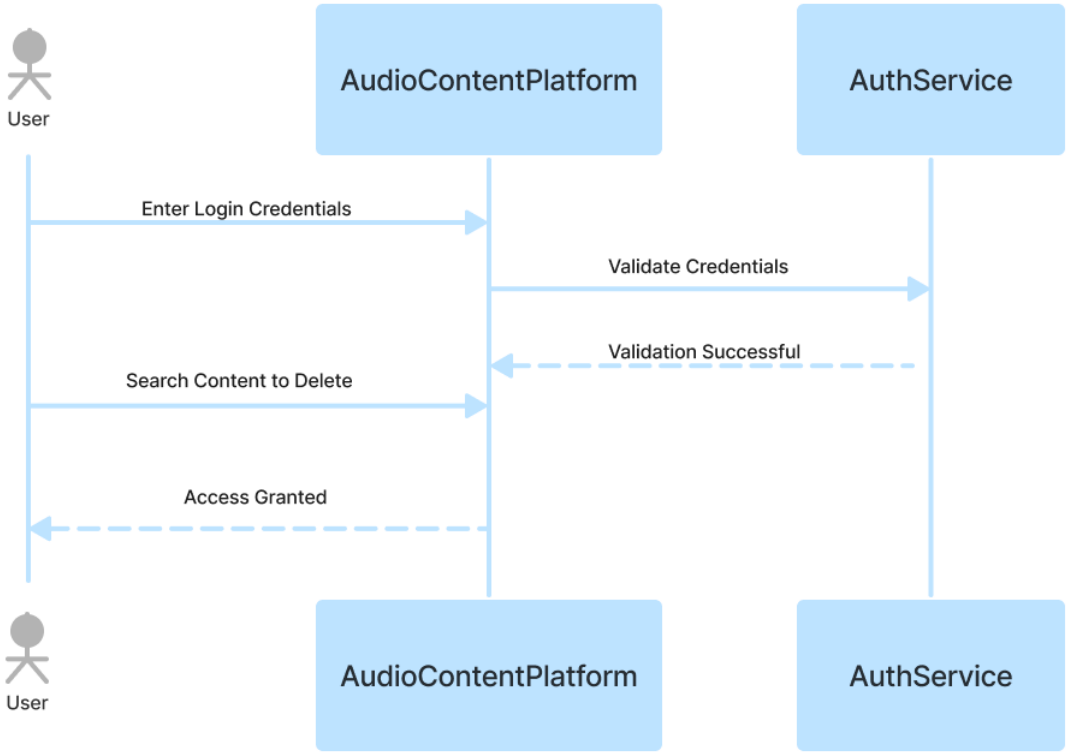

Figure 32 Sequence diagram for login

CHAPTER 3

IMPLEMENT

3.1. Discovery page

This is a minimalist start screen - Discovery Page - for those who are new to the website - asking about the user's needs for 2 options: listening or reading.

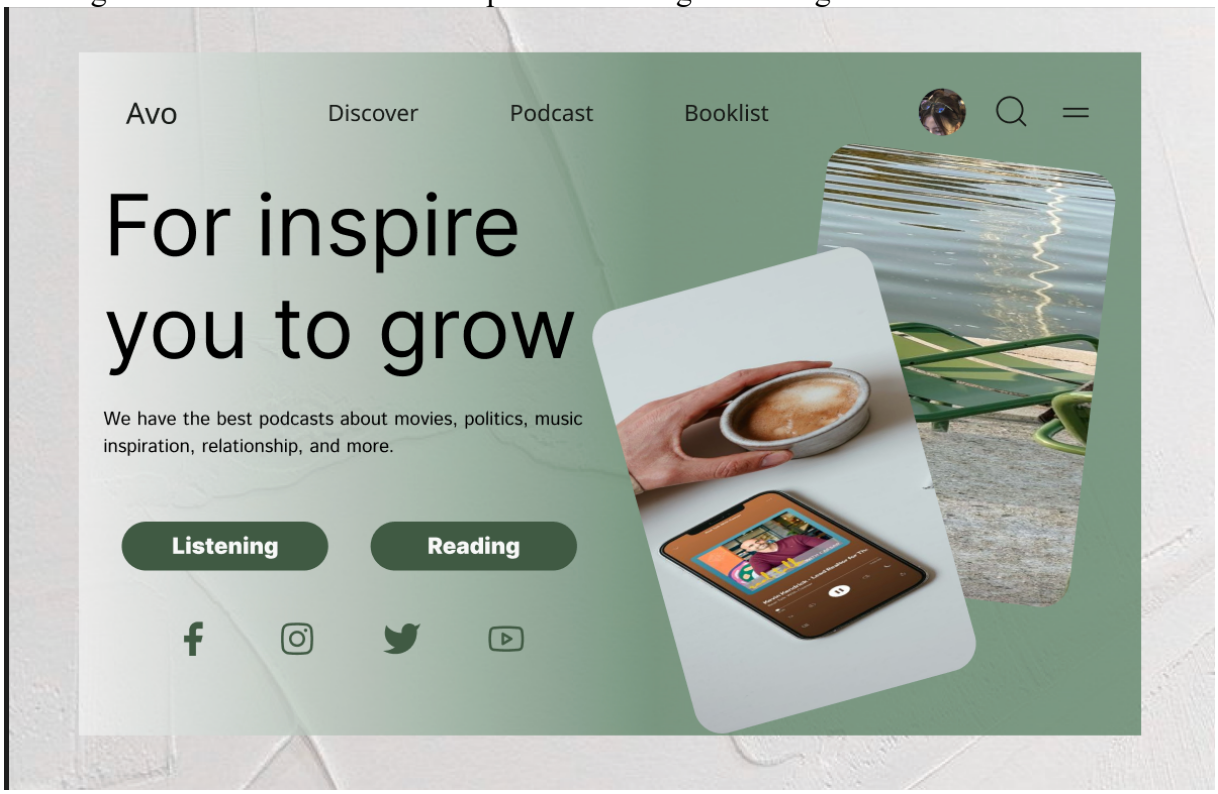

Figure 33 Discovery page.

3.2. Login page

This is the login screen. In addition to visiting this website, to be able to unlock features and save according to personal preferences, we created a login to do that, in addition, for convenience, users can also sign in with google easily.

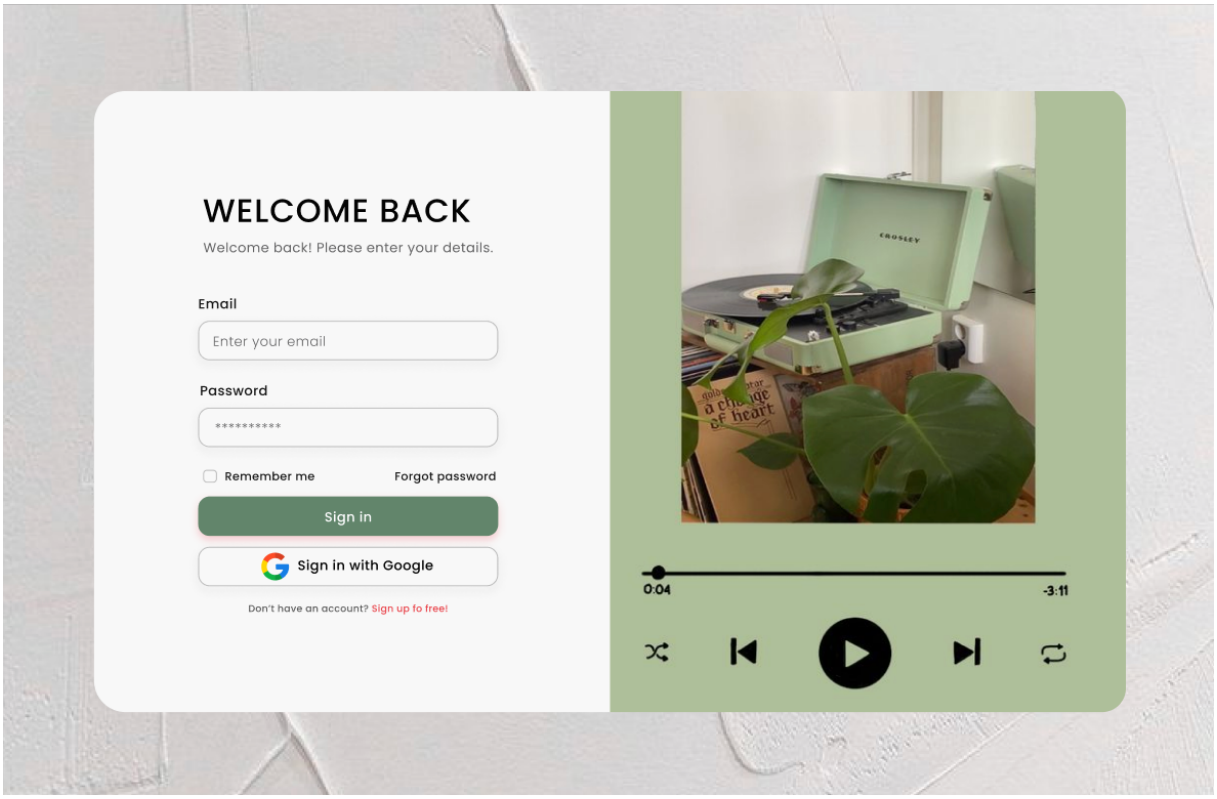

Figure 34 Login page.

3.3. Register page

If you don't have an account, sign up now, no need for too much complicated information.

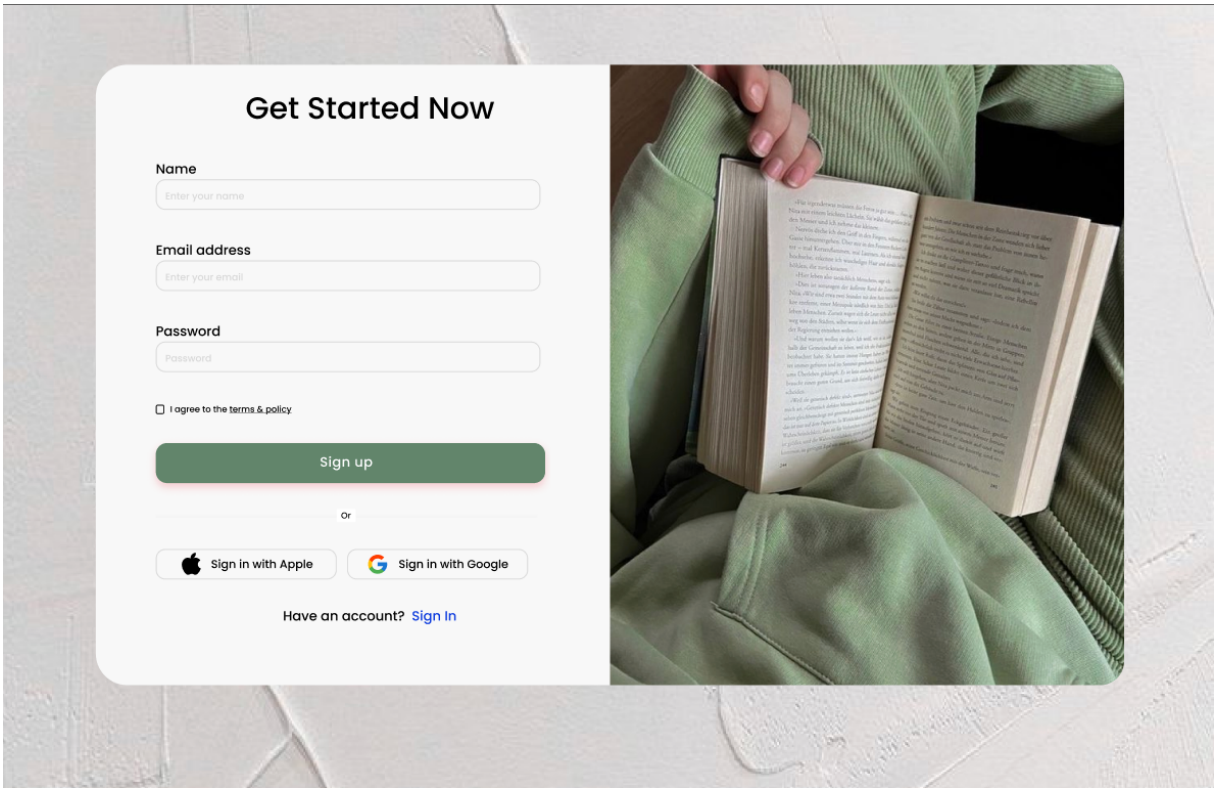

Figure 35 Register page.

3.4. Listening podcast page

If you want to listen. We will take you to the podcast page, We give you both light or dark mode to suit everyone's preferences. The left sidebar will help you control switching between pages, you may want to switch to reading without having to go back to the discover page. Since you are logged in to the system, we provide you with the feature to create your own playlists. And My love Playlist is the default feature when you click favorite.

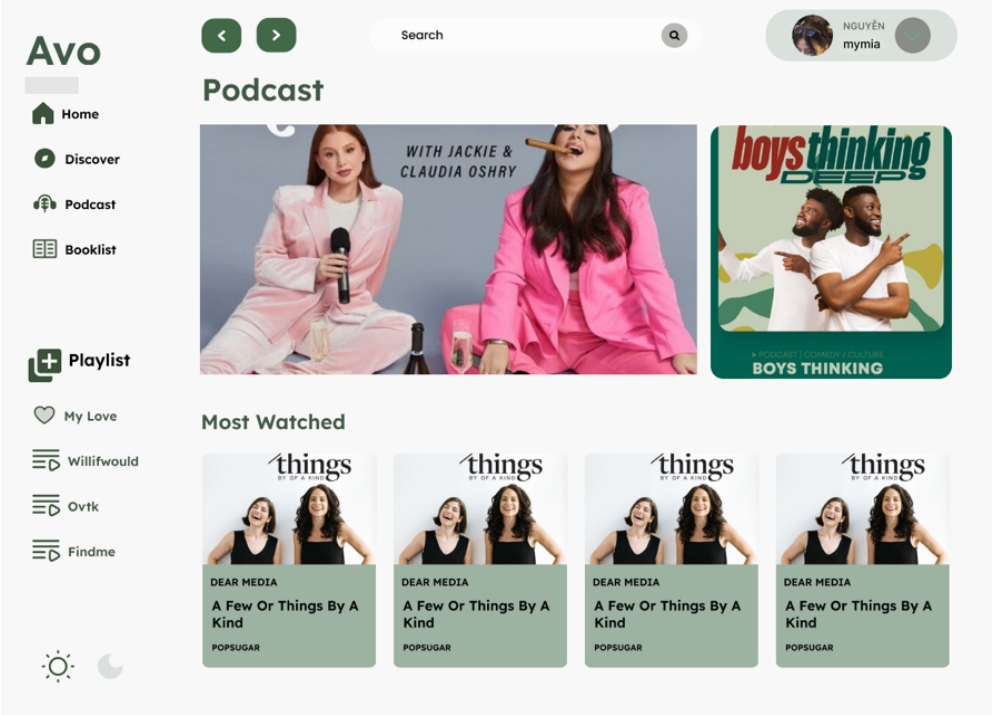

Figure 36 Podcast page with light mode.

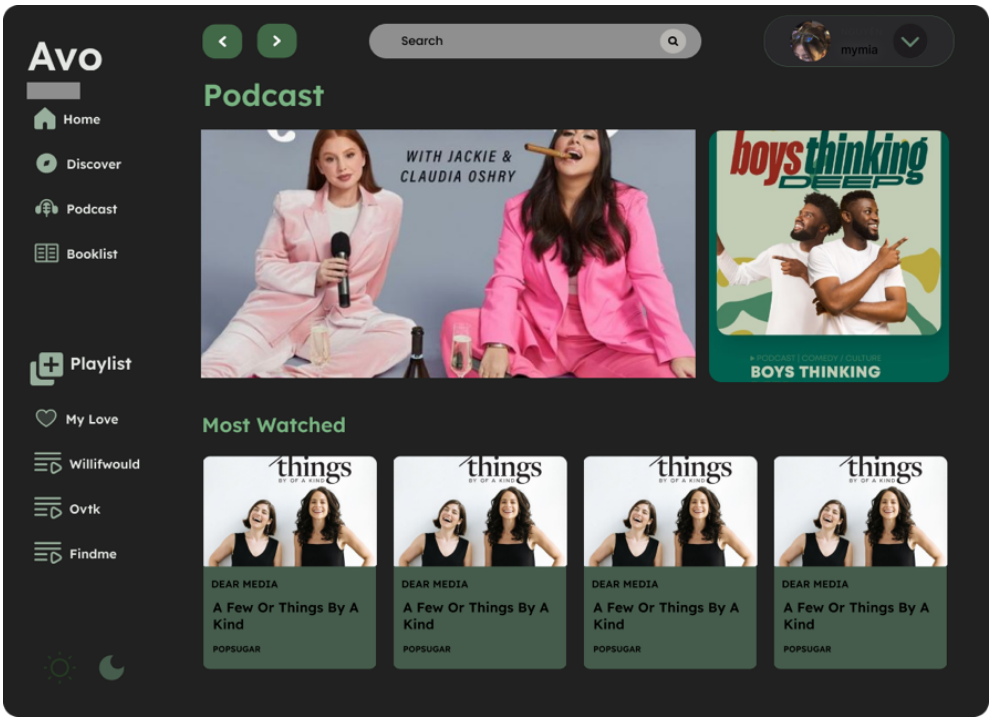

Figure 37 Podcast page with dark mode.

When you click inside the podcast page, we limit Show to you the podcast list by episode so you can listen.

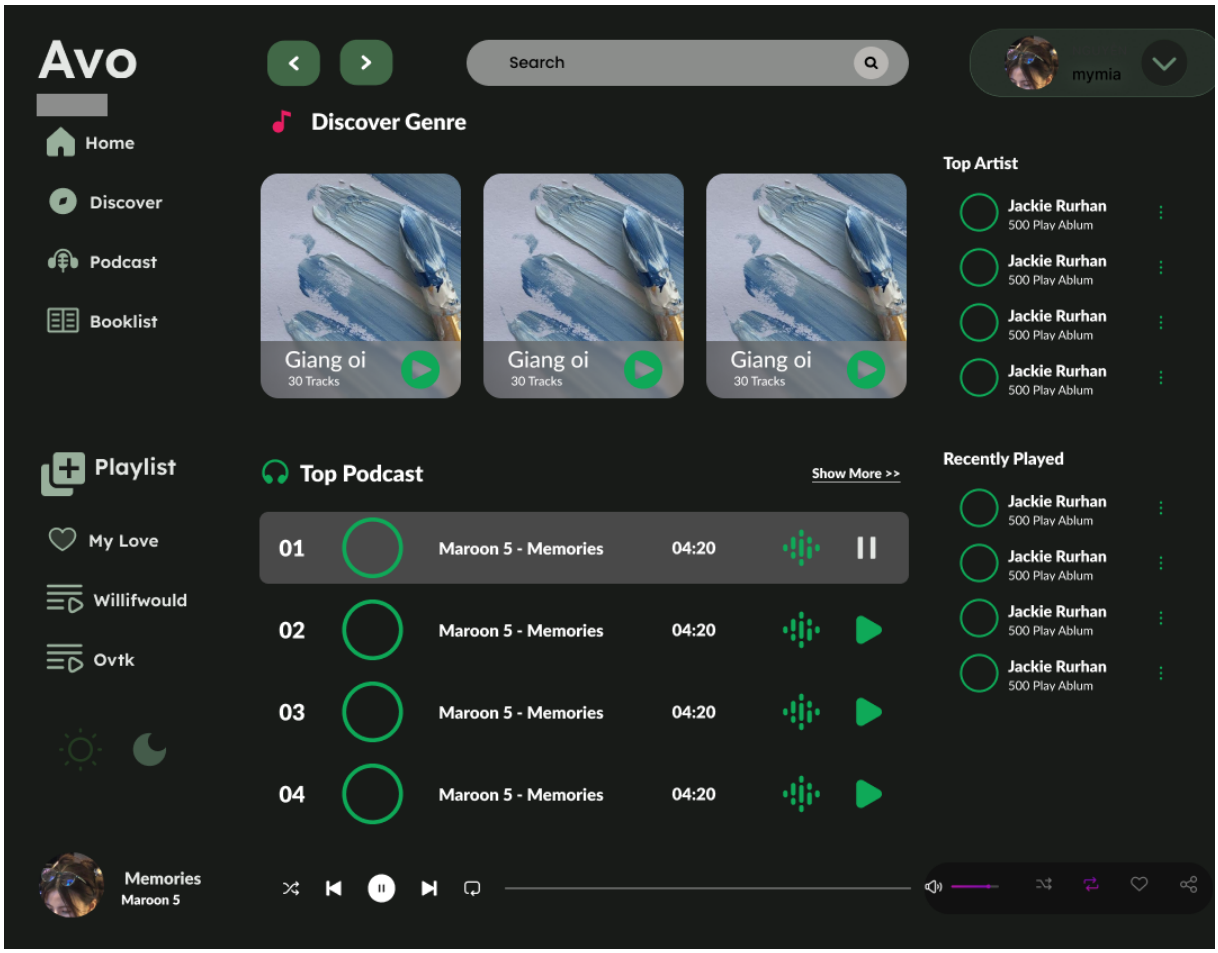

Figure 38 List of podcast page.

3.5. Reading book page

If you choose reading, we will provide you with popular trending books, and, also have 2 light and dark modes for you to choose from like the podcast.. If you are too lazy to read, we also have audiobook mode. So while listening, you can do other things.

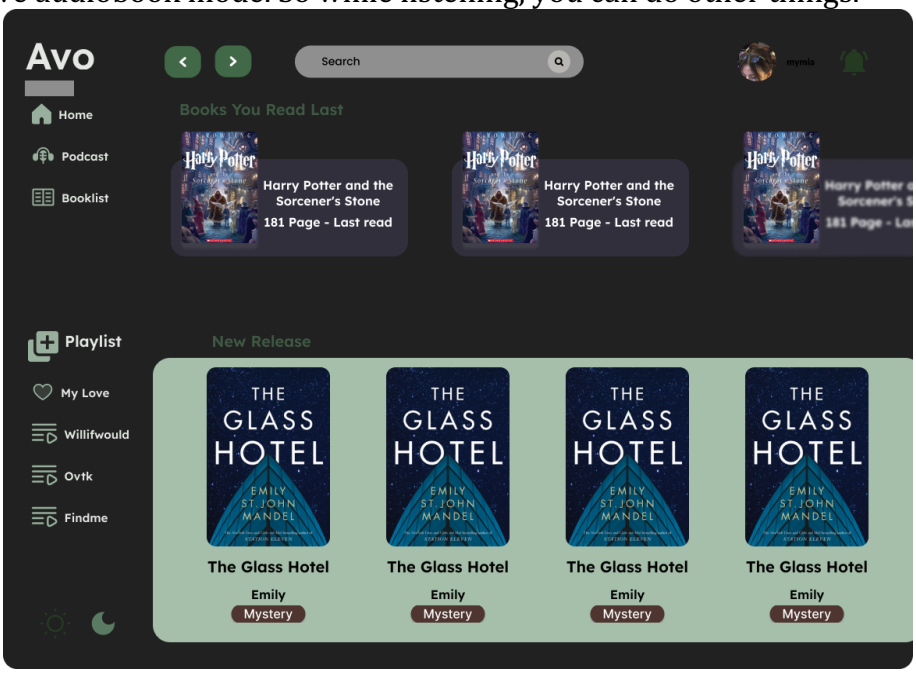

Figure 39 Reading book page with dark mode.

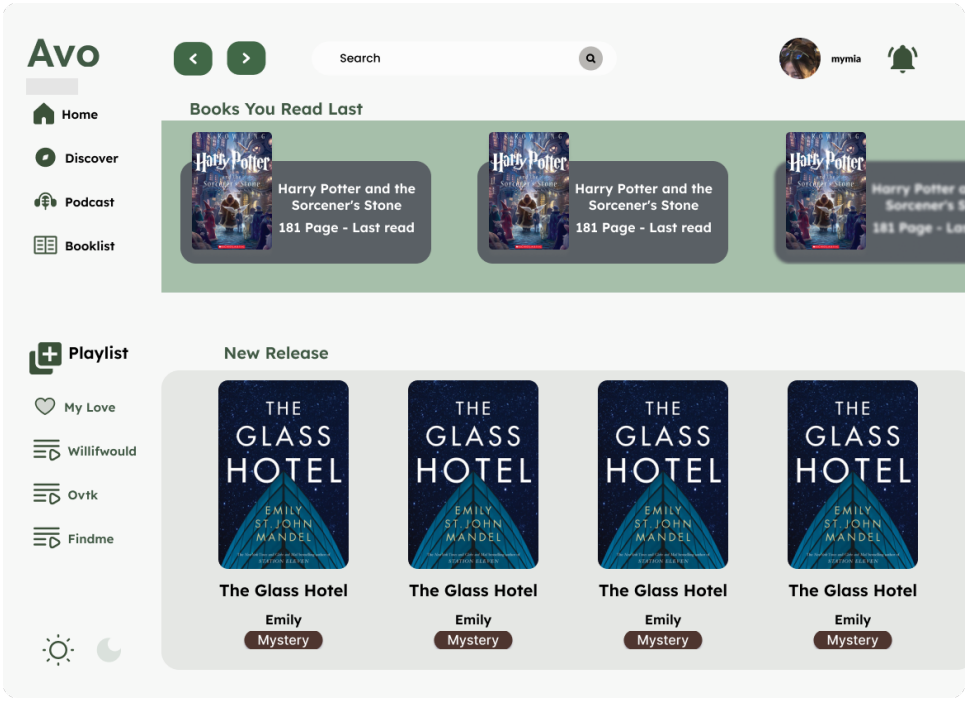

Figure 40 Reading book page with light mode.

CHAPTER 4

DISCUSSION AND CONCLUSION

4.1. Discussion

An The AVO platform is a powerful concept that brings an important step forward of offering people mental health information through digital means. A mobile and Web responsive project has been established based on the requirement of the users found in different groups in society and main aims in the project are to facilitate a user-friendly design in addition to providing information access to a large number of topics. The development has been a structured and well-driven process that follows the Agile and Scrum mentally that is very effective and enhances flexibility for the AVO team so as to make a series of adjustments and improvements of the product in case of an emerging need based on the user feedback and a change of their demands.

Throughout the development phase, the team encountered several challenges:Throughout the development phase, the team encountered several challenges:

- User Interface Adaptation: One of the most significant difficulties was to create a user interface where the intricate functionalities can be used by laymen, as well as experienced users. To make the final interface accessible for unique devices became an additional challenge added to the design.
- Content Diversity and Quality: It was rather time-consuming and required a lot of effort to gather a vast, but comprehensive array of the relevant articles that might meet the interests and demands of as many global citizens as possible. Catering to these requirements as well as guaranteeing that the content posted on the site is both relevant and beneficial to the intended audience while also developing a solid stream of fresh content for the site could prove to be a challenging task.
- Technical Scalability: As the popularity of the platform increased, one of the main technical issues was the ability to manage backend resources to meet the growing levels of visitation efficiently and without a dip in site speed. The efficient incorporation of Internet-capable solutions and the fine-tuning of accessing databases were the foundations able to deliver a seamless experience to the user.
- Security and Privacy Concerns: Due to the social nature of the platform to cater for the needs of the individuals, the platform deals with sensitive user information which includes health and personal growth data thus the need for data privacy and security. Having secure access in handling information has been an essential element in creating a security measure on the system.
- Multilingual Support: To create a platform that could support multiple languages for a worldwide audience, localized listings were crucial; however, language translation and localization posed major issues.

4.2. Conclusion

Accordingly, AVO is a successful platform launched to meet the mental wellness needs of different individuals through offering extensive personal growth and physical health content and also a place to belong. The design and technical performance of the platform have fulfilled its main aim of being user-friendly, safe, and adaptable to different scales: enabling an efficient way for both content consumers and admins. The future steps of AVO would include increasing the content available on AVO and encouraging it.

REFERENCES

[1] D. S. D. G. Sheetal Sharma, "Agile Processes and Methodologies: A Conceptual Study," *International Journal on Computer Science and Engineering,* vol. 04, no. 05, pp. 892-898, 2012.

[2] O. S. Wael Zayat, "Framework Study for Agile Software Development Via Scrum and Kanban," *International Journal of Innovation and Technology Management,* vol. 17, no. 4, p. 2030002 (24 pages), 2020.

[3] A. A. Dragos-Paul Pop*, "Designing an MVC Model for Rapid Web Application Development," in *24th DAAAM International Symposium on Intelligent Manufacturing and Automation, 2013*, Romania, 2013.

[4] M. Thakkar, "Introducing React.js," in *Building React Apps with Server-Side Rendering*, April 2020, pp. 41-91.

[5] J. S. K. W. D. I. D. S. P. U. R. D. Z. G.H.T.R. Irushika, "Saubhagya: An Online Food Donation Platform for Ending Hunger and Malnutrition in Sri Lanka," *Buana Information Technology and Computer Sciences (BIT and CS),* vol. 4, no. 2, pp. 63-75, July 2023.

[6] M. E. Hema Krishnan, "MongoDB – a comparison with NoSQLdatabases," *International Journal of Scientific & Engineering Research, ,* vol. 7, no. 5, pp. 1035-1037, May 2016.

[7] N. M. Patrikalakis, "The Digital Ocean.," in *Computer Graphics International, 2000. Proceedings*, USA, 2000.

[8] D. Hack, "Redux," *Victorian Literature and Culture,* vol. 51, no. 3, pp. 351-353, 2023.

[9] C. S. Massimo Nardone, "JSON Web Token (JWT) Authentication," in *Pro Spring Security*, December 2023.