



Bilkent University
Department of Computer Engineering

CS 491- Senior Design Project

Project Name: Papyrus



Papyrus

Analysis Report

Group Members: Ant Duru, Atay Kaylar, Enver Yiğitler

Supervisor: Fazlı Can

Jury Members: Shervin Arashloo, Hamdi Dibeklioğlu

Innovation Expert: Murat Kalender

1. Introduction	6
2. Current Systems	7
2.1 Kindle	8
2.2 duoBooks	9
2.2 Parallel Books	9
3. Proposed System	10
3.1 Overview	10
3.2 Functional Requirements	10
3.3 Non-functional Requirements	12
3.3.1 Usability	12
3.3.2 Security	12
3.3.3 Cost	12
3.3.4 Reliability	12
3.3.5 Performance	12
3.3.6 Accessibility	13
3.3.7 Compatibility	13
3.4 Pseudo-requirements	13
3.5 System Models	14
3.5.1 Scenarios	14
3.5.1.1 Scenario 1	14
3.5.1.2 Scenario 2	14
3.5.1.3 Scenario 3	14
3.5.1.4 Scenario 4	15
3.5.1.5 Scenario 5	15
3.5.1.6 Scenario 6	15
3.5.1.7 Scenario 7	15
3.5.1.8 Scenario 8	16
3.5.1.9 Scenario 9	16
3.5.1.10 Scenario 10	16
3.5.1.11 Scenario 11	16
3.5.1.12 Scenario 12	17
3.5.1.13 Scenario 13	17
3.5.1.14 Scenario 14	17
3.5.1.15 Scenario 15	18
3.5.1.16 Scenario 16	18
3.5.1.17 Scenario 17	18
3.5.1.18 Scenario 18	19
3.5.1.19 Scenario 19	19
3.5.1.21 Scenario 21	20
3.5.1.22 Scenario 22	20
3.5.1.23 Scenario 23	20
3.5.1.24 Scenario 24	20

3.5.1.25 Scenario 25	21
3.5.1.26 Scenario 26	21
3.5.1.27 Scenario 27	21
3.5.1.28 Scenario 28	21
3.5.1.29 Scenario 29	22
3.5.1.30 Scenario 30	22
3.5.1.31 Scenario 31	22
3.5.1.32 Scenario 32	22
3.5.1.33 Scenario 33	23
3.5.1.34 Scenario 34	23
3.5.1.35 Scenario 35	23
3.5.1.36 Scenario 36	24
3.5.1.37 Scenario 37	24
3.5.1.38 Scenario 38	24
3.5.2 Use-case Model	25
3.5.2.1 Use-Case Diagram for Log In	25
3.5.2.2 Use-Case Diagram For Discover Page	26
3.5.2.3 Use-Case Diagram For FlashCard Decks	28
3.5.2.4 Use-Case Diagram for FlashCard Practice	29
3.5.2.5 Use-Case Diagram for in Book Actions	30
3.5.3 Object and Class Model	32
3.5.3.1 Book Class	33
3.5.3.2 Audiobook Class	33
3.5.3.3 AlignmentCollection Class	33
3.5.3.4 Navigator Class	33
3.5.3.5 TextNavigator Class	33
3.5.3.6 AudioNavigator Class	33
3.5.3.7 NarratedNavigator Class	33
3.5.3.8 MyBooks Class	34
3.5.3.9 Parser Class	34
3.5.3.10 User Class	34
3.5.3.11 Shelf Class	34
3.5.3.12 MyFlashCards Class	34
3.5.3.13 Deck Class	34
3.5.3.14 FlashCard Class	34
3.5.4 Dynamic Models	35
3.5.4.1 Sequence Diagram to Get the Aligned Text	35
3.5.4.2 State Diagram of FlashCard Practice	36
3.5.4.3 Activity Diagram of Text Navigation Hub	38
3.5.5 User Interface - Navigational Paths and Screen Mock-ups	39
3.5.5.1 Login/Sign Up Screen	39
3.5.5.2 Discover Panel	40
3.5.5.3 Opening a Collection in Discover	42

3.5.5.4 Search Screen in Discover	43
3.5.5.5 An Author's Page in Discover	44
3.5.5.6 FlashCards Main Screen	45
3.5.5.7 FlashCards Main Screen: Edit Decks	46
3.5.5.8 Edit Decks More Option Screen	47
3.5.5.9 FlashCard Practice	48
3.5.5.10 Adjust Front/Back of a FlashCard	49
3.5.5.11 Show All Cards Screen	50
3.5.5.12 Add New Card Screen	51
3.5.5.13 FlashCard Pop-up Settings	52
3.5.5.14 MyBooks Panel	53
3.5.5.15 MyBooks Sort and View Menu	54
3.5.5.16 My Books Shelf View	55
3.5.5.17 Navigator: Text Mode	56
3.5.5.18 Navigator: Word Selection Pop-ups	57
3.5.5.19 Navigator: Phrase Selection Pop-ups	58
3.5.5.20 Navigator: Pop-ups Settings	59
3.5.5.21 Navigator: Aligned Contents Settings	60
3.5.5.22 Navigator Creating Annotations and Highlights	61
3.5.5.23 Navigator Creating Bookmarks and Annotations	62
3.5.5.24 Navigator: Chapter Navigator and Font Settings	63
3.5.5.25 Navigator: Change Mode: Audiobook, Narrated Book, Text Book	64
3.5.5.26 Navigator: Audiobook Mode	65
Figure 35: Audiobook Mode Screen	65
3.5.5.27 Navigator: Narrated Mode	66
3.5.5.28 Account Panel	67
4. Other Analysis Elements	68
4.1 Consideration of Various Factors in Engineering Design	68
4.1.1 Public Health	68
4.1.2 Safety	68
4.1.3 Welfare	68
4.1.4 Global	68
4.1.5 Cultural	69
4.1.6 Social	69
4.1.7 Environmental	69
4.1.8 Economic	70
4.2 Risks and Alternatives	71
4.3 Project Plan	75
4.3.7 Gantt Chart	87
4.4 Ensuring Proper Teamwork	88
4.5 Ethics and Professional Responsibilities	88
4.6 Planning for New Knowledge and Learning Strategies	89

5. References**90**

1. Introduction

With globalization and the progression of technology, people are able to communicate with almost the whole world. People can reach limitless resources, meet new people and cultures, and are able to share their thoughts with anyone around the globe via the internet. This new experience caused people to be interested in other cultures, people, and languages. People began learning languages to communicate with a specific community that shares a common interest with them, read their favorite book in its language, and understand their favored foreign songs. Apart from hobbies, learning a new language helps people to advance in their careers. Studying or working abroad, studying a culture or literature of a foreign community, translating books or speeches require excellent knowledge of the respective language as well.

To learn a new language, several popular methods are used among people: Going to courses, watching videos, being exposed to the target language, etc. PAPYRUS offers an e-reading environment specifically designed for language learners. The ability to read in a foreign language is a significant part of learning a new language. Yet, fully understanding a novel written in a foreign language is not something easy to do. In PAPYRUS, original books are aligned, with sentence alignment algorithms, with the translated books sentence-by-sentence to make the learners feel as if they were reading these two books concurrently to provide an instant understanding of a passage or a sentence. Secondly, PAPYRUS will include narrated audiobooks to learn pronunciations accurately. Users will be able to

annotate on books, create book collections and share them with everyone to have a great social media experience.

Another critical requirement of learning a new language is to have a rich vocabulary. In PAPYRUS, users can create flashcards to practice new words with repetitive learning algorithms. A word can be added as a flashcard while reading the book, with its definition, example sentences, etc., taken from a dictionary chosen by the user.

PAPYRUS's goal is to help language learners by providing a social media-like environment, where they can read books in foreign languages while being able to see the translated versions of passages and sentences, practicing their vocabulary, sharing their notes and annotations with learners all around the whole world and have a great learning process.

2. Current Systems

Papyrus combines five main features to enhance reading experience for language learners: sentence alignment for polylingual texts, forced alignment for narration, flashcard system, paragraph & word selection pop-ups and lastly sharable annotations. Currently, there isn't any other system that combines all of these features at once. Besides we couldn't find any app which provides alignment services for users' own files. However, there are some reading applications which provide some of these functionality with some level. After some market research we

chose three applications which are the closests to the Papyrus. Following are these applications with comparisons.

2.1 Kindle

Amazon's Kindle app and devices provide a reading platform with a rich library store and a lot of features. Table below compares Kindle with Papyrus according to main features that Papyrus provides.

Features	Kindle
Sentence Alignment / Forced Alignment	Not provided
Narrated Text	Provided (Book and audiobook should be bought with the extra Audible Narration cost.)
Paragraph and Word Selection Pop-ups	Kindle devices provide custom selectable dictionaries and web pop-ups. Kindle mobile app provides dictionary, machine translation and wikipedia search pop-ups.
Flashcard System	Very primal flashcard system. Flashcards only created with two custom text notes to each side without any help from dictionaries and web.

Table 1: Features of Kindle

2.2 duoBooks

duoBooks is an iOS e-reading app. Its library has a very limited number of (around 20) bilingual books with a very limited number of language choices (English, Russian, Spanish).

Features	duoBooks
Sentence Alignment / Forced Alignment	Library includes some bilingual books.
Narrated Text	Not provided
Paragraph and Word Selection Pop-ups	Word selection pop-ups one word translation without definition or dictionary.
Flashcard System	Very primal flashcard system. It only saves words with one word translations.

Table 2: Features of duoBooks

2.2 Parallel Books

Parallel Books is an iOS e-reading app. Again, its library has a very limited number of bilingual books with a very limited number of language choices (English, Russian, Spanish).

Features	Parallel Books
Sentence Alignment / Forced Alignment	Very limited number of multilingual books.
Narrated Text	Not provided
Paragraph and Word Selection Pop-ups	Apple's own dictionary provided.
Flashcard System	Not provided

Table 3: Features of Parallel Books

3. Proposed System

3.1 Overview

Each user will have a library to themselves containing books, collections, audiobooks, and dictionaries. Information about books, collections, users, flashcards, etc., will be stored in a database. PAPYRUS will be an iOS application. Swift programming language will be used for implementation.

3.2 Functional Requirements

- The application will require a signing up process for public users.
- The application will require a signing up process for publishers.
- Users will be able to sign up via Google, Facebook, or Apple.
- Users will be able to change their passwords.
- Publishers will be able to publish new books.
- Users will be able to see the books that they saved.
- Users will be able to open a book.
- Users will be able to sort their books alphabetically or by their language.
- Users will be able to search for books, authors, genres, collections.
- Users will be able to create book collections.
- Users will be able to add books to their book collections.
- Users will be able to sort books in a collection alphabetically or by their language.
- Users will be able to see shared annotations for a book.
- Users will be able to share their annotations for a book.

- Users will be able to see flashcard decks.
- Users will be able to add or remove flashcard decks.
- Users will be able to edit flashcard decks by changing their name, adding or removing cards to/from a specific flashcard deck.
- Users will be able to sort flashcard decks.
- Users will be able to search for shared flashcard decks.
- Users will be able to share their flashcard decks with other users.
- Users will be able to practice with their flashcard decks.
- Users will be able to declare their knowledge about a specific flashcard.
- Users will be able to edit a flashcard.
- Users will be able to see their progress for a flashcard deck.
- Users will be able to change the view of their library.
- Users will be able to open a dictionary.
- Users will be able to see detailed information about a book, author, collection, or genre.
- Users will be able to switch between menus.
- Users will be able to see the translated version of a passage or a sentence while reading the book.
- Users will be able to make a flashcard out of a word while reading a book.
- Users will be able to select definitions, example sentences from the dictionary they choose for a flashcard.
- Users will be able to add fields to both the front and back of a flashcard.
- Users will be able to end flashcard practice at any moment after starting.

3.3 Non-functional Requirements

3.3.1 Usability

- The application can be used on iOS platforms.
- The application will provide an easy-to-use interface.

3.3.2 Security

- User information will not be shared with third parties.
- Publisher information will not be shared with third parties.

3.3.3 Cost

- Users will be charged for subscription to the app.
- Publishers will be paid for adding their books to the app with respect to reading time of their book contents.

3.3.4 Reliability

- The storage in the database should be reliable.
- The application will be stable.

3.3.5 Performance

- The app should switch between menus fast.
- Adding a flashcard should happen instantly.

- Changing pages should happen very fast.
- Aligned sentences should be displayed instantly.
- The Login process should happen very fast.

3.3.6 Accessibility

- The application should be downloadable for free.
- The application should be downloadable on AppStore.

3.3.7 Compatibility

- Since we use Swift language and SwiftUI, the system should run in iOS 7 or a higher version.

3.4 Pseudo-requirements

- The application will operate on iOS platforms.
- The application will be in English.
- Git and GitHub will be used for version control processes.
- Swift language will be used to implement the application.

3.5 System Models

3.5.1 Scenarios

3.5.1.1 Scenario 1

Use Case: Open Book

Primary Actor: User

Entry Condition: User is on the My Books panel.

Exit Condition: User returns to My Books.

Main Flow of Events:

1. User taps on a Book card in My Books.
2. App segues to Text Navigation Hub.

3.5.1.2 Scenario 2

Use Case: Open Chapter Navigation

Primary Actor: User

Entry Condition: User is on the Text Navigation Hub.

Exit Condition: User returns to the Text Navigation Hub.

Main Flow of Events:

1. User taps on the Chapter Navigation button.
2. App segues to Text Navigation Hub.

3.5.1.3 Scenario 3

Use Case: Search in Document

Primary Actor: User

Entry Condition: User is on the Text Navigation Hub.

Exit Condition: User returns to the Text Navigation Hub.

Main Flow of Events:

1. User taps on the search in the document.
2. App shows the searched items in a new page.

3.5.1.4 Scenario 4

Use Case: Open annotations and highlights from the Text Navigation Hub.

Primary Actor: User

Entry Condition: User is on the Text Navigation Hub.

Exit Condition: User returns to the Text Navigation Hub.

Main Flow of Events:

1. User taps on the annotations icon on the text navigation hub.
2. App opens the annotations and highlights page.

3.5.1.5 Scenario 5

Use Case: Open annotations and highlights from the My Books Page.

Primary Actor: User

Entry Condition: User is on the My Books page.

Exit Condition: User returns to My Books.

Main Flow of Events:

1. User taps the corresponding three dots in the My Books page for the desired book.
2. App opens a pop up screen which shows the operations for the book.
3. User selects the annotations from the pop up page.
4. App shows annotations page.

3.5.1.6 Scenario 6

Use Case: Change mode of book reader to narration.

Primary Actor: User

Entry Condition: User is on the Text Navigation Hub.

Exit Condition: User changes the mode to a different mode.

Main Flow of Events:

1. User taps mode selection wheel on the Text Navigation Hub.
2. Mode selection wheel is showing the modes for the hub.
3. User selects the narration mode.
4. Mode is changed to narration mode.

3.5.1.7 Scenario 7

Use Case: Change mode of book reader to AudioBook.

Primary Actor: User

Entry Condition: User is on the Text Navigation Hub.

Exit Condition: User changes the mode to a different mode.

Main Flow of Events:

1. User taps mode selection wheel on the Text Navigation Hub.
2. Mode selection wheel is showing the modes for the hub.
3. User selects the AudioBook mode.
4. Mode is changed to AudioBook mode.

3.5.1.8 Scenario 8

Use Case: Open alignment settings in the Text Navigation hub.

Primary Actor: User

Entry Condition: User is on the Text Navigation Hub.

Exit Condition: User returns to the Text Navigation Hub.

Main Flow of Events:

1. User taps the alignment settings icon in the Text Navigation Hub.
2. Alignment settings page is opened.

3.5.1.9 Scenario 9

Use Case: Open book's flashcard.

Primary Actor: User

Entry Condition: User is on the Text Navigation Hub.

Exit Condition: User returns to the Text Navigation Hub.

Main Flow of Events:

1. User taps the flashcard icon in the Text Navigation Hub.
2. Flashcard is opened.

3.5.1.10 Scenario 10

Use Case: Open book's flashcard.

Primary Actor: User

Entry Condition: User is on the Text Navigation Hub.

Exit Condition: User returns to the Text Navigation Hub.

Main Flow of Events:

1. User taps the flashcard icon in the Text Navigation Hub.
2. Flashcard is opened.

3.5.1.11 Scenario 11

Use Case: Open book's flashcard.

Primary Actor: User

Entry Condition: User is on the Text Navigation Hub.

Exit Condition: User returns to the Text Navigation Hub.

Main Flow of Events:

1. User taps the flashcard icon in the Text Navigation Hub.
2. Flashcard is opened.

3.5.1.12 Scenario 12

Use Case: Get book info

Primary Actor: User

Entry Condition: User is on the Text Navigation Hub.

Exit Condition: User returns to the Text Navigation Hub.

Main Flow of Events:

1. User taps the info icon in the Text Navigation Hub.
2. Flashcard is opened.

3.5.1.13 Scenario 13

Use Case: Go to a selected page in the Text Navigation Hub

Primary Actor: User

Entry Condition: User is on the Text Navigation Hub.

Exit Condition: User goes to the desired page number

Main Flow of Events:

1. User taps the “go to page” icon in the Text Navigation Hub.
2. Pop up screen opens in the Text Navigation Hub asking the desired page.
3. User enters the desired page number.
4. Desired page number is opened.

3.5.1.14 Scenario 14

Use Case: Open flashcard decks for the book

Primary Actor: User

Entry Condition: User is on the Text Navigation Hub.

Exit Condition: User goes to the desired page number

Main Flow of Events:

1. User taps the flashcard icon in the Text Navigation Hub.
2. “Show book’s deck” page is opened, where flashcards of the book are shown.

3.5.1.15 Scenario 15

Use Case: User responds to flashcard

Primary Actor: User

Entry Condition: User is on flashcard study view

Exit Condition: New flashcard is viewed

Main Flow of Events:

1. User selects a response ("fail", "good", "not sure") for the flashcard.
2. New flashcards are shown.

3.5.1.16 Scenario 16

Use Case: User edits a flashcard

Primary Actor: User

Entry Condition: User is on flashcard study view

Exit Condition: User returns to flashcard study view

Main Flow of Events:

1. User selects the desired sentence or word in the flashcard by tapping.
2. User is able to edit the sentence or word.
3. User edits the desired sentence or word.
4. User ends the edit.

3.5.1.17 Scenario 17

Use Case: User adjusts the back and front of the card.

Primary Actor: User

Entry Condition: User is on flashcard study view.

Exit Condition: User returns to flashcard study view.

Main Flow of Events:

1. User taps the edit icon on the flashcard.
2. Pop-up screen opens, which shows the options of front and back.
3. User selects front or back.
4. Edit screen is shown, where flashcard's items are shown.
5. User activates or deactivates the items by tapping on them.
6. User saves the changes.

3.5.1.18 Scenario 18

Use Case: User searches for an element

Primary Actor: User

Entry Condition: Discover page

Exit Condition: User is taken to the desired element, or user changes panel

Main Flow of Events:

1. User is on the discover panel.
2. User types the desired element.
3. The database searches for the element.
4. It is returned, and the user finds the element.

3.5.1.19 Scenario 19

Use Case: User narrows the search results.

Primary Actor: User

Entry Condition: Discover page

Exit Condition: The search results are narrowed, or user changes panel

Main Flow of Events:

1. User is on the discover panel.
2. User searches for an element(the previous use case)
3. User is not satisfied with the results, and likes to narrow the results.
4. User selects a particular element(author, book, audioBook, etc.)
5. Results are narrowed.

3.5.1.20 Scenario 20

Use Case: Open element

Primary Actor: User

Entry Condition: User is on discover panel

Exit Condition: Desired elements page is opened

Main Flow of Events:

1. User finds the desired element and selects it.
2. It is opened.

3.5.1.21 Scenario 21

Use Case: Sort Collection

Primary Actor: User

Entry Condition: User is on collection page

Exit Condition: User navigates to a different page or the collection is sorted.

Main Flow of Events:

1. User opens the sort pop-up page.
2. User selects the desired sort.
3. Collection is sorted.

3.5.1.22 Scenario 22

Use Case: Search in collection

Primary Actor: User

Entry Condition: User is on collection page

Exit Condition: Searched item is returned or user navigates to a different page.

Main Flow of Events:

1. User types the desired element into the search bar.
2. Elements are returned from the search engine to the page.

3.5.1.23 Scenario 23

Use Case: Add book to collection

Primary Actor: User

Entry Condition: User is on collection page

Exit Condition: User navigates to a different page or the adding operation is finished.

Main Flow of Events:

1. User is on the collection page.
2. User selects a book with opening add to shelf pop-up page.
3. Selects the desired collection.
4. Book is added to the collection.

3.5.1.24 Scenario 24

Use Case: See annotation for a book

Primary Actor: User

Entry Condition: User is on collection page

Exit Condition: User navigates to a different page or the annotation page is opened.

Main Flow of Events:

1. User desires to see the annotation of a book.
2. User selects the desired sort.
3. Annotations are opened.

3.5.1.25 Scenario 25

Use Case: Add book to book

Primary Actor: User

Entry Condition: User is on collection page

Exit Condition: Book is added to my books.

Main Flow of Events:

1. User selects a book and selects “add to my books” for that book.
2. Book is added to “my book”.

3.5.1.26 Scenario 26

Use Case: Open a deck

Primary Actor: User

Entry Condition: User is on flashcard view

Exit Condition: Deck is opened

Main Flow of Events:

1. User selects a deck and taps on the icon.
2. Deck is opened.

3.5.1.27 Scenario 27

Use Case: Add card to deck

Primary Actor: User

Entry Condition: User is on flashcard view's deck view

Exit Condition: A card is added to a deck

Main Flow of Events:

1. User taps on the add a new card icon.
2. Enters the name.
3. Card is created.
4. Card creation page is opened.

3.5.1.28 Scenario 28

Use Case: Add information about the card

Primary Actor: User

Entry Condition: User is on card view

Exit Condition: Information is added

Main Flow of Events:

1. User selects the desired parts of the card to edit.
2. User edits the parts.
3. User saves the card for future usage.

3.5.1.29 Scenario 29

Use Case: Select fields from dictionary

Primary Actor: User

Entry Condition: User is on card view

Exit Condition: Fields are selected from dictionary

Main Flow of Events:

1. User selects fields from a number of dictionaries when adding a card.
2. Fields are added.

3.5.1.30 Scenario 30

Use Case: Add a new deck

Primary Actor: User

Entry Condition: User is on flashcard view.

Exit Condition: A deck is created.

Main Flow of Events:

1. User taps on add a deck in deck view.
2. “Enter a name” page opens.
3. User enters a name, and taps to create.
4. Deck is created.

3.5.1.31 Scenario 31

Use Case: Remove a deck

Primary Actor: User

Entry Condition: User is on flashcard view.

Exit Condition: A deck is removed.

Main Flow of Events:

1. User taps on the three dots on the icon of the deck.
2. Pop-up page is opened.
3. User selects “remove” option.
4. Deck is removed.

3.5.1.32 Scenario 32

Use Case: Change deck name

Primary Actor: User

Entry Condition: User is on flashcard view.

Exit Condition: Name of the deck is changed.

Main Flow of Events:

1. User taps on the three dots on the icon of the deck.
2. Pop-up page is opened.
3. User selects “change name”.
4. User enters a name.
5. Name is changed.

3.5.1.33 Scenario 33

Use Case: See progress

Primary Actor: User

Entry Condition: User is on flashcard view.

Exit Condition: Progress is shown

Main Flow of Events:

1. User taps on the three dots on the icon of the deck.
2. Pop-up page is opened.
3. User selects “see progress”.
4. Progress is shown.

3.5.1.34 Scenario 34

Use Case: Share progress

Primary Actor: User

Entry Condition: User is on flashcard view.

Exit Condition: Progress is made public.

Main Flow of Events:

1. User taps on the three dots on the icon of the deck.
2. Pop-up page is opened.
3. User selects “make public”.
4. Deck is shared.

3.5.1.35 Scenario 35

Use Case: Create account

Primary Actor: User

Entry Condition: User is on “create an account screen”.

Exit Condition: User is on login screen.

Main Flow of Events:

1. User taps on the create account button.
2. “Create an account screen” is opened.
3. User enters mail and password.
4. Account is created.

3.5.1.36 Scenario 36

Use Case: Login

Primary Actor: User

Entry Condition: User is on login screen.

Exit Condition: Main application is opened.

Main Flow of Events:

1. After creating an account, the user must verify it via email.
2. User verifies the account.
3. User enters the credentials in the login screen.

4. User logs in. After first verification no other verification is required.

3.5.1.37 Scenario 37

Use Case: Login with Google, Apple or Facebook

Primary Actor: User

Entry Condition: User is on login screen.

Exit Condition: Main application is opened.

Main Flow of Events:

1. User taps on one of the “Login with Google, Apple or Facebook” buttons.
2. Credentials are automatically logged in via Google, Apple or Facebook services.
3. User verifies the account.
4. User logs in. After first verification no other verification is required.

3.5.1.38 Scenario 38

Use Case: Login

Primary Actor: User

Entry Condition: User is on the “user settings” page.

Exit Condition: Password is changed.

Main Flow of Events:

1. User selects “change password”.
2. User is asked to select a new password.
3. User types in a new password twice.
4. User’s password is changed

3.5.2 Use-case Model

3.5.2.1 Use-Case Diagram for Log In

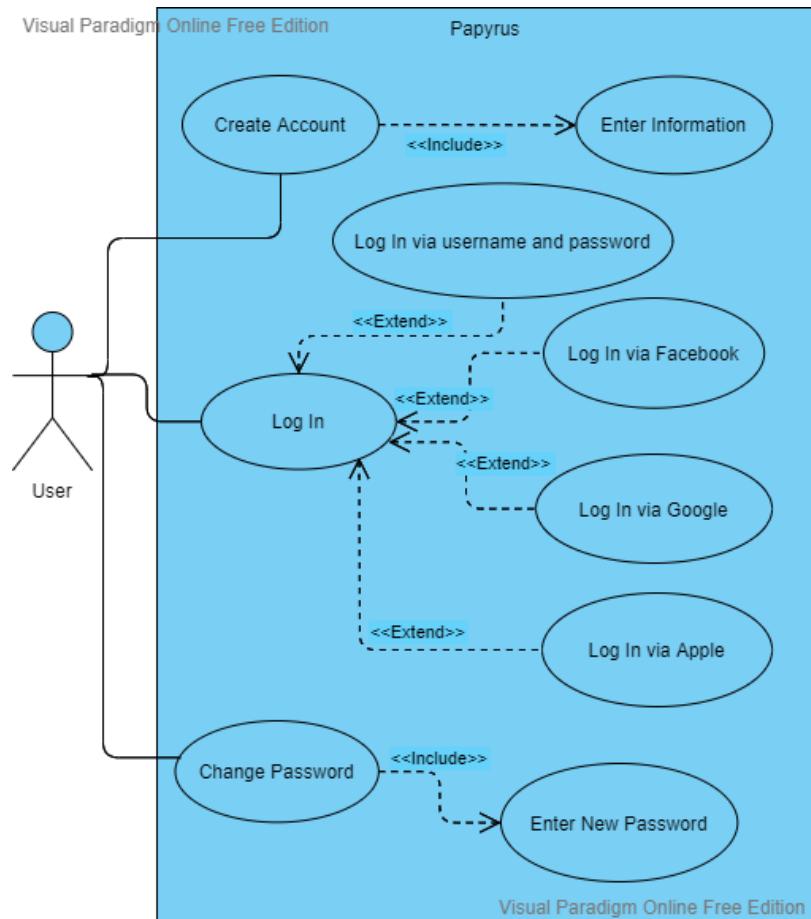


Figure 1: Use-Case Diagram for Log In

In the login screen, the user should be able to enter the application. To have an account, the user must create an account by entering information such as username and password. Or, the user can log in via their Facebook, Google or Apple account. If the user has a previously created account, the user can log in by their username and password. In case the user forgets their password or wants to change it, they can change their password.

3.5.2.2 Use-Case Diagram For Discover Page

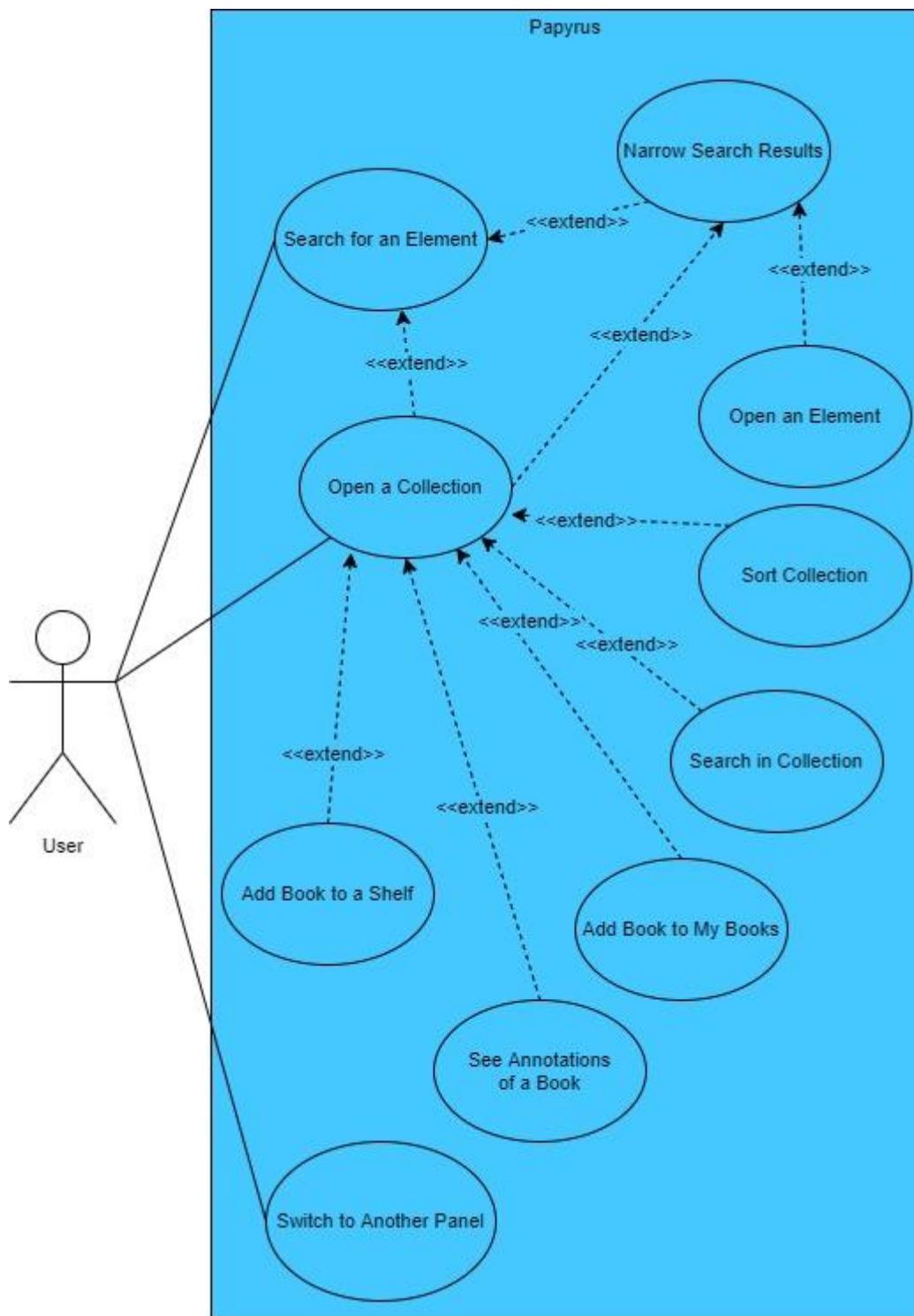


Figure 2: Use-Case Diagram for Discover

After the user logs in, the main panel will be the “Discover” panel. In the Discover panel, there will be book collections, the user can open a collection to see the books inside the collection. Also, in this panel, the user can search for an element in the

whole database. The elements are books, collections, authors, etc. The user can narrow the search results by specifying the type of the element. For instance, after typing “William Shakespeare”, the user will see the most relevant results about Shakespeare. Yet, if they narrow the results to be a collection, the user will not see the author page of Shakespeare, they will only see the collections about Shakespeare. The user can open an element, as well as a collection, after narrowing the results, or just after searching an element. If the user opens a collection, they can sort the collection, add any book in the collection to one of their collections, add any book to their shelf, or see the annotations on a specific book. Of course, the user can switch to other panels from the “Discover” panel.

3.5.2.3 Use-Case Diagram For FlashCard Decks

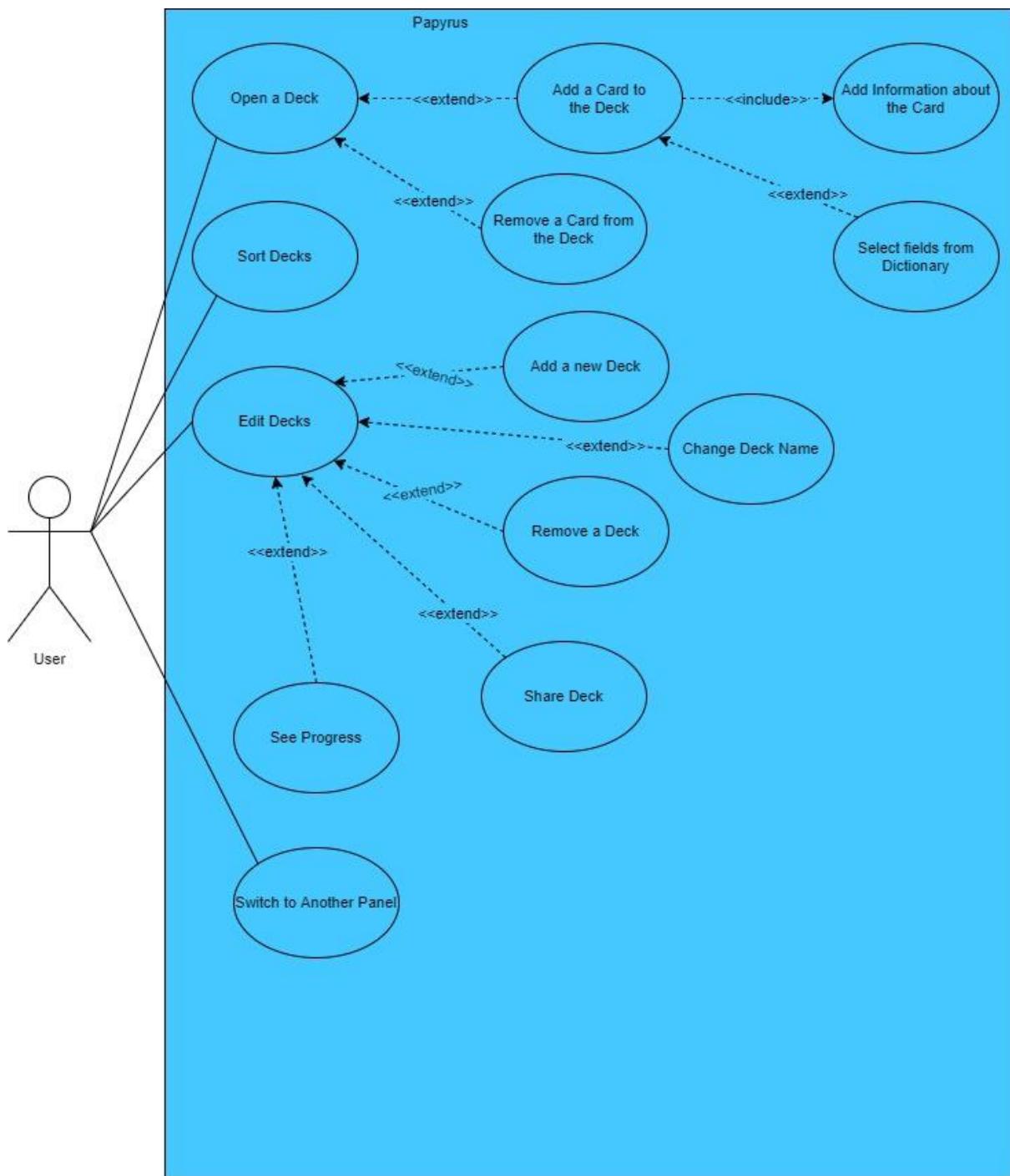


Figure 3: Use-Case Diagram for Decks

Flashcards are saved into the decks. Each user has a panel to see their decks. In this panel, the user can open a deck. The user can add/remove cards to/from this deck. To add a card, the user must enter the information about the card. This information's content is up to the user. The user can select fields from the dictionary

such as the translation, definition, example sentences, synonyms, etc. to add them to the card. The user also sorts the decks. To edit decks on a broader side, users can create a new deck, remove a deck, change the name of a deck, share their deck to the public, and they can also see their progress, meaning that their previous practices on a deck. The user can switch to other panels.

3.5.2.4 Use-Case Diagram for FlashCard Practice

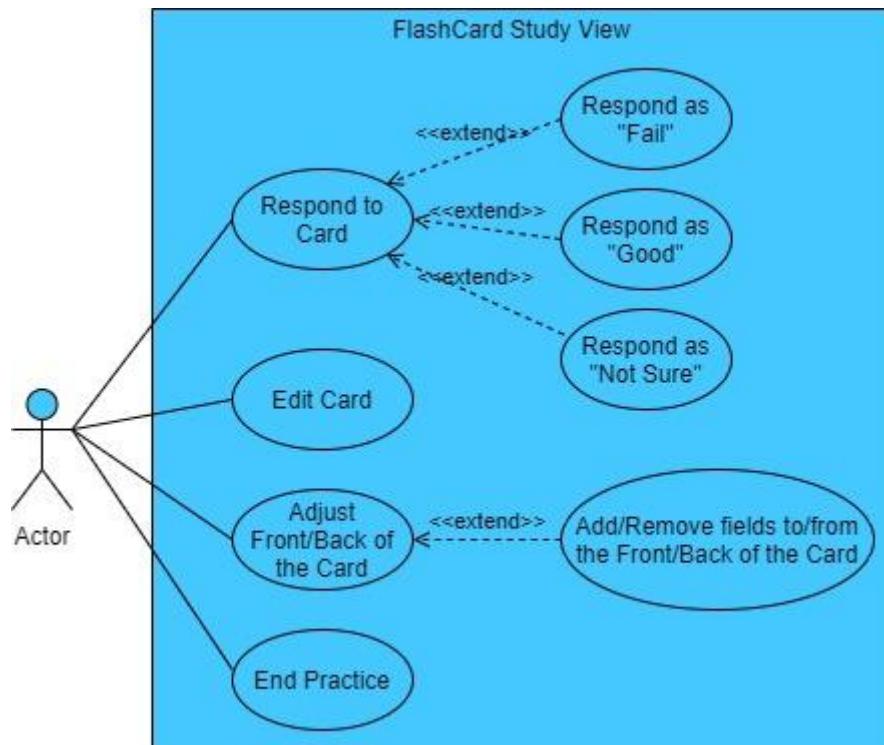


Figure 4: Use-Case Diagram for FlashCard Study

Users can practice (study) flashcards. When a flashcard is displayed at the screen, user can either respond as fail, meaning that they do not know the answer, or good, meaning that they know the answer or they can imply that they are not sure. These answers will be fed into our repetitive learning algorithm to provide a better learning

process to the user. User can adjust a card while training by adding or removing fields. User can end the practice any time they want.

3.5.2.5 Use-Case Diagram for In Book Actions

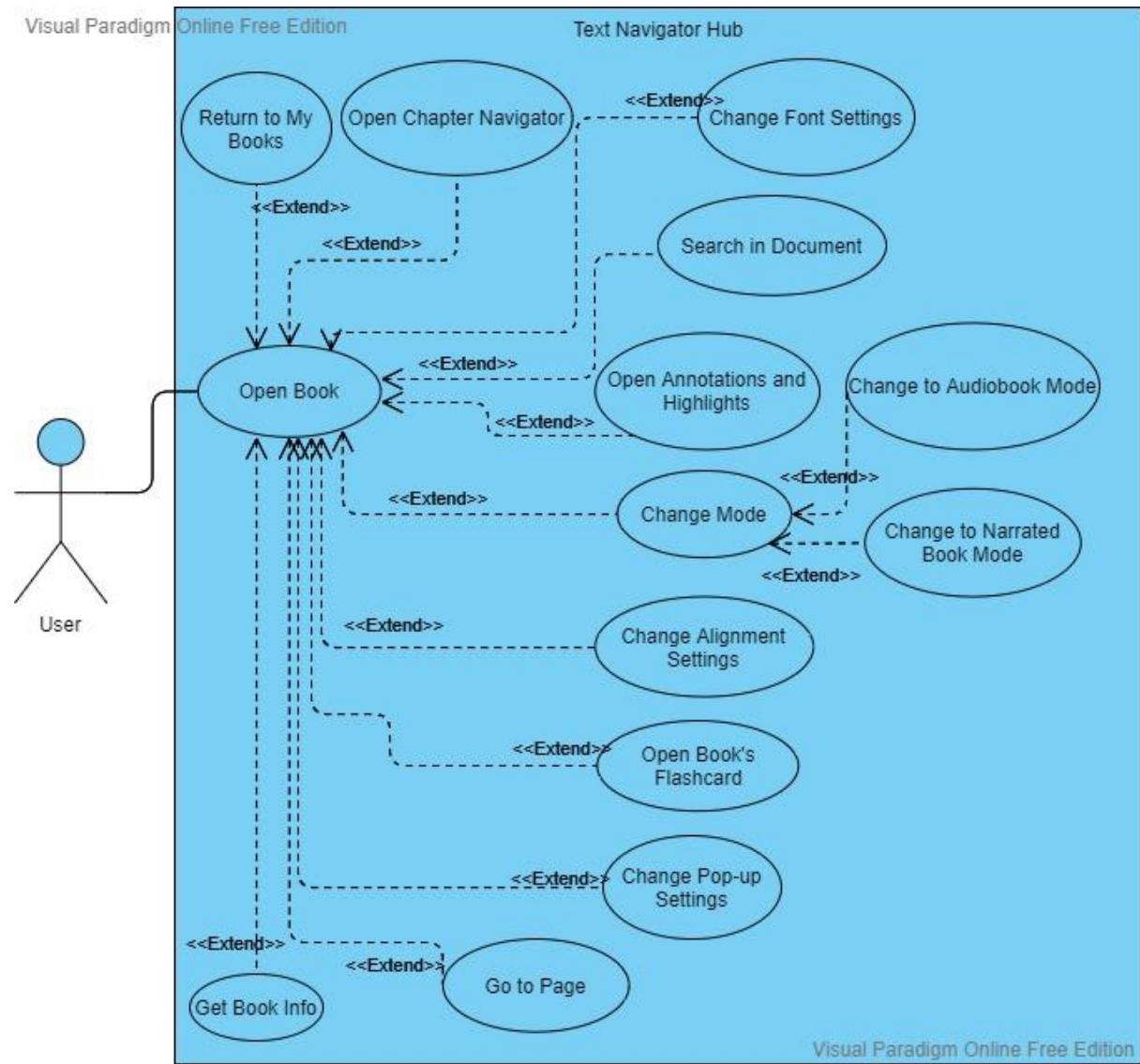


Figure 5: Use-Case Diagram for In Book Actions

User can open books. While reading a book, user can open chapter navigator to see the chapters of the book to just see them or to move between them. User can change the font settings of the book. User can search for a specific word while reading the book. User can open the annotations and highlights they have

downloaded or created. User can change pop-up settings- pop-ups occur when user selects a word or sentence- to have a better experience. User can go to a specific page. User can get the information about the book such as the author, the release date, the biography of the book, etc. User can open the deck of the book if it is created to see the flashcards. User can change the alignment settings of the book and the translated version. User can change the mode of the book, meaning that user can change between the textbook to audiobook or narrated book. User can return to their books by closing the book.

3.5.3 Object and Class Model

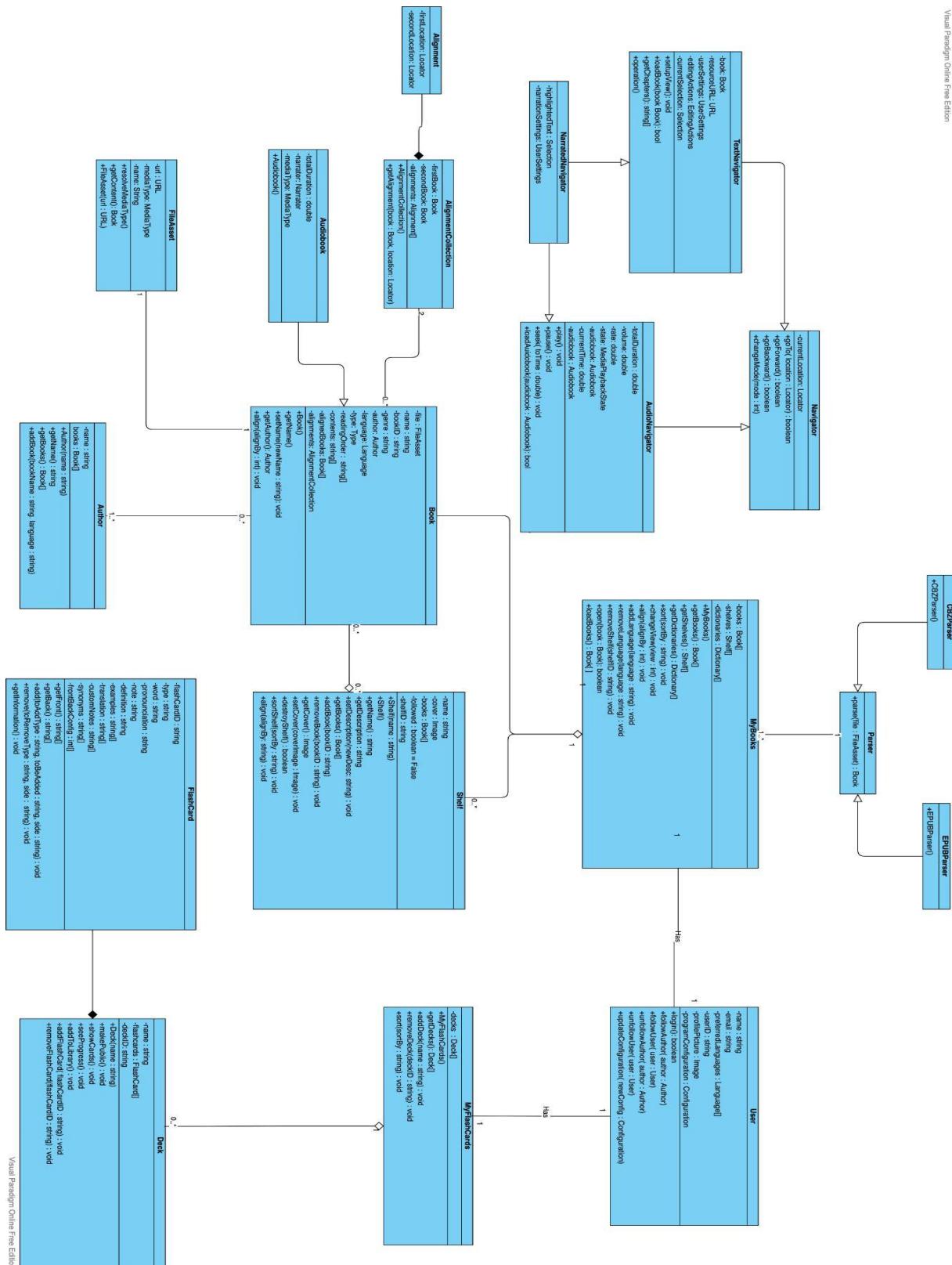


Figure 6: Class Diagram

3.5.3.1 Book Class

Book class represents books and is inherited by the Audiobook class. Book objects are created by parser classes which parses the fileAsset objects.

3.5.3.2 Audiobook Class

Audiobook class represents audiobooks. It inherits from the Book class.

3.5.3.3 AlignmentCollection Class

AlignmentCollection class stores the Alignment objects and some meta informations about these alignments.

3.5.3.4 Navigator Class

Navigator is the abstract base class of all navigators. It includes common attributes and operations for navigation of book contents.

3.5.3.5 TextNavigator Class

TextNavigator is responsible for viewing text content and navigating the books.

3.5.3.6 AudioNavigator Class

AudioNavigator is responsible for playing and navigating audiobooks.

3.5.3.7 NarratedNavigator Class

NarratedNavigator is for narration mode of the texts and audiobooks. It inherits from both of these classes because of its dual functionality.

3.5.3.8 MyBooks Class

MyBooks class represents one of the four navigation panels. It includes books, dictionaries and shelves of the user. It loads the book files and parses them with parsers.

3.5.3.9 Parser Class

Parser Class is only used to parse the files in specific format to create book objects from them.

3.5.3.10 User Class

User class represents the user. It includes his\her necessary information and some basic functionalities like login, following another user, etc.

3.5.3.11 Shelf Class

Shelf objects are collections of book objects.

3.5.3.12 MyFlashCards Class

MyFlashCards class represents one of four main navigation panels of the app. It includes the flashcard decks of the user. It is responsible for viewing and editing flashcards.

3.5.3.13 Deck Class

Deck class is the collection class of flashcard objects.

3.5.3.14 FlashCard Class

FlashCard Class represents flashcards created by the user.

3.5.4 Dynamic Models

3.5.4.1 Sequence Diagram to Get the Aligned Text

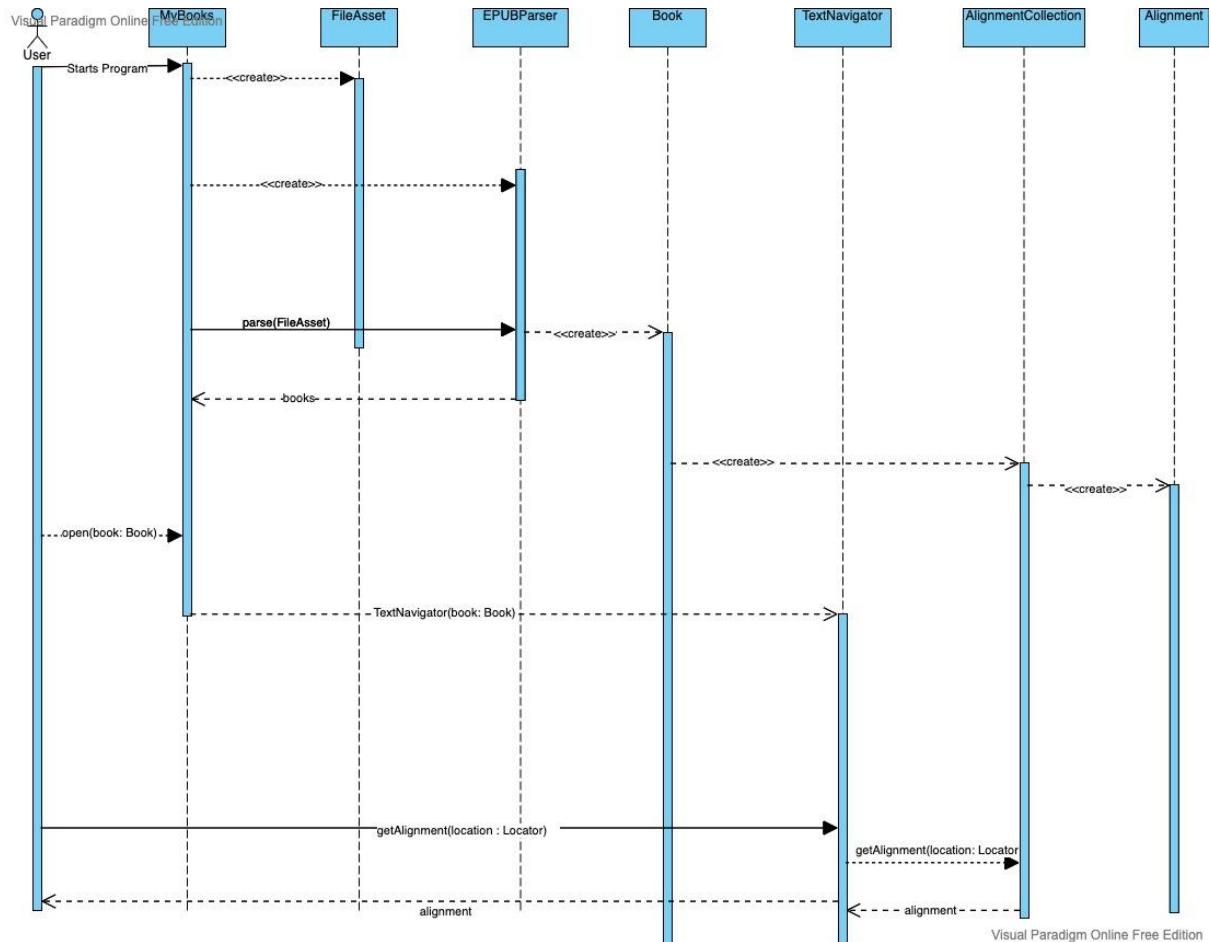


Figure 7: Sequence Diagram to get the aligned text

At the program initialization a MyBooks object is created which searches the database and creates FileAsset objects from the files. FileAsset objects are parsed with the parser with respect to their file type (We will initially only support EPUB format). Parses will produce Book (or Audiobook) objects. The navigator object views the book object. Then, the user selects a phrase to see its aligned text. This

messages the navigator object to fetch the aligned text from AlignedCollection.

Finally, the navigator will show the aligned text to the user with a pop-up view.

3.5.4.2 State Diagram of FlashCard Practice

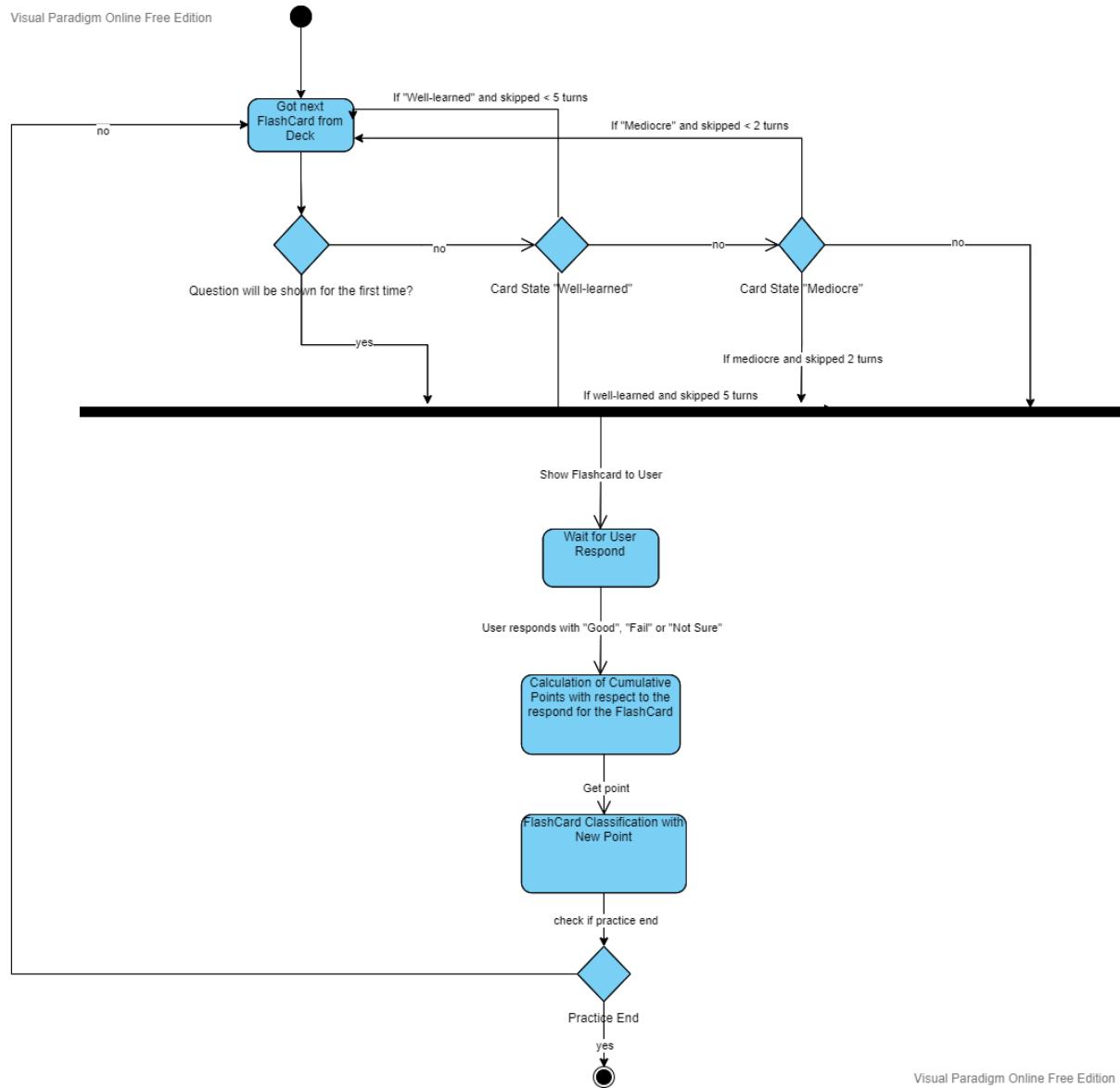


Figure 8: State Diagram for FlashCard Practice

This state diagram shows the states of the system in a flashcard practice. The system shows flashcards to the user one-by-one to get their answer. Therefore, firstly, the system will be in the “Got FlashCard” state. After that, the system will evaluate the situation of the card. A card may be asked for the first time. If so, the

system will continue. If not, the system checks the understanding of the user of that flashcard. This process is done by scoring the user based on their answers to the cards. The system gets feedback from previous attempts to score the understanding of the user of that card. If the state of the user on that card is “Well-learned”, the system will not show that card to the user for five turns. In the fifth turn, the system will continue showing that card. If a card is not “Well-learned”, the system will check whether the state is “Mediocre” and does the same process for three turns. If the state is not “Mediocre” the card will be shown. After this deciding process, the flashcard will be shown to the user and the system goes into “Wait for User Respond” state. When the user responds to the card, the system will go into a calculation state to get the cumulative score of that card. When the calculation state ends, the system will go into classifying the card as “Well-learned”, “Mediocre” or “Fail”. By that, one card’s process will end. The system will check if the practice ends, meaning that there is no other card that will be shown. If the system decides that the practice ends, the practice will end. If not, the system will get the next card from the deck.

3.5.4.3 Activity Diagram of Text Navigation Hub

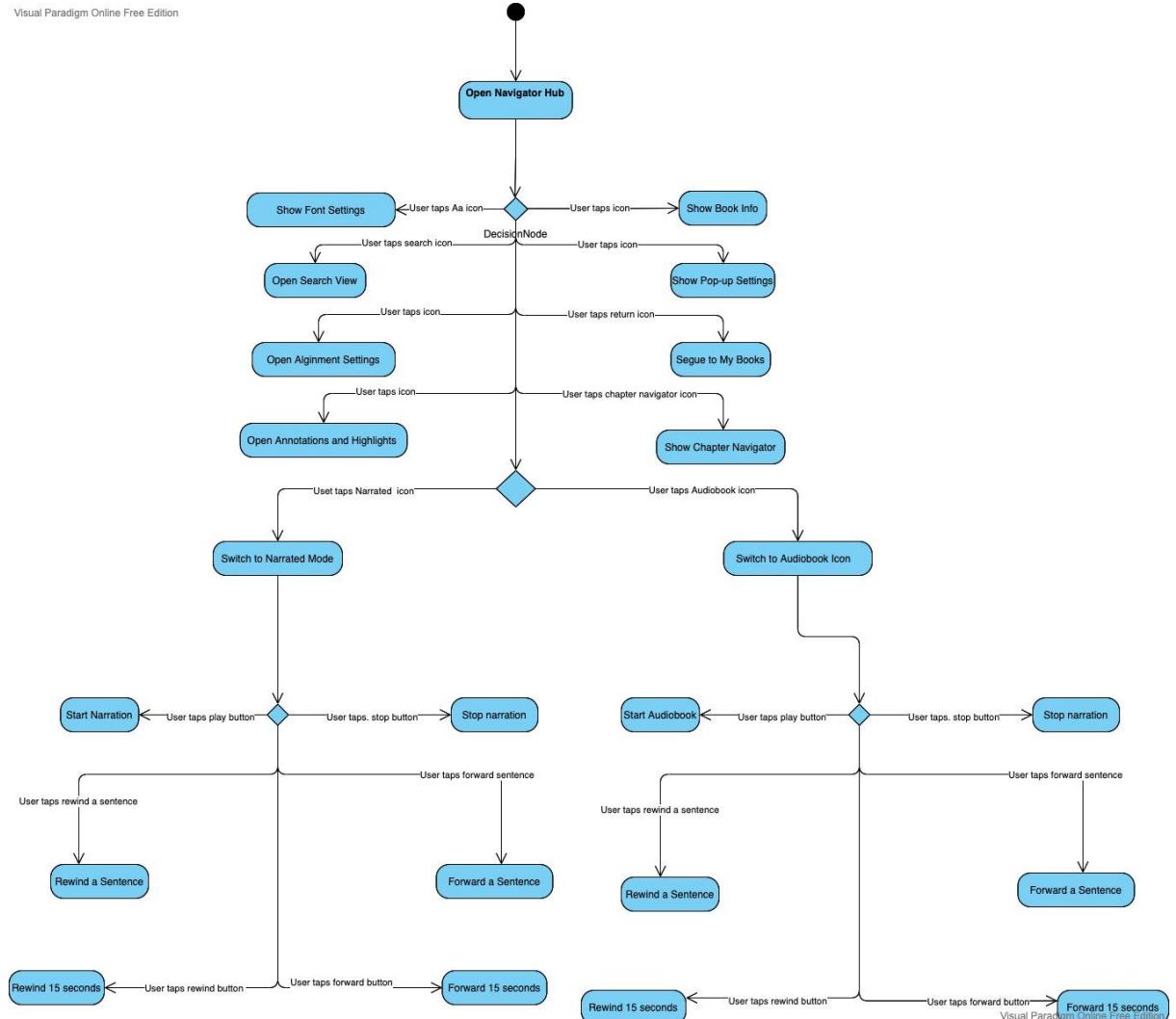


Figure 9: Activity Diagram of Text Navigation Hub

This activity diagram models the main activities that the user can do with navigator (book or audiobook). After opening a book or audiobook from the MyBook panel, the user can initiate many activities. Some of those activities are: Open alignment settings: with which the user can change aligned texts and audiobooks, open annotations and highlights, open chapter navigator, return to MyBooks panel, open pop-up settings in where the user can change pop-up order and contents, open general book information page where user can see the aligned contents, annotations, author, narrator, etc. of the book. User can change the mode of the navigator using the wheel mode navigator. After the user opens the audiobook

mode, the user can toggle start\stop the audiobook, rewind and forward a sentence or rewind or forward 15 seconds. Same activities can be done in the narrated book mode.

3.5.5 User Interface - Navigational Paths and Screen Mock-ups

3.5.5.1 Login/Sign Up Screen

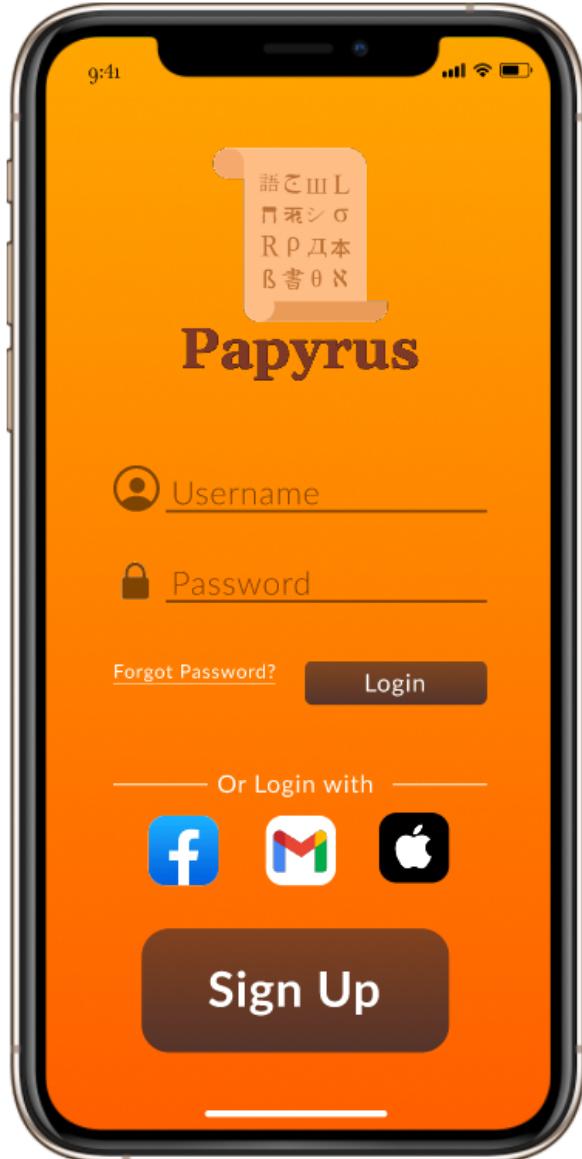
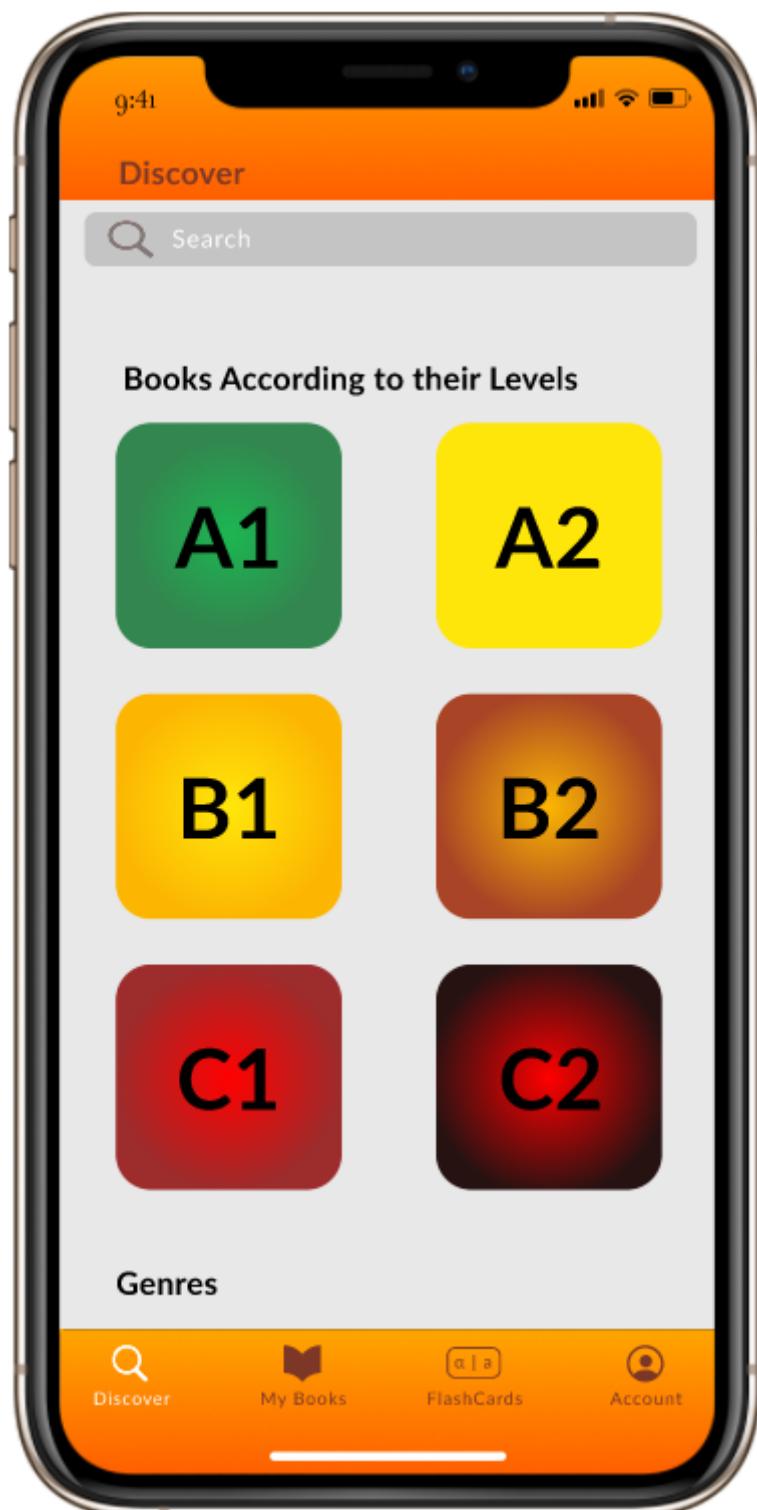


Figure 10: Log In Screen

On this screen, the user will be able to log in by username/password combination, via Facebook, via Google or via Apple. Or, they will be able to create an account. The user will be able to change their password as well.

3.5.5.2 Discover Panel



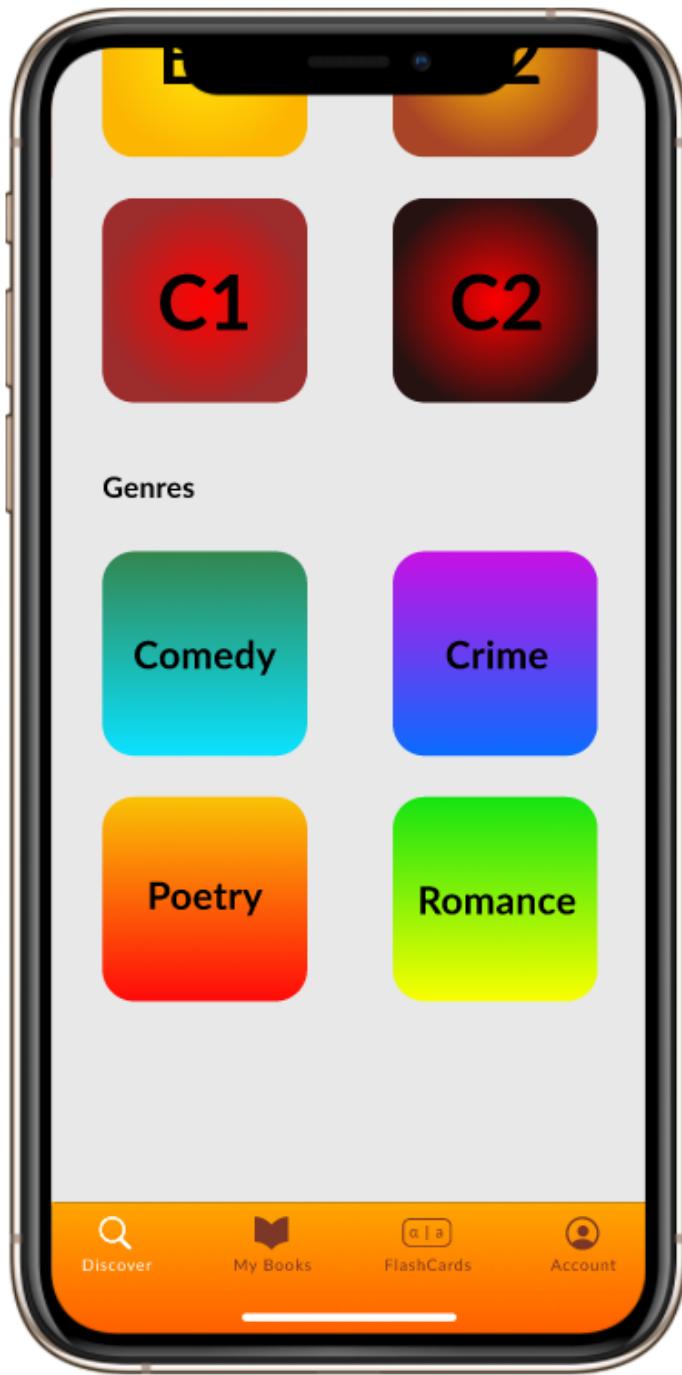


Figure 11: Discover Screen

This is the first screen that the user will encounter when they open the application.

On this screen, Book collections and genres will be shown to the user. This will be a scrollable screen. Also, there will be a search bar to search for a specific element in the database. The user can navigate through other panels by pressing the little buttons on the bottom of the screen.

3.5.5.3 Opening a Collection in Discover

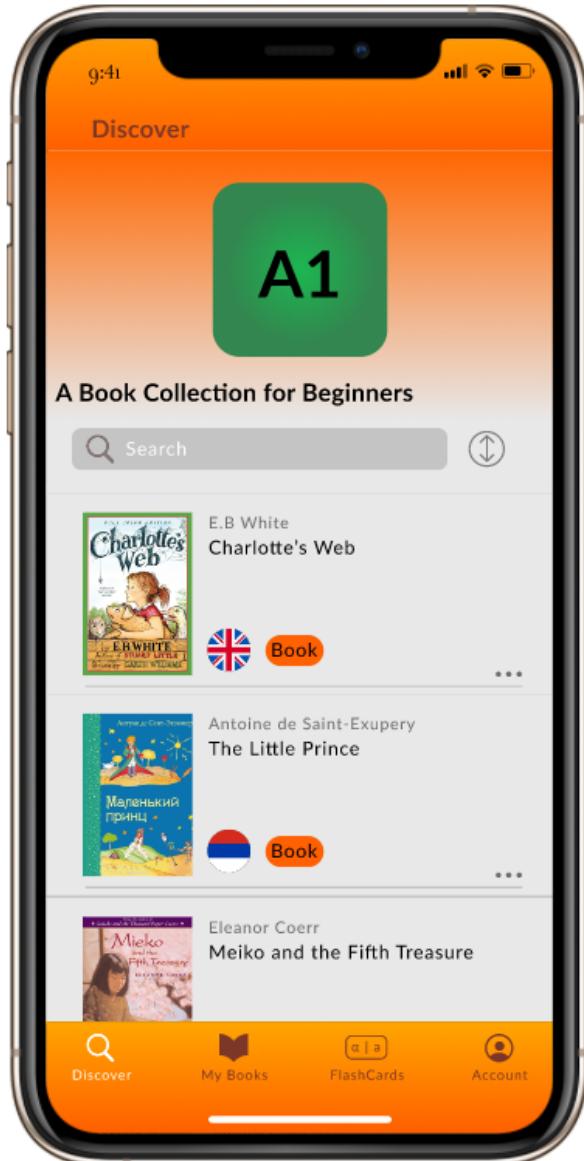


Figure 12: Collection Screen

This is how a collection is shown to the user when the user presses on a collection while discovering. The name of the collection and the cover image will be shown. The books in that collection will be shown one after the other vertically. The names of the books, as well as the languages of the books, will be shown. Again, this will be a scrollable screen. The user can use the search bar to search for a specific book or the user can sort the books by pressing the button next to the search bar.

3.5.5.4 Search Screen in Discover

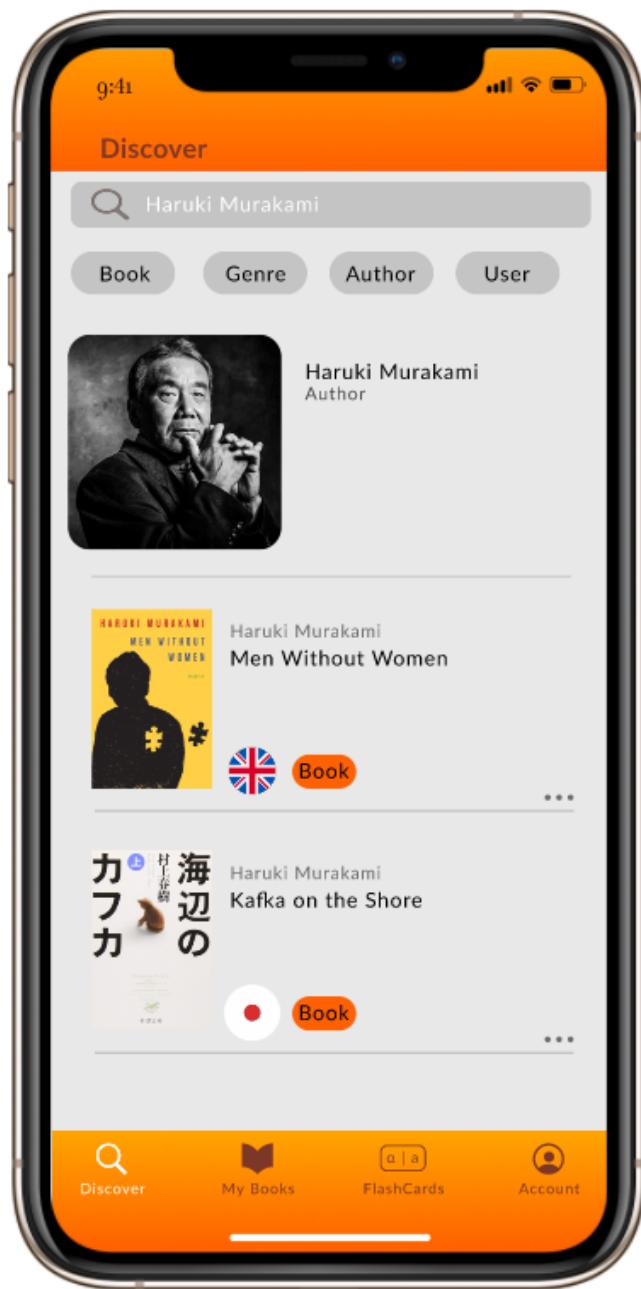


Figure 13: Search Results Screen

This is the results page when the user searches for something. The user can narrow the search results by specifying the type of the element that they want to see by using buttons below the search bar. In the results page, books, genres, authors and collections will be shown.

3.5.5.5 An Author's Page in Discover

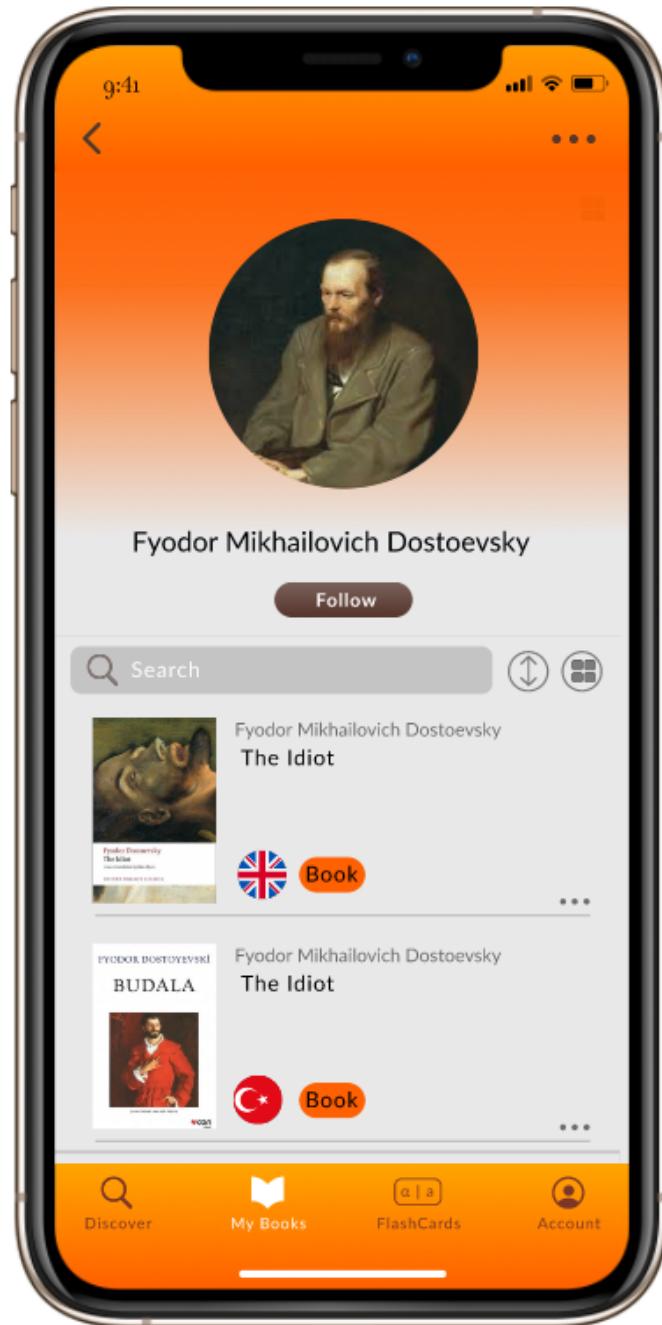


Figure 14: Author Screen

This screen is what is shown when an author is selected in the Discover Panel. The user can follow the author by pressing the follow button. The books of the author will be shown. The user, again, can search for a specific book by using the search bar, or sort the books.

3.5.5.6 FlashCards Main Screen



Figure 15: FlashCards Main Screen

This is the main screen of the FlashCards panel. Here, all the decks created by the user, as well as the decks downloaded by the user will be shown. The user can search for a deck using the search bar or sort the decks. With the edit button, the user can go to the Edit Flashcard screen. With the button next to the sort button, the user can change the view. The user can also see the number of cards in a deck.

3.5.5.7 FlashCards Main Screen: Edit Decks



Figure 16: Edit Decks Screen

This screen appears after the user presses on the edit button on FlashCards Main Screen. The user can create a new deck by clicking on the “New Deck” button. The user can remove a deck by clicking on the minus button for that deck. The user can see more options by clicking on the “Three Dot” button on a deck.

3.5.5.8 Edit Decks More Option Screen

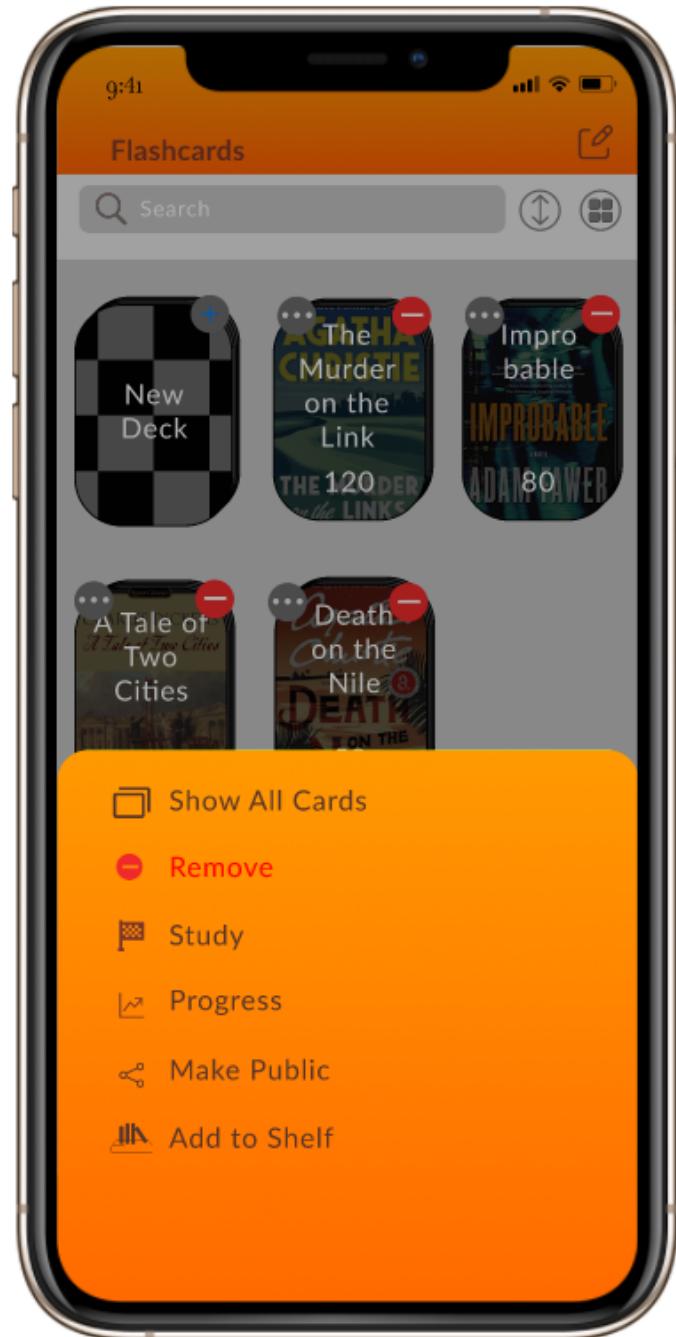


Figure 17: Decks More Option Screen

This pop-up appears after the user clicks on the “Three Dots” button. Clicking on the respective buttons of this pop-up, the user can see all the cards in a deck, remove a deck, study that deck, see their progress on their deck, make the deck public or they can add their deck to their shelf (My Books).

3.5.5.9 FlashCard Practice

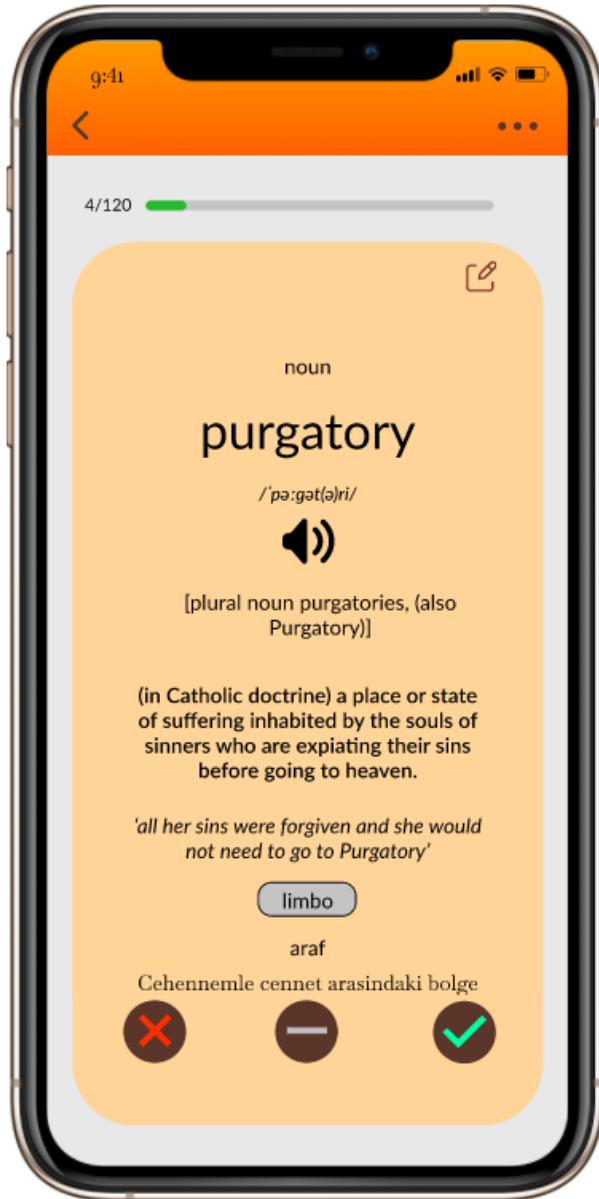


Figure 18: FlashCard Screen

This is the screen of the flashcard practice. In the practice, cards will come one-by-one and the user will answer as either fail, or not sure, or good based on their knowledge. This is the front side of a card. On this card, selected fields about a word will be displayed. By clicking on the card, the user can see the back of the card. The user can edit the card by clicking on the edit button top left on the card.

3.5.5.10 Adjust Front/Back of a FlashCard

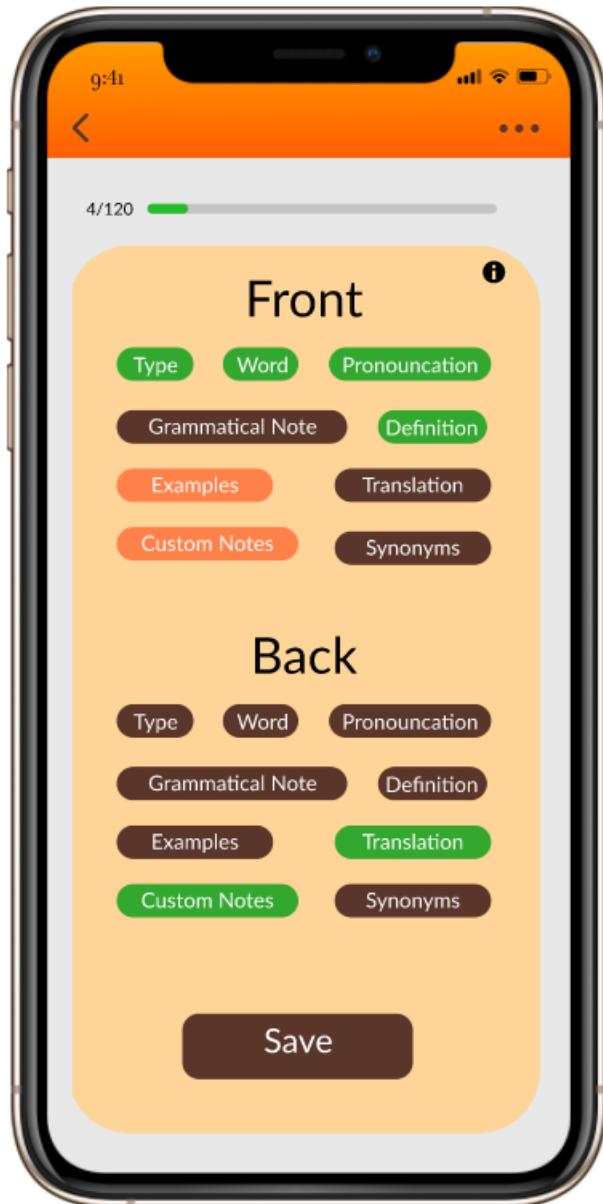


Figure 19: Adjust Front/Back of FlashCard Screen

This screen is shown when the user clicks the Edit Card button. Here, the fields on the front and back of the card can be managed by clicking on the field. By clicking on the field, the user can make the field visible or hidden (Green means visible and Brown means hidden), and by holding a button, the user can make the field optional, hidden on a card yet can be seen at any moment by clicking on it.

3.5.5.11 Show All Cards Screen

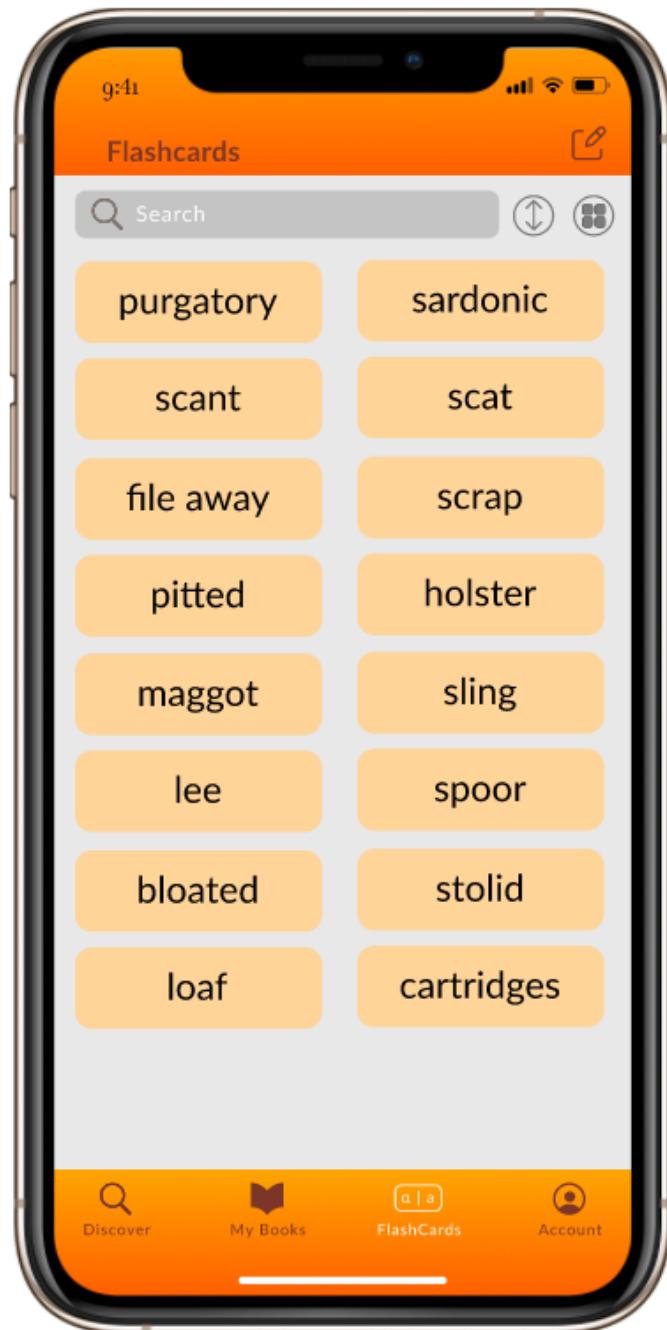


Figure 20: See All Cards Screen

This screen is shown when the “Show All Cards” button on the “More Options” screen is clicked. The user can see all of the cards’ words at once. By clicking on a card, the user will see the whole card. The user can add a new card by clicking on the edit button on the top right.

3.5.5.12 Add New Card Screen



Figure 21: Add New Card Screen

This is a horizontally scrollable screen. When the user creates a new card, the application provides different dictionaries' information of that word. The user can select fields from these dictionaries to be on the card.

3.5.5.13 FlashCard Pop-up Settings



Figure 22: FlashCard Pop-up Settings Screen

This is the screen for flashCard pop-ups settings. Users can order and choose the content for pop-ups for the words.

3.5.5.14 MyBooks Panel

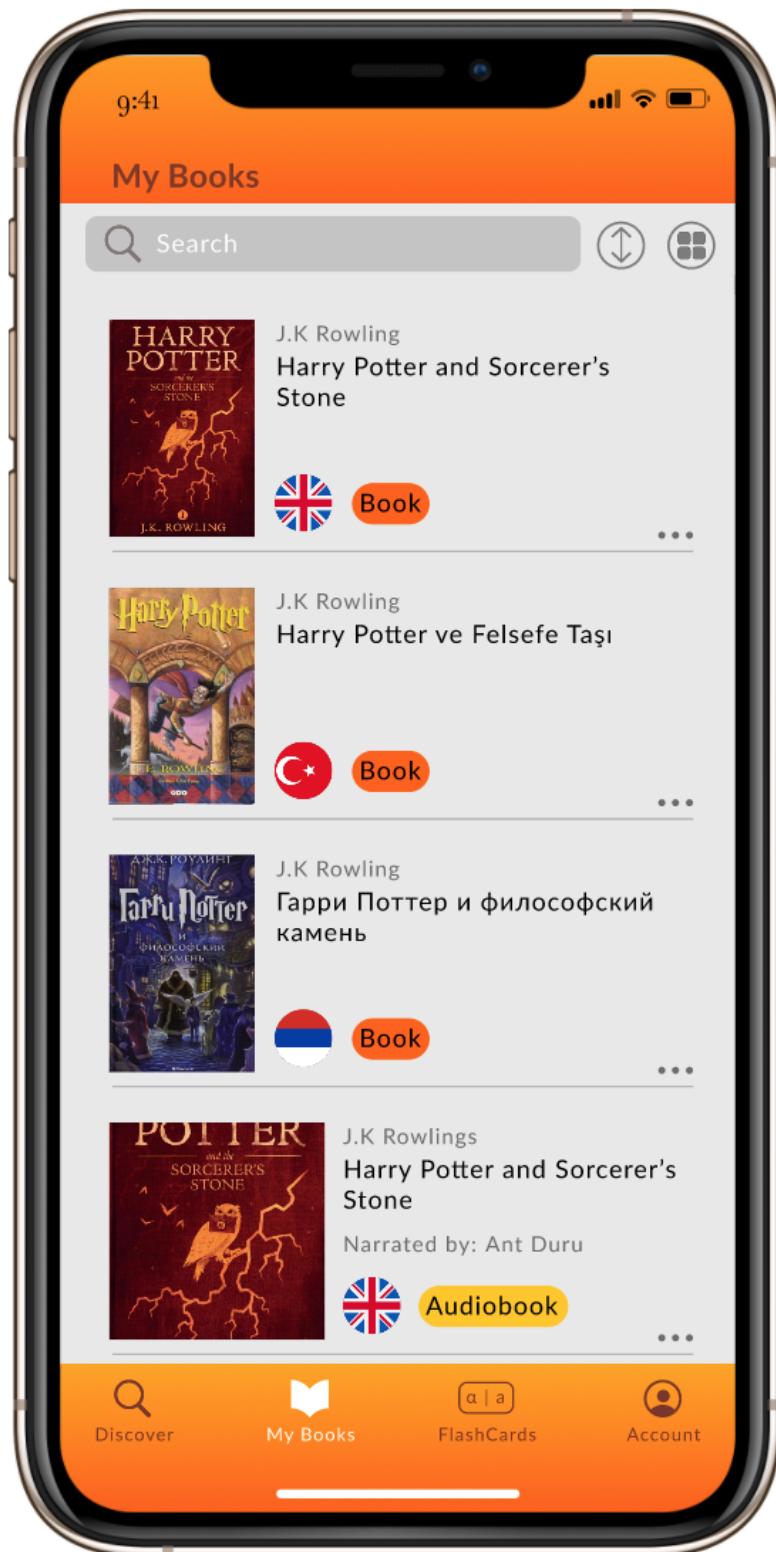


Figure 23: MyBooks Screen

This is the MyBooks screen where all the books, shelves, dictionaries can be seen and opened.

3.5.5.15 MyBooks Sort and View Menu

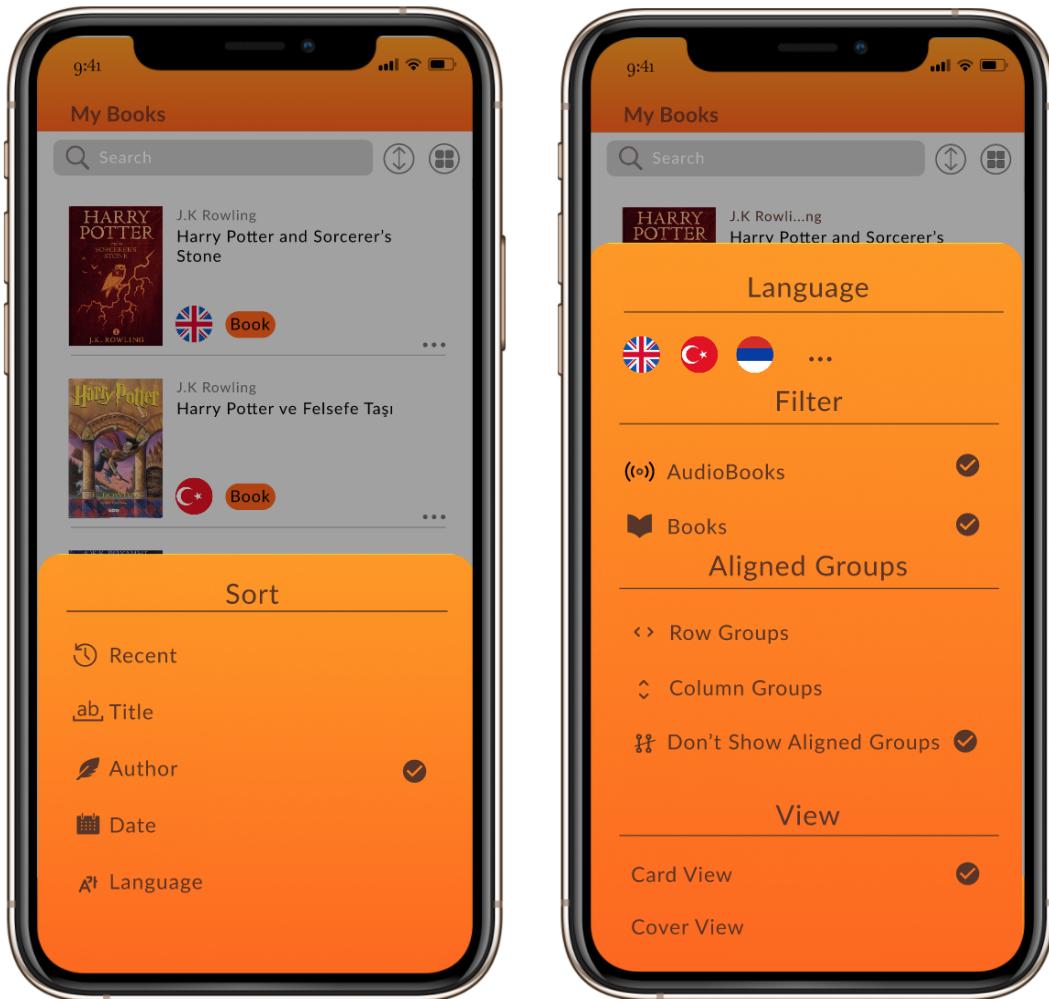


Figure 24: MyBooks Sort and View Screen
These menus are for viewing and sorting settings of the MyBooks panel.

3.5.5.16 My Books Shelf View



Figure 25: Shelf Screen

This is a shelf view -collection of books.

3.5.5.17 Navigator: Text Mode

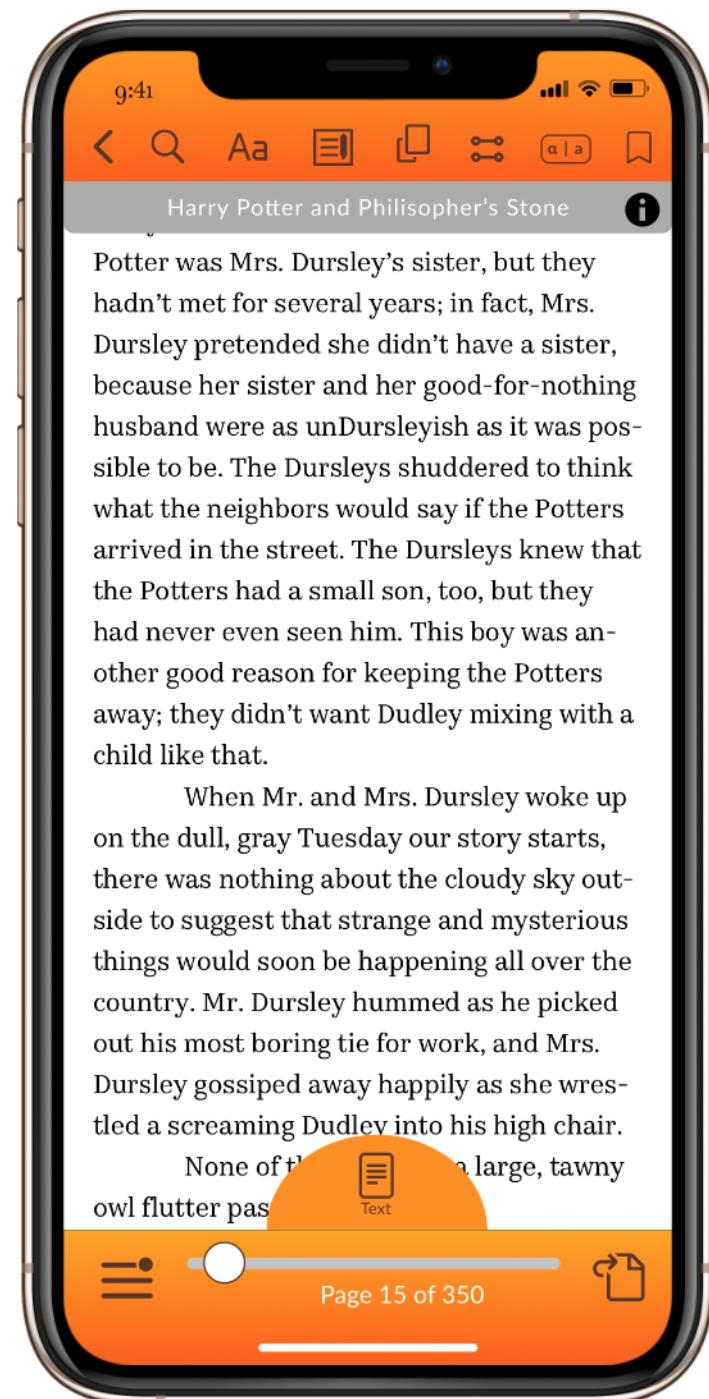


Figure 26: Navigator: Text Mode Screen

This is the navigator user interface in text mode. User can read book and adjust various settings from here.

3.5.5.18 Navigator: Word Selection Pop-ups



Figure 27: Word Selection Pop-up Screens

When users select a word by holding customisable pop-ups will show up. These would be dictionaries, web sources, machine translations or aligned texts. Users can annotate or highlight, create a flashCard from word, or copy and search the word.

Users can hide pop-ups with a swipe.

3.5.5.19 Navigator: Phrase Selection Pop-ups

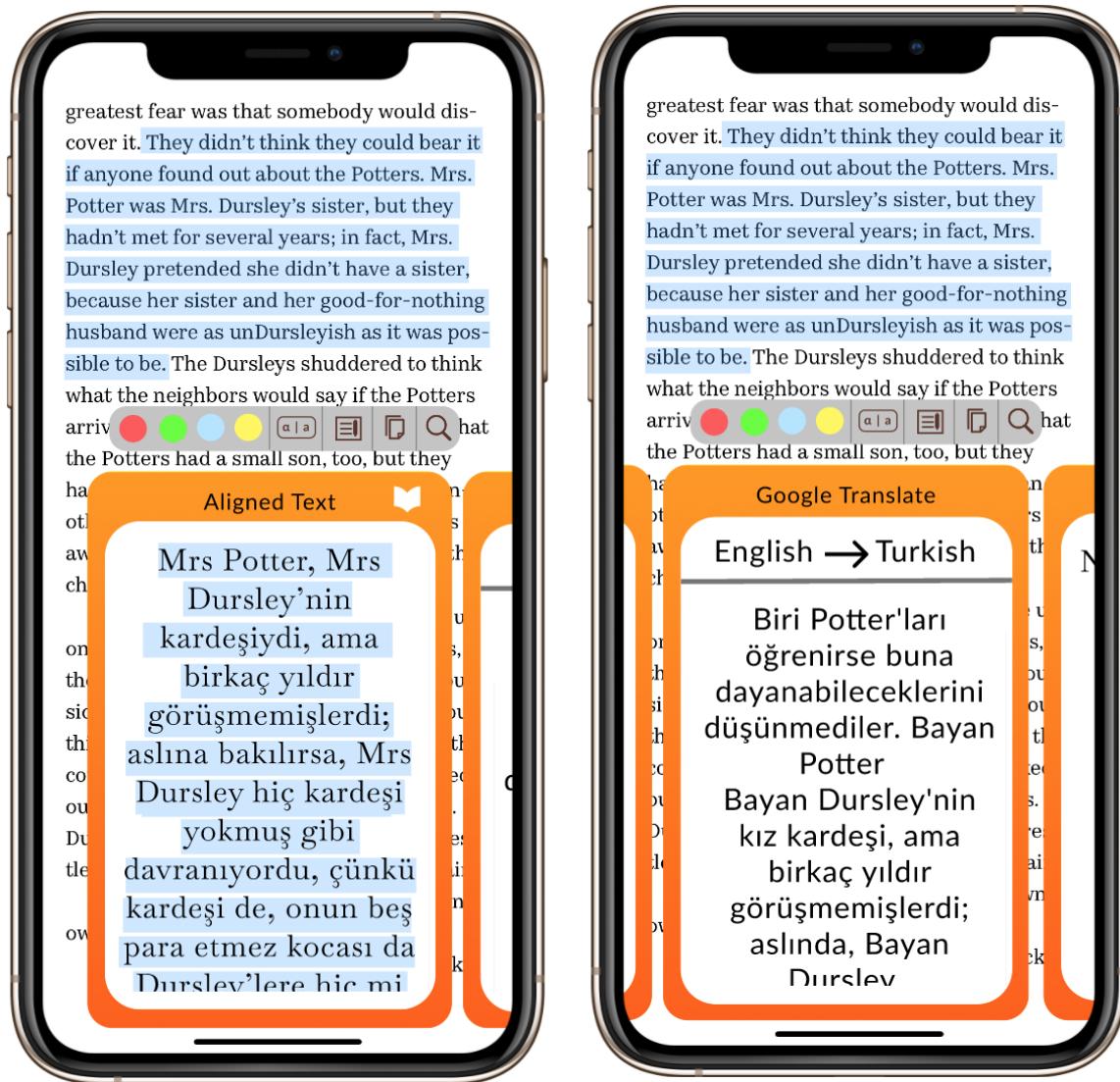


Figure 28: Navigation: Phrase Selection Pop-up Screen

These are pop-ups created when the user selects a phrase. This shows one of the important features of the Papyrus. Users will be able to see the aligned texts of selected phrases directly. Users can open the translation books by tapping the book icon if the aligned portion is not sufficient. Users can use the machine translations too.

3.5.5.20 Navigator: Pop-ups Settings



Figure 29: Pop-up Settings Screen

Users may want different pop-ups for word and phrase selections. This is why we separated them. Users can change the order and the content of the pop-ups from these screens.

3.5.5.21 Navigator: Aligned Contents Settings

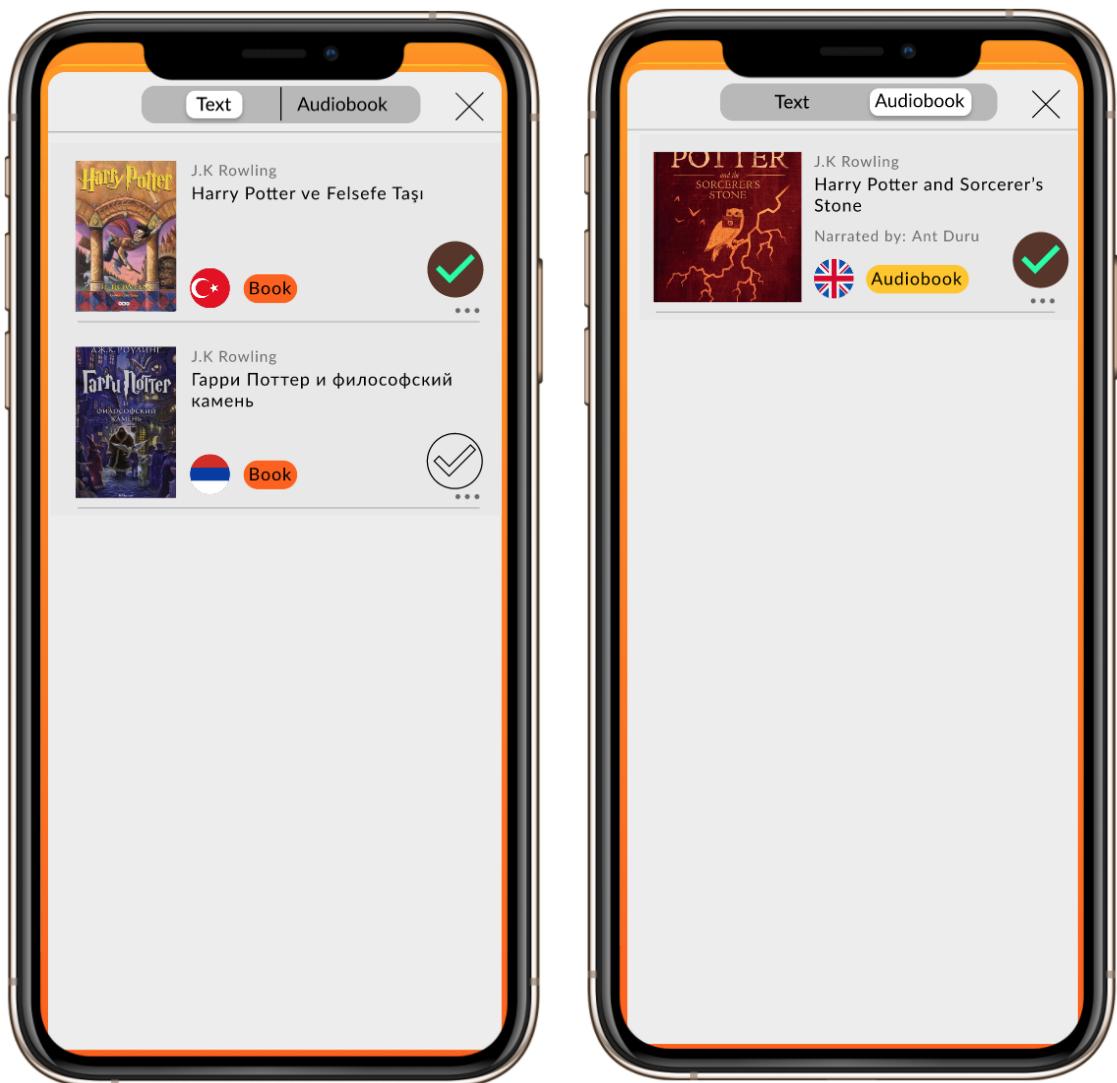


Figure 30: Aligned Content Settings Screen

This screen is for adjusting aligned texts and aligned audiobooks. The user can select multiple aligned texts to see them in selection pop-ups. Only one audiobook can be selected for audiobook mode and narration mode.

3.5.5.22 Navigator Creating Annotations and Highlights

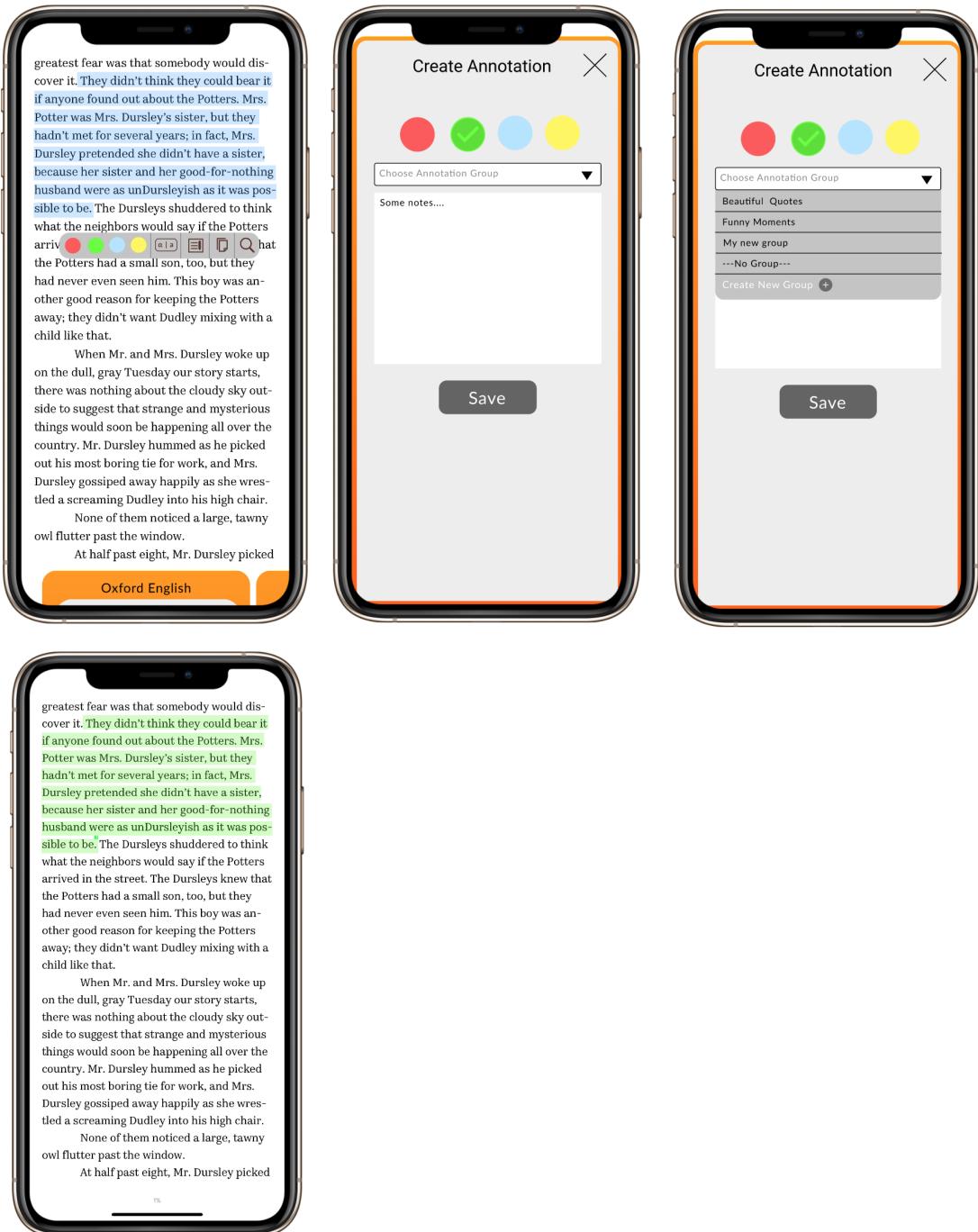


Figure 31: Annotation Creation Screen

These mockups show highlights and annotation processes. After selecting a phrase or word, the user taps the annotation icon or directly one of the colors to only highlight without a note. The user can create annotation groups and share them with others.

3.5.5.23 Navigator Creating Bookmarks and Annotations

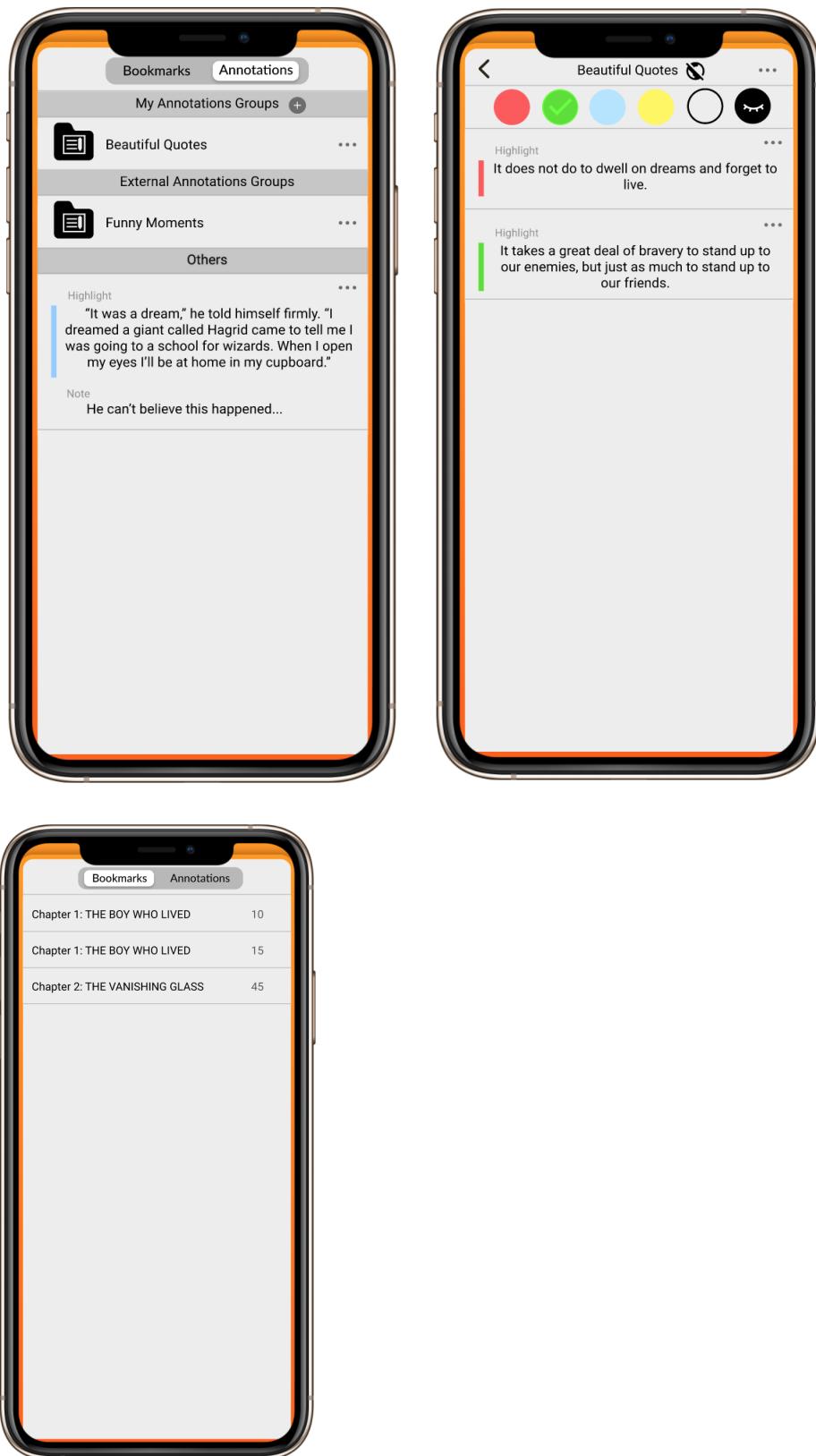


Figure 32: Create Bookmark Screen

These screens show the book's annotation groups and bookmarks.

3.5.5.24 Navigator: Chapter Navigator and Font Settings

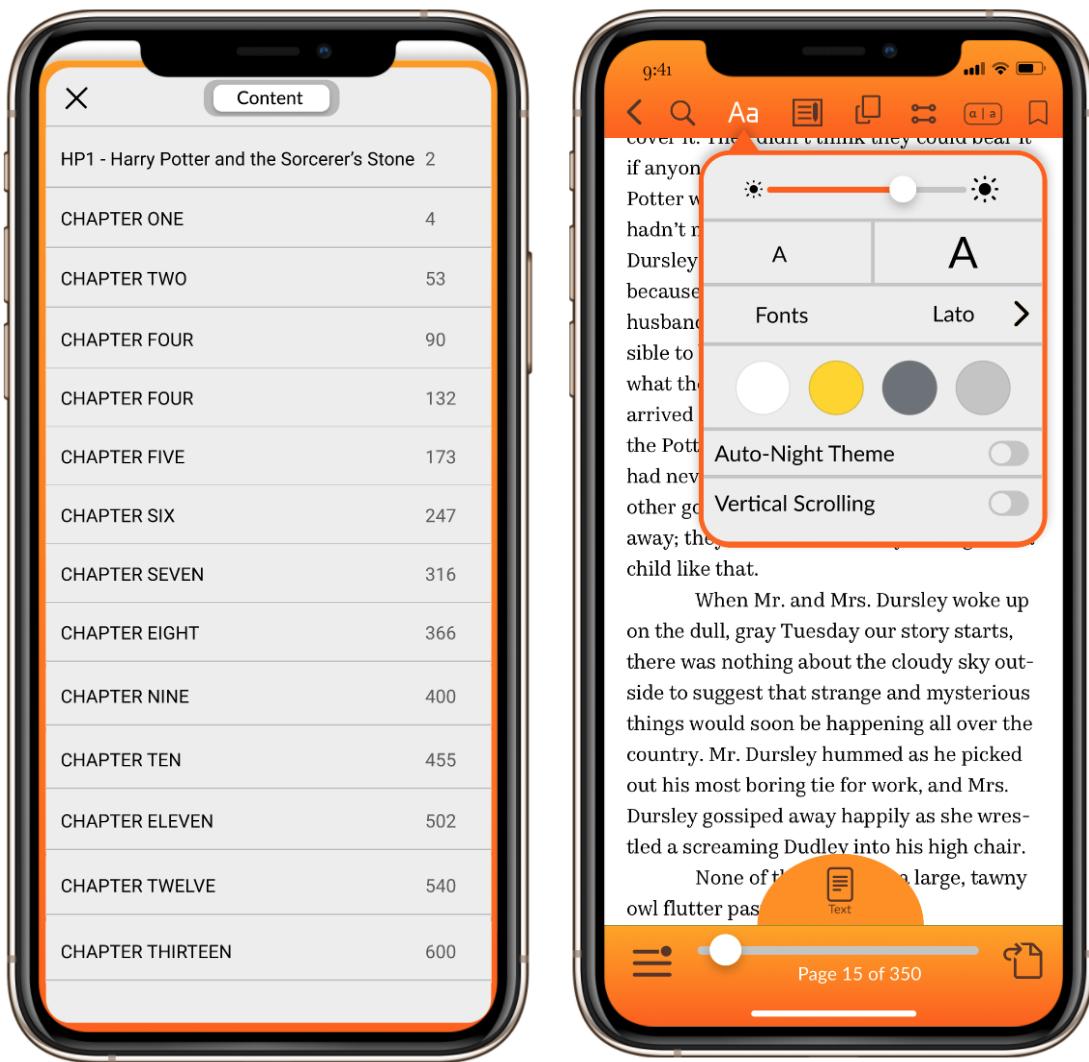


Figure 33: Chapter Navigator and Font Settings Screen

3.5.5.25 Navigator: Change Mode: Audiobook, Narrated Book, Text Book

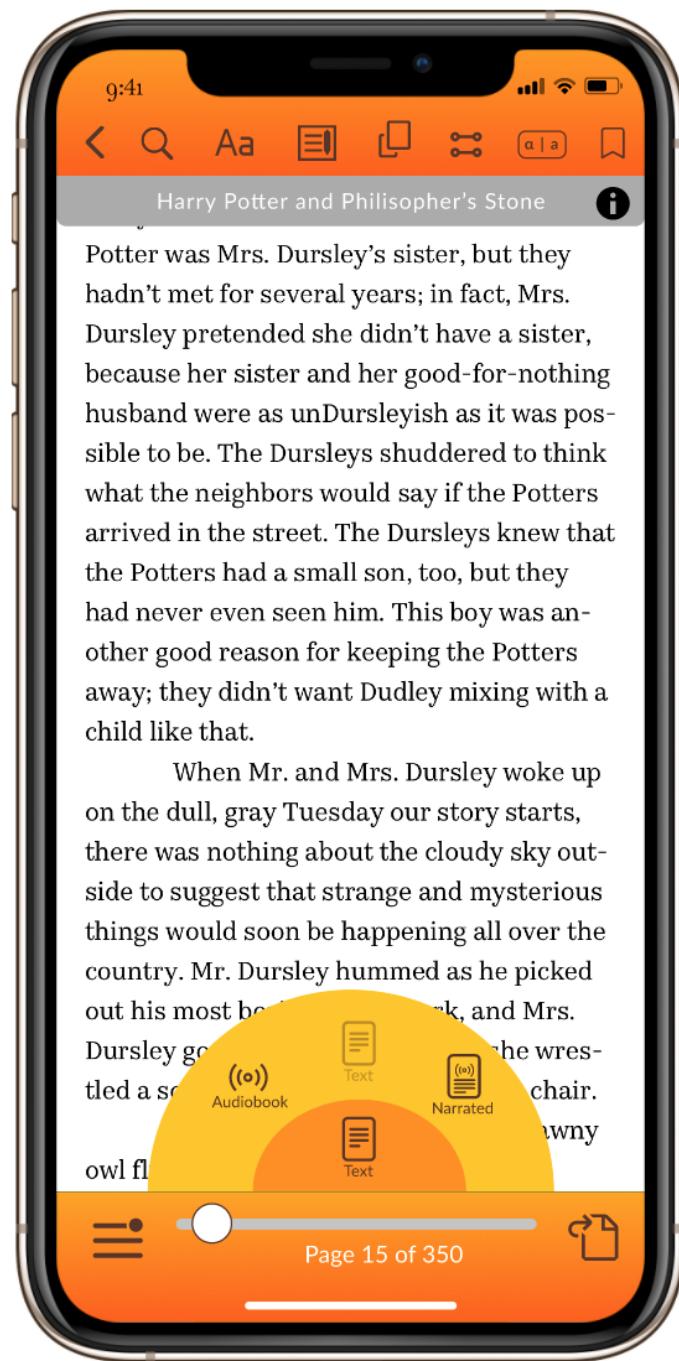


Figure 34: Change Mode Screen

This screen shows the wheel navigator to change the mode of the navigator. Users can switch to Audiobook mode, