TED UNIVERSITY

CMPE 491 / SENG 491 Senior Project

<<BooTunes>>

Project Analysis Report

Fall 2024

Team Members:

Mehlika Eroğlu, 58717180232, Software Engineering

Melih Aydın, 28180576276, Computer Engineering

Elif Alptekin, 31376233198, Computer Engineering

Batuhan Dalkılıç, 11491297756, Computer Engineering

Supervisor: Venera Adanova

Jury Members:

Gökçe Nur Yılmaz

Eren Ulu

Firat AKBA

Contents

1. Introduction

BooTunes project is a cross-platform, artificial intelligence-supported, emotion-based music and visualization platform, provides an innovation in the field of personalized hobby in the technology world by integrating artificial intelligence, emotional analytics and media interaction. This innovative system allows users to load digital books, analyzes the emotions conveyed on each page, and breathes new life into the reading experience with music and interactive visuals that dynamically match the mood of the text.

The main aim is to create a platform that attracts individuals to the world of books, with reinforcing elements suitable for many emotional states, enriched by harmonizing the emotional state of the reader with audio and visual elements. This report provides a detailed analysis of the BooTunes project requirements, system specifications, and planned implementation strategies.

2. Current System

Currently, there is no existing system that combines real-time text emotion analysis with dynamic music and visual content generation for books. Traditional e-book platforms lack personalized audio-visual integration. Existing solutions may offer standalone music recommendations or limited visual features, but BooTunes' novel approach leverages AI to provide a truly immersive reading experience. We, as a group, were still in the planning phase for this project, so we haven't moved on to the implementation phase yet.

3. Proposed System

3.1 Overview

BooTunes is a cross-platform application that transforms the reading experience by integrating AI-powered music and visuals. Accessible on mobile devices and computers, BooTunes increases reader immersion by performing real-time emotional analysis of text and adjusting music and visuals to match the predominant mood of each page. Using advanced natural language processing (NLP), BooTunes detects changes in emotional tone (such as tension, joy or melancholy) and perfectly matches the mood with the song selected from categorized tracks. While the app offers blurry backgrounds based on your mood on mobile devices to visualize what you're reading and deliver a more concrete read, PC users experience more dynamic, illustrations that change in sync with the narrative. Beyond its core features, BooTunes offers personalized music taste, bookmarks, and story summaries, as well as recommendations based on reader preferences and past emotional responses. BooTunes redefines reading by blending literature with music and visuals, delivering an engaging, personal, versatile experience for users on every device.

3.2 Functional Requirements

- *User Authentication:* Users can log in and access personalized content using unique accounts with usernames and passwords.
- *Book Library Access:* Users can browse and search for books within a large library of titles.

- *Emotion-Based Audio Matching:* AI-driven analysis of text pages provides music tracks that align with emotional content.
- *Visual Backgrounds*: On mobile, BooTunes provides emotion-based blurred backgrounds, while on larger screens, it displays more detailed visuals.
- *Personalized Playlists and Booklists:* Users may create personalized book lists comprised of books they have read and enjoyed. Also Users can save and access a personalized playlist of liked music tracks based on reading preferences.
- *Bookmarking and Summaries*: The system keeps track of where users last left off. Additionally, to help the reader stay refreshed, they can re-adapt to the book with a summarization system. The user can choose between a summary from the beginning, or a summary of the last parts read, based on their preference, and continue their reading adventure.
- *Offline Access:* Users can download books and associated music for offline reading and listening, allowing for an uninterrupted experience without internet access.
- *Recommendation System:* A recommendation system suggests books and music tailored to the user's preferences and previous experiences in the application.

3.3 Nonfunctional Requirements

- *Cross-Platform Compatibility:* The application must work seamlessly across major operating systems, including iOS, Android, Windows, and macOS, while offering similar functionality and user experience.
- *User Interface (UI) Design Consistency:* A consistent, intuitive UI that ensures ease of navigation across platforms, with accessible design standards to accommodate users with visual or hearing impairments.
- *Responsiveness:* UI components should load efficiently on various screen sizes without compromising quality or performance.
- *Low Latency Audio and Visual Processing:* The system should maintain a low response time (under 200ms) for audio and visual changes, providing a smooth and immersive reading experience without interruptions.
- *High Availability and Uptime*: Ensures the application is available 99.9% of the time to handle user requests and prevent downtime or disruptions during reading sessions.
- *Security and Data Privacy:* Compliance with data protection regulations (e.g., GDPR), implementing encryption for sensitive data (such as login credentials and reading preferences), and secure handling of user information.
- *Energy Efficiency*: For mobile devices, the app should optimize battery usage, especially for background audio and visual processing.
- *Error Handling and Reliability:* Robust error handling to prevent app crashes, with graceful degradation for features that may encounter issues, such as API data retrieval failures.
- *Accessibility Standards:* Compliance with accessibility guidelines (WCAG 2.1) for visually and hearing-impaired users, such as screen reader compatibility and captions for visual content.

- *Internationalization and Localization*: Support for multiple languages and cultural adaptations in both the UI and emotion analysis model, ensuring inclusivity for a global audience.
- *User Privacy Control:* Offers users control over data permissions and settings, including options to disable data tracking or analytics.
- *Performance Monitoring and Analytics*: Continuous monitoring of the app's performance to identify bottlenecks and track user engagement, enabling data-driven optimizations.

3.4 Pseudo Requirements

- *External API Integration for Music:* Robust integration with Spotify or other music APIs, ensuring secure API calls, caching of frequently accessed data, and graceful handling of API downtime or data-fetching errors.
- *Cloud-Based Architecture*: Deployment on a scalable cloud platform, such as AWS, Azure, or Google Cloud, to support high availability, data storage, and scalable ML model hosting.
- *Real-Time Machine Learning Model Updates:* Ability to update machine learning models without downtime, enabling continuous improvement of emotion detection and content recommendations based on user feedback and new data.
- *Offline Mode Support*: Enables users to download books and pre-fetch music and visuals for offline access, ensuring functionality in areas with limited connectivity.
- *Modular and Scalable Codebase*: A modular application architecture (e.g., microservices) that simplifies adding new features or scaling individual components like emotion analysis or recommendation engines.
- *Load Balancing and Auto-Scaling:* Dynamically manage server load and performance, especially for tasks like emotion analysis, music recommendation, and streaming, to handle peak usage efficiently.
- *In-App Feedback System*: Allow users to provide real-time feedback on content recommendations, music choices, and the overall experience, feeding directly into improving recommendation algorithms and user satisfaction.
- *User Personalization Settings*: Detailed customization options for users to set preferences, such as the level of emotional intensity for music, types of visuals, and notification settings.
- *Customizable Reading and Audio Settings:* Users can adjust text size, color scheme, and audio volume within the app, enhancing readability and comfort.
- *Data Backup and Recovery:* Regular data backup and restoration capabilities for user data, including reading progress, playlists, and preferences.

3.5 System Models

3.5.1 Scenarios

Scenario 1: The user wants a deeply immersive experience while reading a book, with audio and visuals enhancing the mood. The user logs in, selects a book, and begins reading. BooTunes dynamically analyzes

the emotional tone of each page and adjusts the background music to reflect shifts in mood—from suspenseful to calm. Simultaneously, the visual background adapts, enhancing the narrative.

Scenario 2: If a user pauses their reading for a month, upon their return, BooTunes provides a summarized overview of the story's events up to their last reading point. Or, based on the user's preference, it summarizes a shorter time frame, like the period between their last two reading sessions. Background music enhances the overall mood of the summary.

Scenario 3: While reading, the user manually adjusts the mood sensitivity settings, preferring a less intense emotional shift. BooTunes responds by making the transitions between music and visuals more subtle, tailoring the experience to the user's preference.

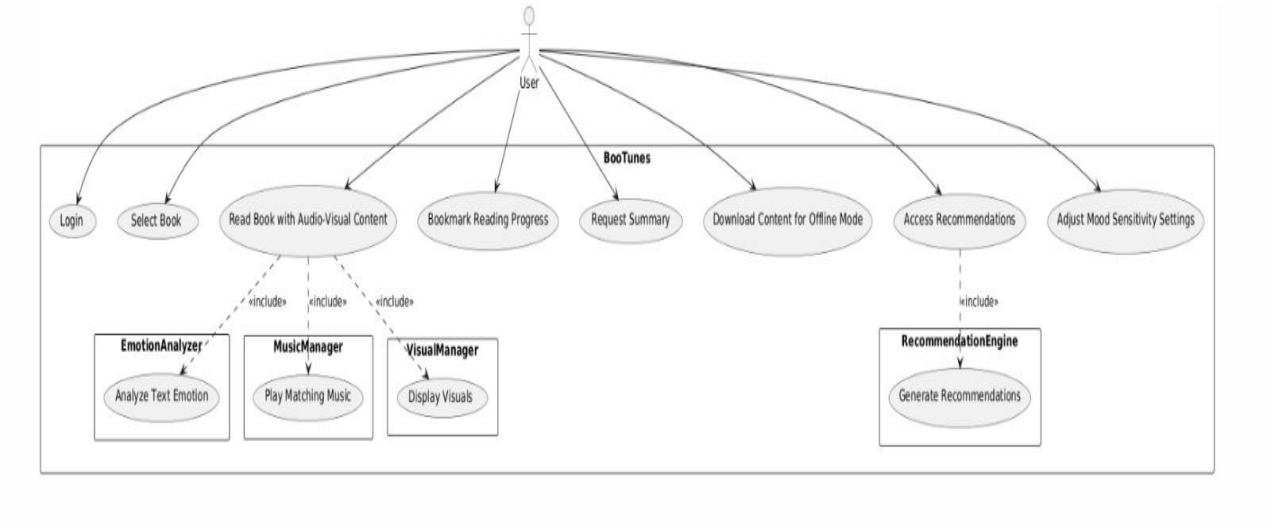
Scenario 4: A user decides to explore recommendations after finishing a book. Based on the emotional tone of the completed book, BooTunes suggests similar titles and curated playlists, providing a smooth transition to the next reading experience.

Scenario 5: The user activates offline mode before a long trip, downloading a book and its related audio content. While reading without an internet connection, BooTunes maintains the same mood-based transitions between music and visuals, creating a consistent experience regardless of connectivity.

Scenario 6: A user starts reading a book on their mobile device during a commute and continues on their PC at home. BooTunes automatically syncs their progress, ensuring a seamless transition between devices.

3.5.2 Use Case Model

- *Login* – Users log in using their credentials.
- *Select Book* – Users search for and select a book to read.
- *Analyze Text Emotion* – AI model analyzes the emotional tone of the text on each page.
- *Play Matching Music* – Th e system selects a corresponding track and plays it.
- *Display Visuals* – Based on the reading device, the app displays suitable visuals.
- *Bookmark & Summary* – Automatically marks the last reading point and offers summaries based on the time away from the book.
- *Access Recommendations* – The system provides book and music recommendations.
- *Adjust Mood Settings* – Users can adjust the intensity of emotion-based effects for music and visuals.
- *Download Content for Offline Mode* – Users can download books and related audio-visual content for offline reading.



3.5.3 Object and Class Model

Key classes in the BooTunes system include:

- **User Class**

  Attributes:
  - username: String
  - password: String
  - preferences: Preferences
  - readingHistory: ReadingHistory

  Methods:
  - login(): Authenticates user credentials.
  - updatePreferences(): Updates user-specific settings like mood sensitivity or preferred genres.
  - getRecommendations(): Fetches book and music recommendations based on user preferences.

  Briefly, the duties of this class are:
  - Stores user data, including preferences and playlists.
  - Manages personalized settings, such as mood sensitivity.
  - Tracks reading and listening history for recommendations.

- **Book Class**

  Attributes:
  - title: String
  - author: String
  - genre: String
  - moodMetadata: MoodData
  - offlineAvailability: Boolean

  Methods:
  - loadContent(): Fetches and prepares book content for the reader.
  - updateBookmark(): Saves the user's current reading position.
  - generateSummary(): Provides a summary of the book or last read sections.

Briefly, the duties of this class are:
- Contains details for each book in the library, including genre and mood metadata.
- Tracks bookmarks and summaries based on user interactions.
- Holds offline availability status for each book.

- **EmotionAnalyzer Class**

  Attributes:
  - emotionModel: NLPModel

  Methods:
  - analyzeText(pageContent: String): EmotionData: Uses NLP to detect emotional tone from text.
  - adjustSensitivity(level: Int): Changes the analysis sensitivity based on user settings.

  Briefly, the duties of this class are:
  - Analyzes text to identify emotional tones and transitions.
  - Provides real-time analysis as the user navigates between pages.
  - Adapts to user mood settings, adjusting sensitivity as configured.

- **MusicManager Class**

  Attributes:
  - currentTrack: MusicTrack

  Methods:
  - selectTrack(emotion: EmotionData): MusicTrack: Selects appropriate music for the detected emotion.
  - play(): Plays the selected music track.
  - pause(): Pauses the current music playback.
  - preloadForOffline(): Pre-downloads music for offline listening.

  Briefly, the duties of this class are:
  - Selects and plays music that aligns with detected text emotions.
  - Adjusts audio transitions smoothly based on emotional shifts.
  - Allows pre-loading of tracks for offline use when enabled.

- **VisualManager Class**

  Attributes:
  - visualSettings: VisualSettings

  Methods:
  - generateVisual(emotion: EmotionData): Creates visuals based on detected emotions.
  - adjustIntensity(level: Int): Adjusts the visual effects according to user settings.
  - cacheForOffline(): Prepares visuals for offline mode.

  Briefly, the duties of this class are:
  - Generates dynamic visuals for mobile and PC views based on analyzed emotions.
  - Adjusts visual intensity according to user settings.
  - Supports caching visuals for smoother offline experiences.

- **RecommendationEngine Class**

Methods:

- generateRecommendations(userPreferences: Preferences): List<Book>: Suggests books and music based on user history and preferences.
- updateRecommendations(): Updates recommendations dynamically as the user interacts with the system.

Briefly, the duties of this class are:

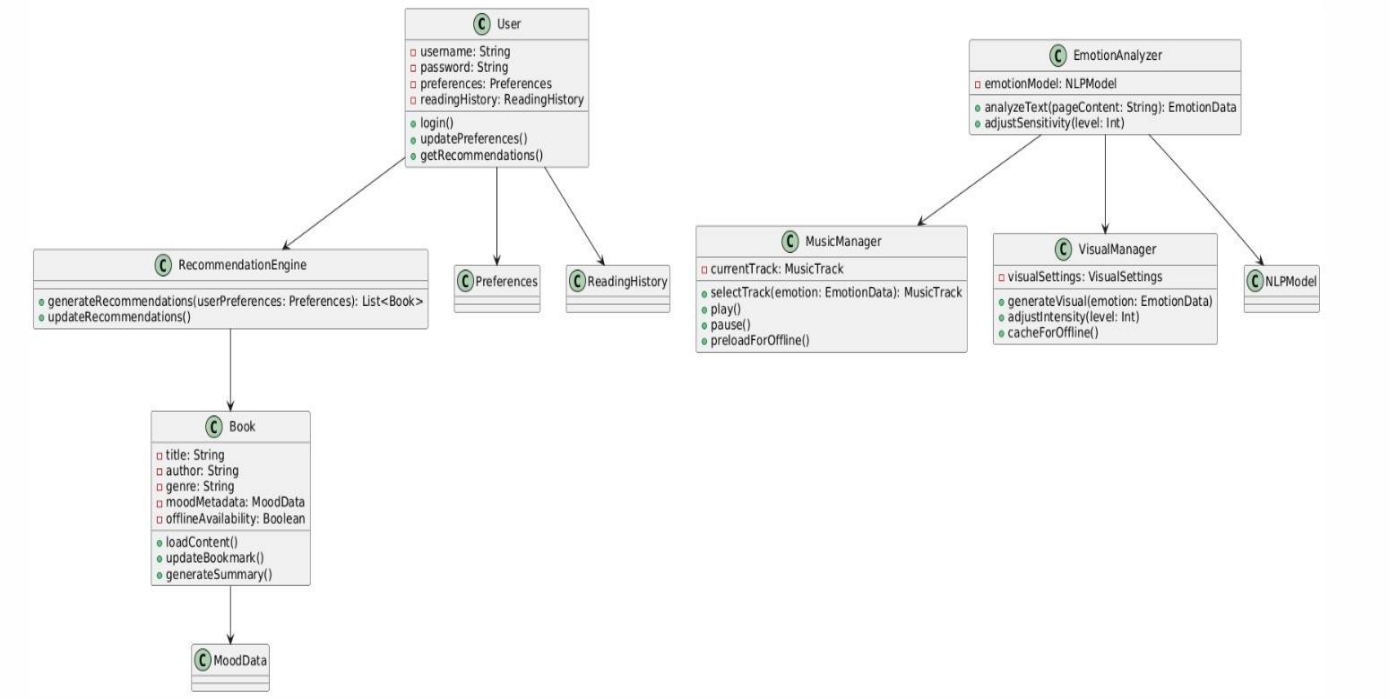
- Recommends books and music based on user preferences and recent history.
- Suggests similar content based on emotional themes of previously read books.
- Updates recommendations in real-time as user preferences evolve.

- **MoodData Class**
  - Represents metadata related to the mood of a book, such as detected emotional tones or transitions.

- **ReadingHistory Class**
  - Tracks user activity such as previously read books, last read position, and associated emotional responses.

- **Preferences Class**
  - Stores user-specific settings, including genres, preferred mood sensitivity, and visual or music customization options.

- **NLPModel Class**
  - Used internally by EmotionAnalyzer to process text for emotional analysis.

