BOOKBYTE

Functional Specification

(SUPERVISOR) JAMAL
(STUDENT C00260735) CONOR BROOKE
HABIT-BUILDING READING APP ON ANDROID

Table of Contents

1. Introduction

BookByte is an Android application designed to improve users reading habits. The core features required to reach this objective are outlined throughout this specification. This section provides a brief introduction to the primary functionalities of BookByte.

BookByte enables users to upload reading materials, including books and research papers, to the app's library. The app segments uploaded material according to user metrics like reading preferences and skills. This app provides options to later adjust the size and complexity of these reading segments, granting users customizability. BookByte is also designed to dynamically adapt reading materials to align with user's abilities and schedules.

Additionally, BookByte includes features that boost motivation, like reading streaks (gamification) and progress tracking. This app integrates the Androids notification system, to prompt users to complete their daily reading. Sending its users daily reminders can serve as another habit building feature of the app.

The purpose of the BookByte application is to facilitate regular reading and learning through its core features. This specification will detail the objectives, use cases, and functionality required to develop this application.

1.1. Purpose Of Specification

This specification provides all functional and non-functional requirements for this application. The document gives in-depth descriptions as well as diagrams, to help guide the development and testing of BookByte.

Wikipedia defines "a functional specification (FSD) in software development as a document that specifies the functions that a system or component must perform" (Wikimedia Foundation, 2023). This document provides the app's blueprint, detailing requirements in clear use case diagrams and descriptions. This provides each feature with a visual representation and a clear path of operation.

This functional specification covers all of BookByte's functionalities outlining what this application must do. It also offers metrics for the non-functional requirements, which are "requirements that specify criteria that can be used to judge the operation of a system, rather than specific behaviors" (Wikimedia Foundation, 2023).

1.2. Project Scope

Project’s Objective

The project’s objective is to implement the core features /requirements detailed in this specification. Most requirements specified in this document align with BookByte’s primary goal of building its users reading habits. Implementing functionality such as the uploading of reading material and text segmentation is essential for meeting this objective.

Project’s Target Audience

The target audience for BookByte are those that struggle to engage with reading material. These readers often lose interest and get distracted when the material becomes overwhelming. Which leads them to procrastinate and continuously ignore the material until a deadline or avoid the material completely.

This audience may become frustrated when attempting to engage with popular reading apps like Kindle, ReadEra, and Apple Books. They might have tried other apps but were never given optimal features to improve their engagement with reading. This audience seeks an alternative app that offers a solution to this problem. BookByte can offer some solutions like dividing their overwhelming text into manageable chunks.

Another audience this app targets are those that struggle to find time to read consistently. These people often have packed schedules and prefer their content to be short and concise instead of long and detailed. It may be overwhelming when they considered the amount of time sacrificed to finish a book. BookByte can provide appropriately sized material to fit all schedules, as users are able adjust the segment size. This feature can prevent the uncertainty felt when users are reading large and complex material.

BookByte is providing a valuable niche in the Android reading-app market, by implementing features that can manage and segment users reading material. Using BookByte’s unique features anyone can stay engaged and enjoy reading.

Project’s Content Scope

The content scope of this app is books and research papers supporting uploads in PDF and EPUB formats. The scope of this content is primarily large texts that better suit the functionality of this app. The project’s initial scope is delivering a Minimum Viable Product (MVP) that allows uploaded content to be segmented. This initial product will implement an accessible and distraction-free UI, to make content uploads effortless.

There is potential to increase the app’s scope in later iterations, like the ability to upload more file formats like XLSX, DOCX. Additionally, adding features similar to those seen on Kindle like transferring notes from BookByte to an external HTML file.

1.3. Application Objectives

To Build Users Reading Habits:

Objective

This app attempts to help build a consistent reading habit for its users.

Features Required

This app will provide this through the implementation of ‘chunked content delivery’, the app will supply its users a new chunk of text each day.

Provide a Customizable Reading Experience:

Objective

This app allows its users to adjust settings, like reading segments size and font size, etc., to better suit their preferences.

Features Required

This app will provide customizability features allowing users to adjust the segments size and complexity. Users are shown this functionality after the app’s original launched. BookByte also dynamically adapts segments sizes to consistently improve a user’s reading.

Keeping User Engagement:

Objective

Try to have users engaged with BookByte, to keep them regularly returning to the app.

Features Required

The app will provide engagement features like notifications and gamification to motivate users to maintain their reading streaks.

User-Friendly Interface Design:

Objective

Providing an easily accessible and distraction-free reading UI and UX.

Features Required

This app will provide an accessible and distraction-free UI, that allows users to easily access the primary features of the app. BookByte will offer a user-interface design similar to the UI/UX seen in Kindle and ReadEra.

2. Problem and Proposition

2.1. Problem Statement

A large percentage of people, especially American teens, and young adults lack the motivation to read. Recent research from the American Psychological Association (APA) revealed that only 20% of American teens had read a book, magazine, or newspaper for enjoyment in 2018, compared to over 80% of teens using social media daily (American Psychological Association, 2018). This research highlights a potentially untapped market for BookByte to offer its niche features.

One of the challenges that could be impacting reading habits today is a scarcity of time. Research published by Amy Watson shows "on average the daily time spent reading among American adults was just 15.6 minutes in 2022" (Watson, 2023). While this study doesn't correlate busy schedules with a reduction in reading, it does highlight a decline in reading habits. Many factors could also have impacted this data, like people that are illiterate or people uninterested in using free time to read.

While this app may not solve all issues identified in these studies, it does offer a practical solution by dividing reading content into manageable chunks. By aligning reading segments to match the user's schedule and skills, the app ensures that reading stays accessible and engaging.

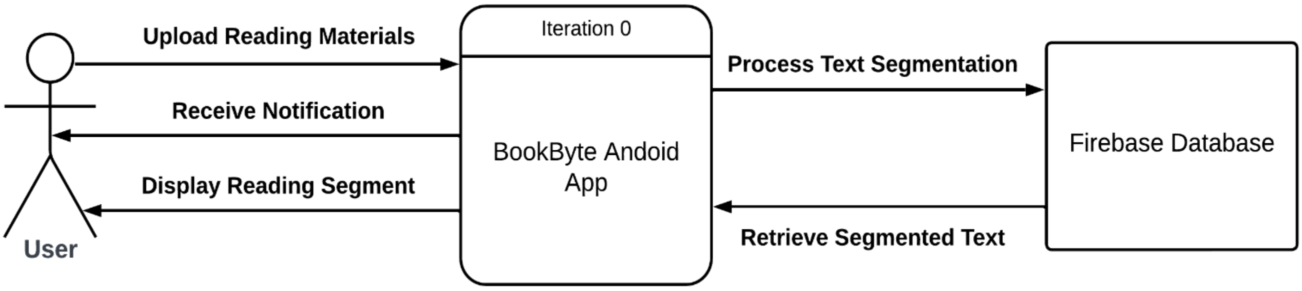
2.2. Value Proposition

BookByte stands out by offering unique reading improvement and habit-forming features. It is designed to transform reading into a consistent and engaging habit. By allowing users to upload their own reading materials, ranging from books to research papers, BookByte provides a personalized reading environment. The app can segment reading material based on user metrics such as reading preferences and skill level. BookByte's also offers the ability to dynamically adjust the size of reading segments to ensure each user's experience is tailored to their current abilities. This app targets a wide range of users from current readers to those looking to improve their reading habits.

3. System Overview

This section includes a high-level overview of the system detailed in this specification. It covers the context and data flow of the core functionality in the system. This section also highlights the 'Assumptions' and 'Dependencies' that will be required for this application to function as expected.

3.1. System Context Diagram



The context diagram provided gives a visual representation of BookByte interacting with important external entities. It displays, at a high-level, the data flow expected between a User and the Android system. The diagram also displays the expected data flow between the Database and the Android system.

The diagram currently specifies Firebase as the database, this is providing a lower level of context than required and will be changed in the next iteration. The context diagram illustrates how the system will interact with external entities, detailing the interfaces that will be established based on the requirements in this specification.

3.2. Assumptions and Dependencies

Assumptions:

- Users have a basic understanding of Android OS.
- Users can download and upload documents without assistance.

Dependencies:

- Android OS version 14, or latest version, for optimal performance.
- Minimum storage requirements to install and run the app.

4. User Characteristics

4.1. User Summary

Users attempting to read consistently:

People attempting to read consistently and are struggling to keep it a habit.

Users seeking self-improvement

Users that are looking to track their progress and grow their reading abilities.

Students Overwhelmed by Material

Students can benefit from the app's ability to break content down into smaller chunks making researching academic papers more manageable.

5. Requirements

This section of the specification details all functional and non-functional requirements for BookByte. It provides detailed descriptions of the FURPS required in this app. The acronym FURPS refers to the functionality, usability, reliability, performance, and security of an application. As software development becomes more efficient and cost-effective, offering innovative features is no longer enough. Improving this applications FURPS could provide a major impact to usability. The section following covers a descriptive overview of the apps requirements.

5.1 Functionality:

5.1.1 Core Features:

This section covers the core features required in this application. Implementing these features as described will allow this app to function as expected. The headings following provide an overview of this applications core features.

Upload Reading Material

This is an important feature that enables users to upload PDF and EPUB files to their personal libraries. Users can upload one file at a time and uploading a valid file will triggers the systems segment reading material feature. A progress indicator is displayed during the segmentation of this uploaded material. Once segmentation is complete, all content is stored in the user's library, and can be access through the 'View Reading Material' feature.

Segment Reading Material

The segmenting of reading material is the primary functionality of this application. After users upload reading material to the BookByte app, the system automatically triggers this segmentation functionality. The system initially analyses the material, collecting necessary details to ensure accurate text segmentation on the sever. This process is critical for allowing the server to return coherent reading segments.

Once analysis is complete, the system notifies the server that the material can be sent for processing. The server then segments the material, avoiding sudden breaks in paragraphs and chapters. The segmented text is temporarily stored in the database until it is transferred back to the user's device, where it remains until the 'Adaptive Reading Segment' algorithm prompts the users to recalibrate.

Adapt Reading Segments

Adapt Reading Segments is a complex feature of this application, it dynamically adapts the segment length of text based on user's reading metrics. Users may trigger this feature after completion of a daily text segment. Users can also be prompted through a notification from the app.

The algorithm utilizes a collection of metrics accessible within the app to accurately calculate the difficulty level. This feature executes dynamically offering adapted segments to match the users improved reading.

Difficulty Metrics:

- **Current Reading Streak**: Keeps track of the user's consecutive days of reading. The current streak impacts the segment length after reaching a milestone set by the app.
- **Average Streak Lifecycle**: Checks user's prior reading streak patterns to find a certain trend. Algorithm adapts during challenging periods for the user (consistently quitting around certain milestone).
- **Duration to Read a Daily Segment**: Track the time it takes a user to read a segment, provides reading speed of user. Always consider reading speed when adapting reading material.
- **User-Set Preferences**: Preferences set manually by the user like segment size.
- **Length Away from App**: Time away from the application in days. Takes a major hit on difficulty level to entice users back to app.

The algorithm decides if an increase in difficulty is needed after calculations. The system prompts users to recalibrate materials when requested by the algorithms. The user can reject the recalibration which impacts the algorithm. The difficulty level in this iteration of the project will only impact segments size.

5.2.1 Additional Features:

The additional features are provided in order of importance to the application.

1. View Reading Material

Allows users to view (open) uploaded reading materials. Will be most used feature by users.

2. Complete Reading Segment

Triggers when a user finishes a reading segment, essential for tracking metrics, building streaks and participates in the functionality behind this apps core features.

3. Update Streak

Brings gamification strategies to the app, it tracks and encourages consistent reading, and improves user engagement.

4. Receive Reading Reminder/ Notification Service

This feature sends users reminders for streaks and alerts for text adaption, can help promote a daily reading habit.

5. Adjust Reading Environment

Can adjust features, by modifying reading settings like font size for a better reading experience.

6. Manage Library

Users can organize and access all their uploaded reading materials through the library collection.

7. Register Account

Users can create an new account within the application.

8. Login to Account

Users can successfully log into the app.

9. CRUD Notes and Highlights

Users can create, read, update, and delete notes or highlights enabling users to interact with the text.

5.2 Usability:

BookByte ensures a relaxing reading experience by providing users with a simplistic UI/UX design. Learning from the design of similar applications I've researched, a clutter-free design is essential for a reading app. This allows users to easily traverse the application, from uploading reading material to completing a daily segment, all features can be completed with ease. Also providing users with customization options, like adjusting the reading segments and environment can improve the app's usability further.

5.3 Reliability:

All BookByte's functionality is designed to encourage a daily reading habit, so if the app is unreliable so will its users habits. Important features like 'chunked content delivery' and the notifications service should work reliably, ensuring that users receive the correct segments and reminders without any errors, keeping users trust in the application.

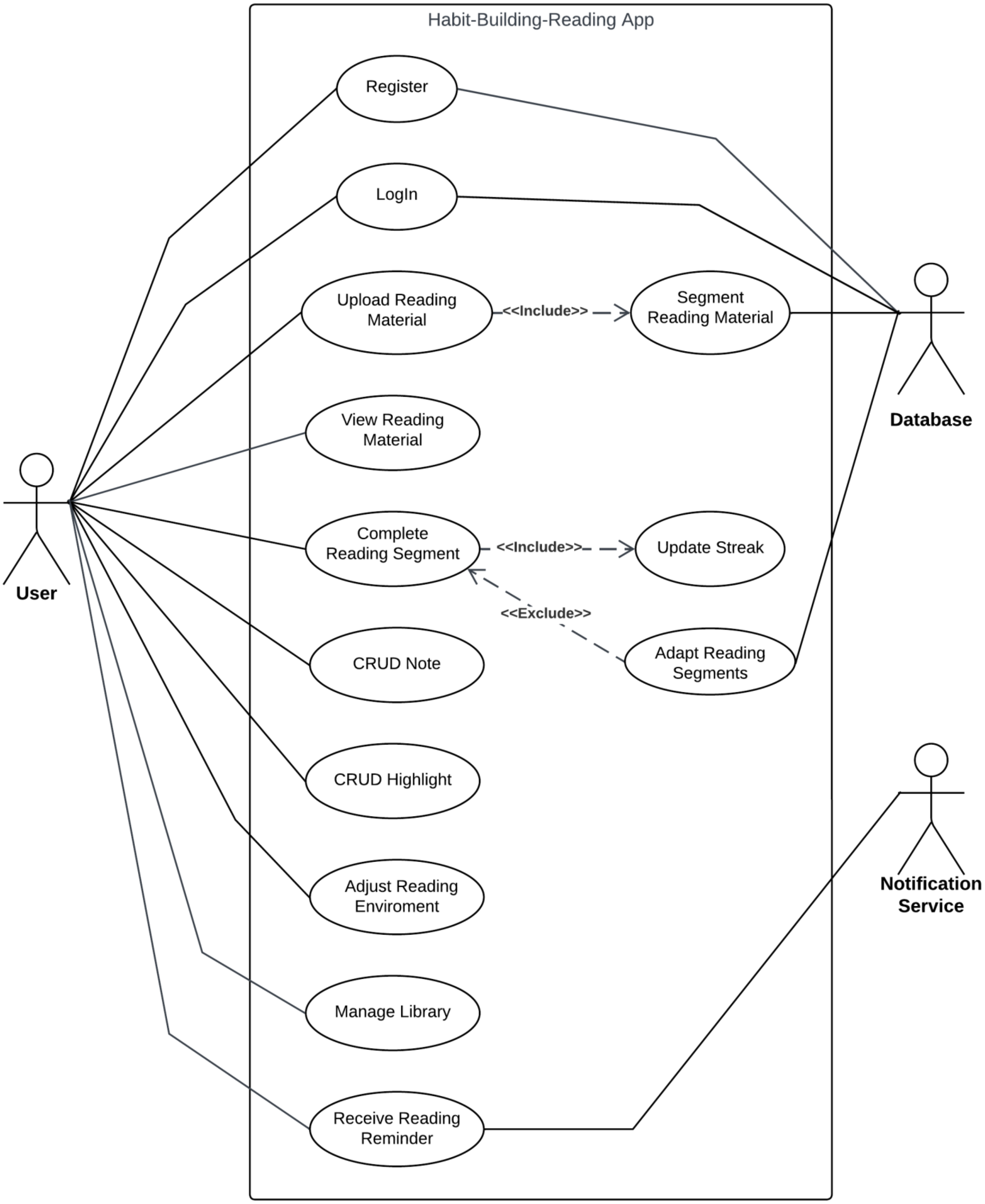
5.4 Performance:

The performance of an app is vital for retaining user engagement. With BookByte, users are expected to use the app daily, any major lag can lead a user's to abandon the app completely. To prevent poor performance, the app should be optimized to load quickly, ideally within 3 seconds on normal Android devices. This ensures that readers can quickly access their reading segments, notes, and highlights without unnecessary waiting, making their daily reading routine smooth and enjoyable. Loading segmented text will be need some brainstorming, to improve performance and upload times.

5.5 Security:

It's important to ensure BookByte's user data remains private and protected, as the target users of this app will be uploading personal reading materials and personal making notes and highlights.

6. Use Case Diagram



7. Detailed Use Cases

7.1 Upload Reading Material

Use Case Name:	Upload Reading Material	**UniqueID:**	UC001
Description:	User can upload valid reading materials to the app as a file. System then notifies user to wait, automatically triggering apps text segmentation feature. User is prompted when the system successfully segments uploaded material. System uploads all text segments to users library. User can now trigger the systems View Reading Material feature.		
Actors:	User		
Preconditions:	● User is registered and currently logged into the app. ● User's device has a stable internet connection. ● The app is functioning, and the server is available.		
Trigger:	User selects the 'Upload Reading Material' button in the app library.		
Main Path:	1. User navigates to library and selects 'Upload Reading Material' button. 2. User grants app permission to access local storage. 3. System prompt user to choose a valid file to upload, must have PDF/EPUB file extension. 4. User selects a file to upload. 5. System prompts user to confirm upload of selected file, user can’t cancel later. 6. System validates file format and size. 7. After validation, system triggers segment reading material functionality. 8. System notifies the user that the material is currently being segmented.		

	9. System uploads the segmented text to users library. 10. System notifies user of a successful upload. 11. Users can now trigger the View Reading Material feature
Postconditions	- The reading material is segmented and stored in the user's library. - The user can read their segmented material.
Alternative Path:	6. Invalid File Error: 6a. System detects an invalid file format or size during upload. 6b. System prompts user of invalid file error and allows the user to re-select a file. .

7.2 Segment Reading Material

Use Case Name:	Segment Reading Material	UniqueID:	UC002
Description:	The system segments the uploaded material into chunks. All text segmentation is processed on a server to improve performance. The system collects information needed for accurate segments. The server then segments the material using the data provided by the system. Servers sends the segmented text back to the system.		
Actors:	System, Server		
Preconditions:	• User is registered and logged into the application. • The app is functioning, and device has a stable internet connection • Server is operational and can receive requests. • User could have initiated the 'Upload Reading Material' feature with a valid file. • User could have initiated the ‘Adapt Reading Segments’ feature.		
Trigger:	The system automatically initiates the segmentation process once valid reading material is provided by the user or the system.		
Main Path:	1. System initially analyses the uploaded material to provide vital data to the server. 2. System gathers internal information(e.g., streaks, reading speed) to determine accurate segmentation parameters. 3. System notifies server that material has been uploaded. 4. The system sends a segmentation request to the server, including the file and other data. 5. The server processes the file, segments the material based on the provided parameters, and ensures coherency (e.g., paragraph integrity) 6. Server prompts system of successful segmentation.		

	7. Server sends the segmented content back to the system. 8. The system receives the segmented text, performs validation checks, and integrates back into 'Upload Reading Material' path. 9. Server removes any temporary files to improve security and increase storage.
Postconditions	- The uploaded text is successfully processed and segmented. - The upload reading materials workflow can continue.
Alternative Flows:	4a Network Issues 1. Users connectivity issues prevent file transfer to the server. 2. System attempts to re-upload the material automatically (two attempts). 3. If after the specified attempts the upload fails, the system prompts the user to try again later. 4. Use case ends, system redirects user back to the prior process. 5a Segmentation Failure 1. Server encounters an error during segmentation process. 2. Server alerts the system of this error. 3. System informs the user of this failure with a prompt. 4. Use case ends, system redirects user back to the prior process.

7.3 Adapt Reading Segments

Use Case Name:	Adapt Reading Segments	UniqueID:	UC003
Description:	Utilizes an algorithm that dynamically adapts the difficulty of reading segments using various user metrics. Maintains app engagement by adapting users segments to align with their metrics and preferences.		
Actors:	User, System		
Preconditions:	• User is registered and logged into the application. • User has been using the ‘Complete Reading Segment’ feature daily. • The relevant reading metrics and user preferences are available in the system.		
Trigger:	User completed a daily reading segment. The system prompts user to adapt current reading material. Users is away from the app. A notification prompts user to adapt current reading material.		
Main Path:	1. System collects user's reading metrics after completing a reading segment. 2. The system gathers metrics including current streak, average streak lifecycle, reading speed, and user preferences. 3. The system provides the adaption algorithm with current metrics. 4. The system utilizes this algorithm to calculate the current difficulty level based on these metrics. 5. The algorithm determines an adjustment is needed, and the system prompts the user to re-segment the material. 6. The user accepts segmenting the text to a new difficulty.		

	7. System triggers the segment reading material functionality, to recalibrate the text. 8. System prompts user of successful recalibration of reading material.
Postconditions	- The reading segments is adapted to match the user's reading abilities. - User can continue reading the new or existing material.
Alternative Flows:	5a No Adjustment required 1. System keeps current difficulty and doesn't prompt the user to adapt reading material. 6a User Declines Adaptation 1. User chooses not to adapt the reading segment difficulty. 2. System logs the decision and maintains current difficulty settings. 3. The algorithm adjusts difficulty level after user declines.

8. Brief Use Cases

8.1. View Reading Material Use Case

Use Case Name:	View Reading Material	UniqueID:	UC004
Primary Actors:	User		
Description:	The user selects a reading material uploaded to their library. The system loads the reading materials current text segment. The segmented text is presented in a readable format to the user. The segment size correlates to the preferences set by the users.		

8.2. Complete Reading Segment Use Case

Use Case Name:	Complete Reading Segment	UniqueID:	UC005
Primary Actors:	User		
Description:	After finishing a reading segment in their chosen material, the user triggers this use case. The user selects complete segment, updating relevant metrics such as reading stats and time spent reading. This action contributes to the user's reading streak. This use case is central to the app's core functionality, enhancing user engagement and tracking reading habits.		

8.3. Update Streak Segment Use Case

Use Case Name:	Update Streak	UniqueID:	UC006
Primary Actors:	User, System		
Description:	This use case is activated when a user completes a reading segment. The system tracks the user's consecutive days of reading and updates their current streak count. The streak update is visually represented in the user interface, accompanied by notifications.		

8.4. Register Account Use Case

Use Case Name:	Register Account	UniqueID:	UC007
Primary Actors:	User, Database		
Description:	This use case starts when a user decides to create a new account within the application. The user is presented with a registration form where they input necessary information, such as name, email address, and a chosen password. Upon submitting the form, the system validates the provided information. If all criteria are met, the system creates a new user account, storing the user's data on a database. Once registered, the user gains access to the app's full features.		

8.5. Login Use Case

Use Case Name:	Login	UniqueID:	UC008
Primary Actors:	User, Database		

Description:	This use case begins when a user attempts to access their account in the application. The user is prompted to enter their login credentials, username and password. The system then verifies the entered credentials against the stored user data. If the credentials are correct, the user is granted access to their account. In case of incorrect credentials, the system provides feedback, such as an error message, and may offer options like password recovery.

References

American Psychological Association. (2018, Aug 20). *Teens today spend more time on digital media, less time reading*. Retrieved from American Psychological Association: https://www.apa.org/news/press/releases/2018/08/teenagers-read-book

Watson, A. (2023, August 31). *Topic: Reading habits in the U.S.* Retrieved from Statista: https://www.statista.com/topics/3928/reading-habits-in-the-us/#topicOverview

Wikimedia Foundation. (2023, 05 March). *Functional specification*. Retrieved from Wikipedia: https://en.wikipedia.org/wiki/Functional_specification

Wikimedia Foundation. (2023, July 25). *Non-functional requirement*. Retrieved from Wikimedia: https://en.wikipedia.org/wiki/Non-functional_requirement