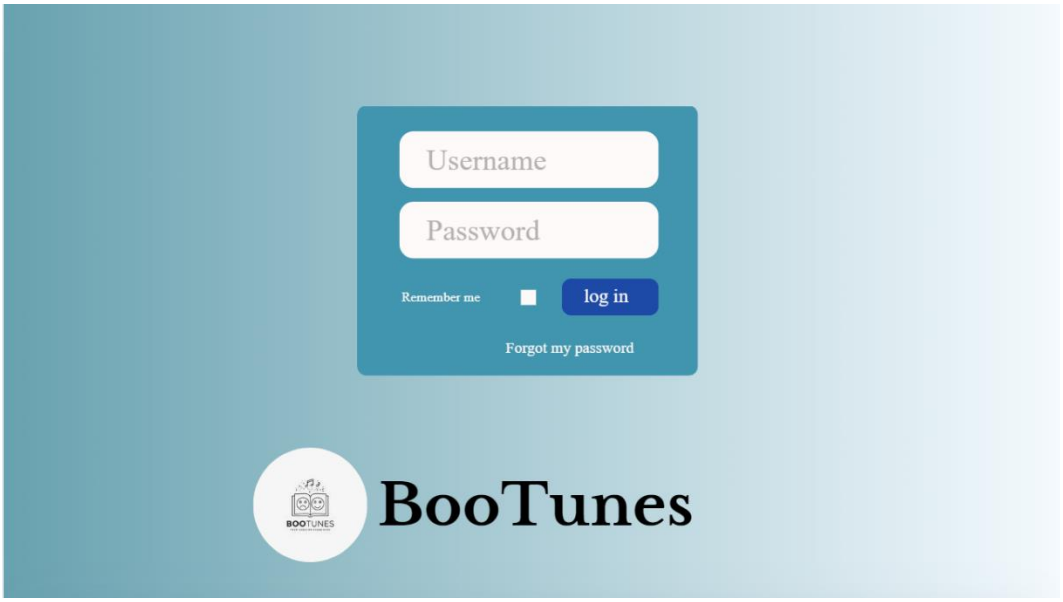


3.5.4 Dynamic Models

- *Text Emotion Processing:* After the user loads a book, the EmotionAnalyzer runs text through NLP models to detect emotional tones, then sends output to the MusicManager and VisualManager.
- *Real-Time Adjustments:* MusicManager dynamically changes the track, and VisualManager adjusts visuals based on each page's mood.
- *User Interaction Flow*: Users navigate between the library, reader, and playlist sections, while the RecommendationEngine updates suggestions based on recent interactions.
- *Continuous Learning Feedback:* The system collects feedback from user interactions with recommended music and visuals, allowing for fine-tuning of recommendations and improving emotional accuracy over time. (Feedback is logged in ReadingHistory and preferences, RecommendationEngine fine-tunes suggestions and improves prediction accuracy.)
- *Offline Mode Support*: When enabled, the system loads pre-downloaded music tracks and visuals based on detected text emotions, allowing uninterrupted offline reading according to MusicManager and VisualManager.

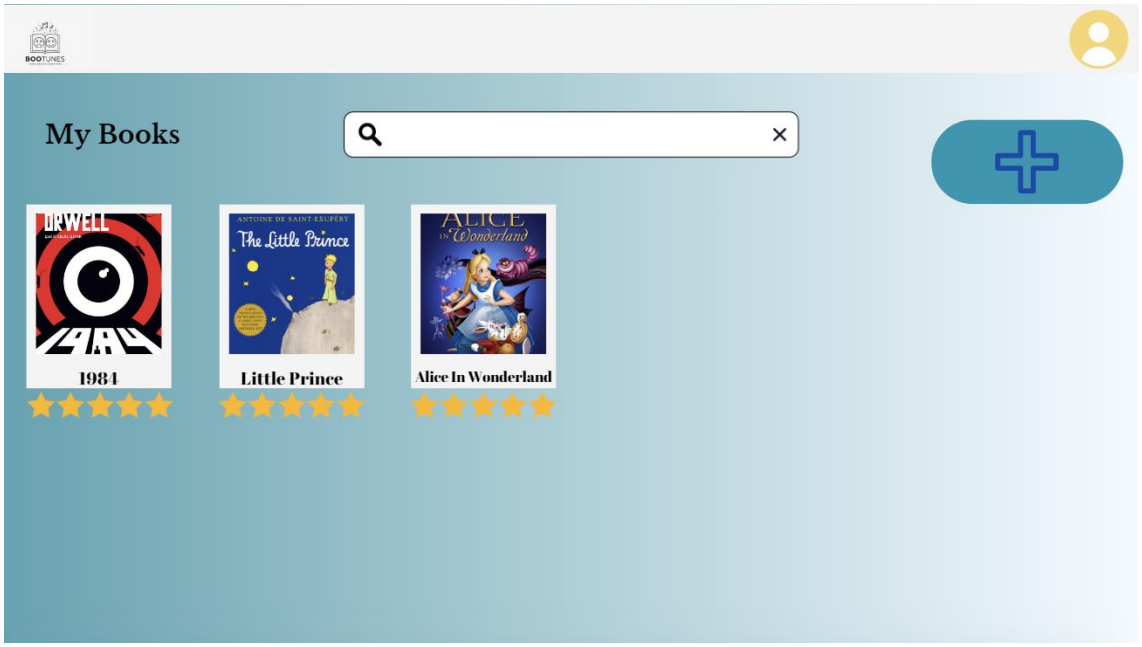
3.5.5 User Interface - Navigational Paths and Screen Mock-Ups

- **Log in page:** The login page is the gateway to the application, where users can either create an account or log in to an existing one by entering their username and password. New users can sign up quickly by providing their preferred nickname and setting a secure password. Returning users can log in to access their personalized settings, preferences, and progress. Additionally, error handling for invalid credentials is integrated to guide users with prompts like "Incorrect username or password." A "Forgot Password?" feature redirects users to reset their password securely.

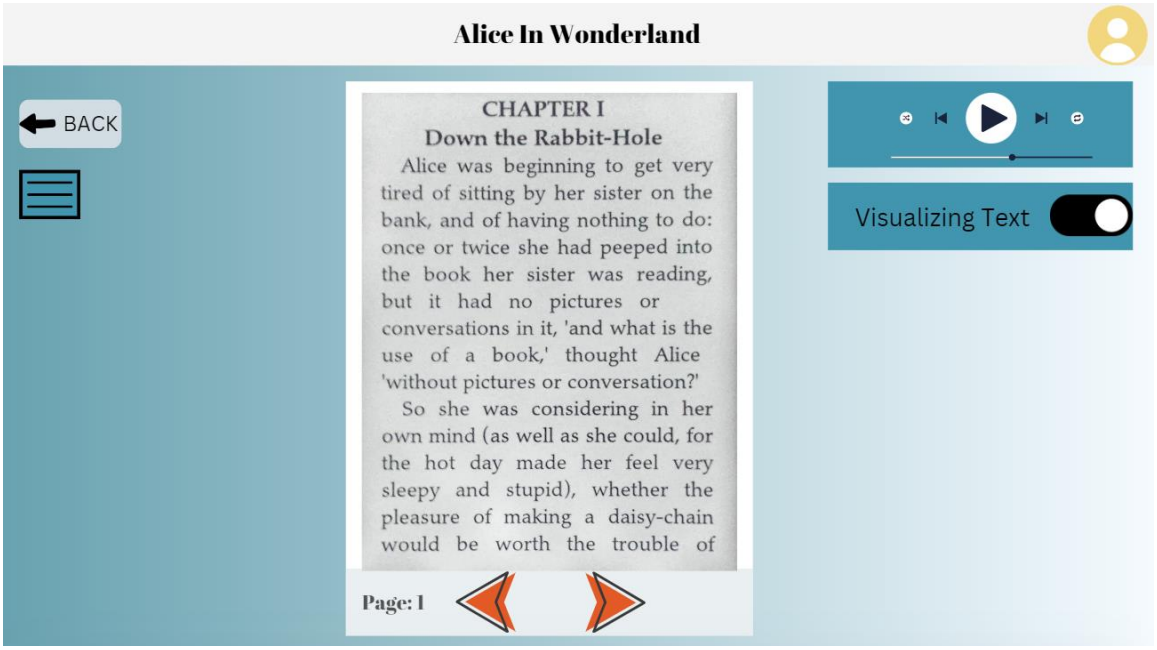


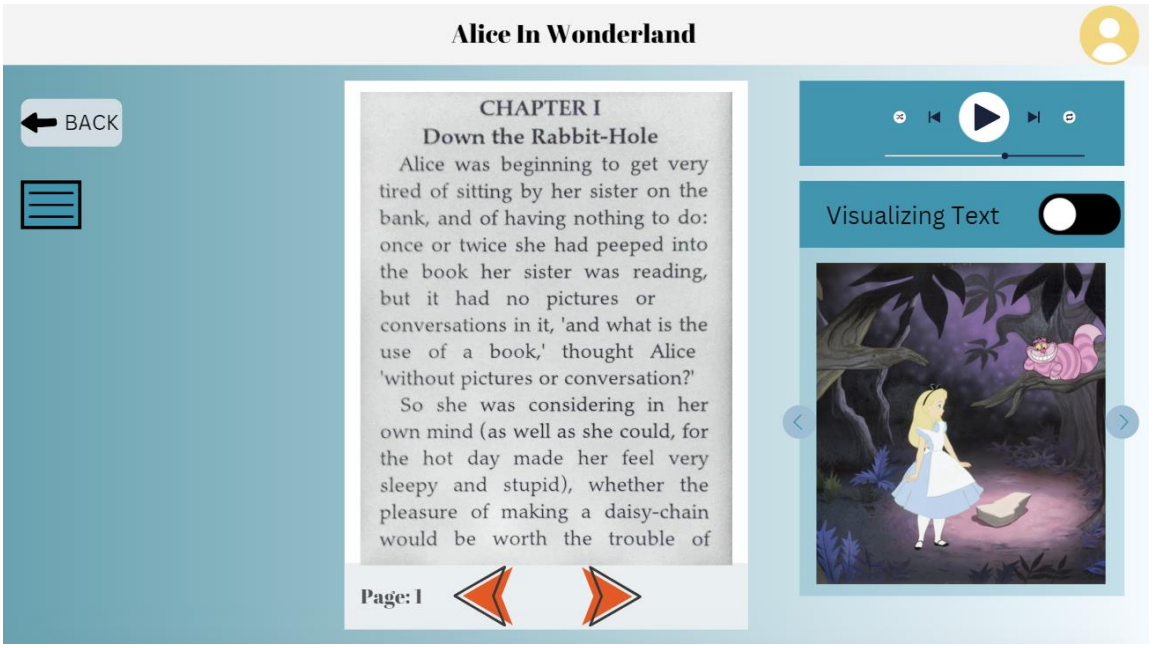
- **Home Screen:** The home screen serves as the central hub of the application, offering intuitive navigation to key features. Users can access their personalized book library, curated music playlists, and dynamically updated recommendations based on their preferences and reading history. The screen showcases visually appealing thumbnails for books and music tracks, along

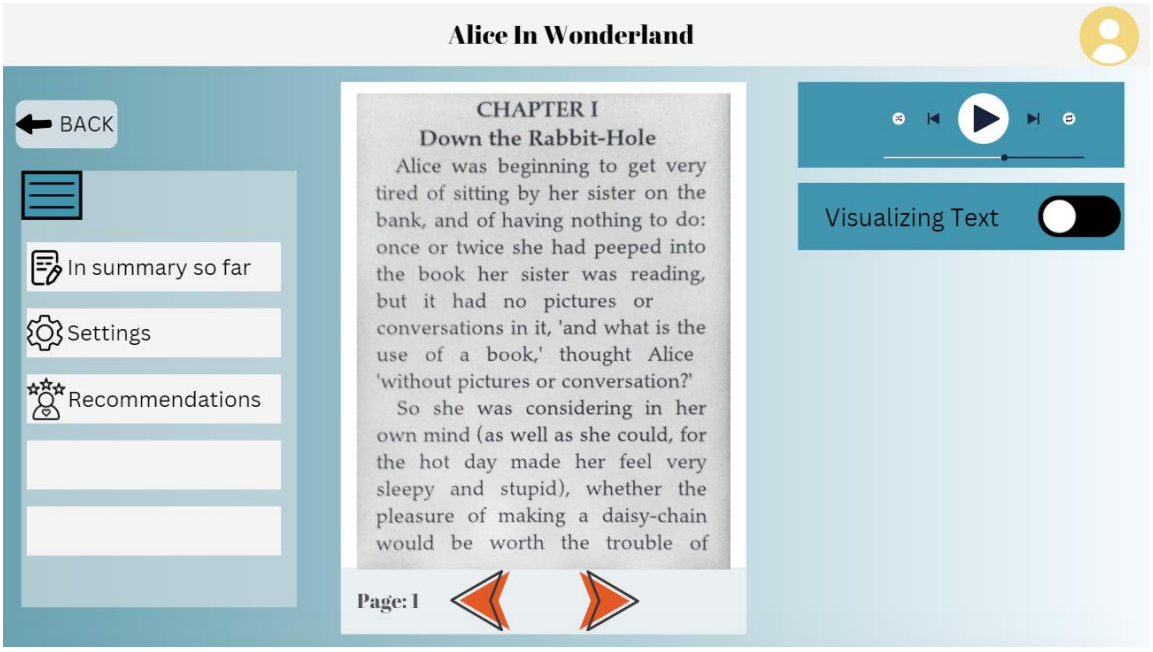
with shortcuts to recent activities. A quick search bar enables users to find books or tracks instantly.

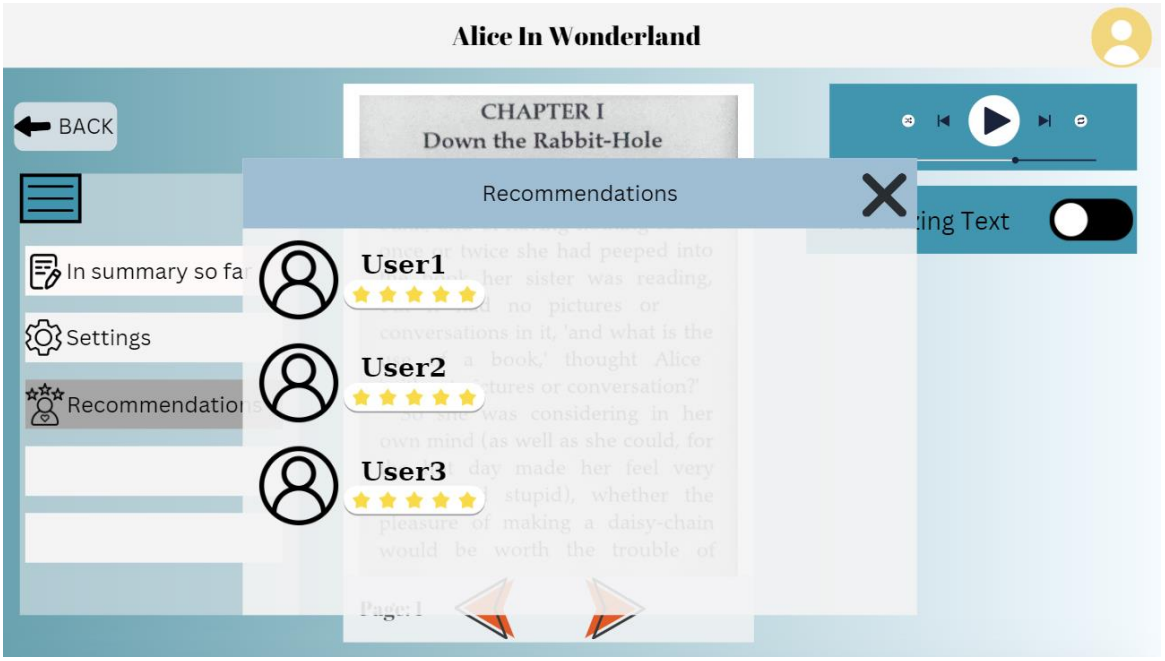


- **Book Reader View:** The book reader view provides a seamless and immersive reading experience by displaying the book's content alongside integrated audio and visual overlays. Readers can adjust the font size, background color. The user also can display the summary so far and recommendations about the book. As users read, the system analyzes the text to detect emotions, dynamically playing mood-matching background music and displaying ambient visuals. Features such as in-app summaries ensure a smooth reading flow, while offline accessibility allows uninterrupted reading even without an internet connection.

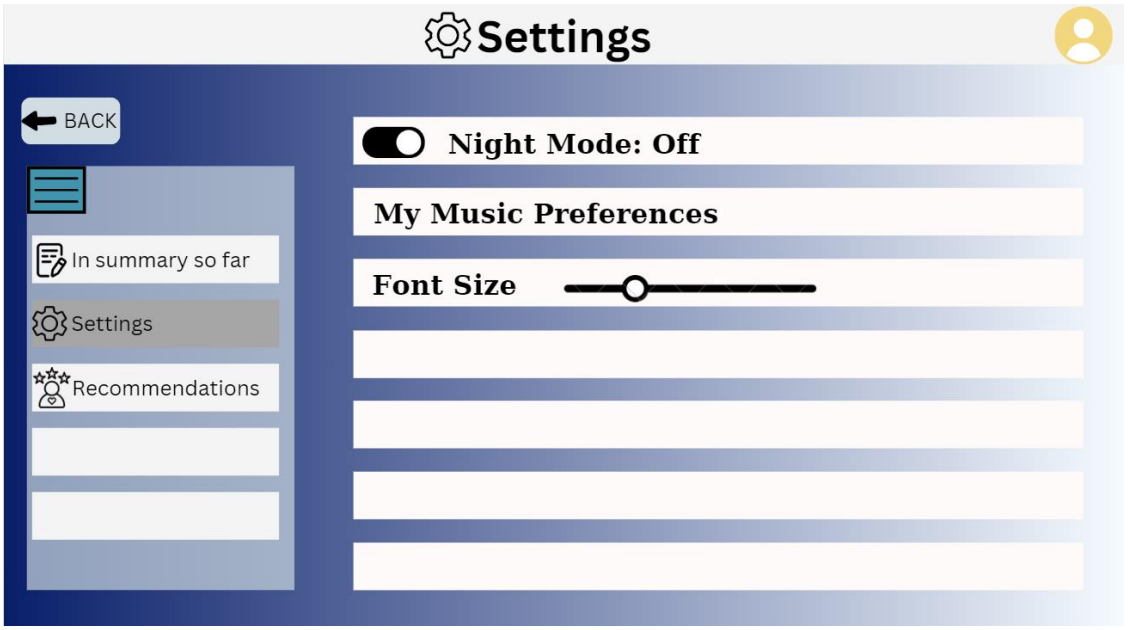



Alice In Wonderland
BACK
CHAPTER I
Down the Rabbit-Hole
Alice was beginning to get very tired of sitting by her sister on the bank, and of having nothing to do: once or twice she had peeped into the book her sister was reading, but it had no pictures or conversations in it, 'and what is the use of a book,' thought Alice 'without pictures or conversation?'
So she was considering in her own mind (as well as she could, for the hot day made her feel very sleepy and stupid), whether the pleasure of making a daisy-chain would be worth the trouble of
Page: I
Visualizing Text


Alice In Wonderland
BACK
In summary so far
Settings
Recommendations
CHAPTER I
Down the Rabbit-Hole
Alice was beginning to get very tired of sitting by her sister on the bank, and of having nothing to do: once or twice she had peeped into the book her sister was reading, but it had no pictures or conversations in it, 'and what is the use of a book,' thought Alice 'without pictures or conversation?'
So she was considering in her own mind (as well as she could, for the hot day made her feel very sleepy and stupid), whether the pleasure of making a daisy-chain would be worth the trouble of
Page: I
Visualizing Text



- **Settings:** Allows users to customize their experience with features like a **Night Mode toggle**, **music preferences** for emotional alignment, and a **Font Size slider** for comfortable reading. These options are accessible through a clear and simple interface, ensuring a user-friendly and personalized experience.



4. Glossary

- *AI (Artificial Intelligence):* The simulation of human intelligence by machines, enabling them to perform tasks such as reasoning, learning, and problem-solving.
- *NLP (Natural Language Processing):* AI subfield focused on processing and analyzing human language.
- *Emotion Analysis:* A technique for detecting and interpreting the emotional tone or sentiment expressed in a piece of text, commonly used in applications like sentiment analysis or emotional categorization.
- *Spectrogram:* A graphical representation showing how the spectrum of sound frequencies varies with time, often utilized in music analysis and processing for visualizing audio signals.
- *API (Application Programming Interface):* A set of rules and protocols allowing different software applications to communicate with one another, enabling integration and functionality sharing between systems.

5. Appendix: Technical Stack and Frameworks

- **Frontend:** React Native or Flutter for mobile, React for PC and web interfaces.
- **Backend:** Django or Flask for server-side logic and ML integration.
- **Database:** MongoDB for user data, bookmarks, summaries, and playlists.
- **Machine Learning Frameworks:** Hugging Face Transformers (BERT, RoBERTa), PaddleOCR for text emotion analysis; PyTorch and TensorFlow for music analysis models.

This analysis document serves as the foundational contract for BooTunes, covering the user needs, system specifications, and a structured roadmap for development. The design choices, technical

implementations, and user interface pathways are tailored to create a seamless AI-driven reading experience.

6. References

- *Fielding, R. T. (2000). Architectural Styles and the Design of Network-based Software Architectures (Doctoral dissertation). University of California, Irvine.*
- *https://en.wikipedia.org/wiki/Spectrogram*
- *https://www.nltk.org/*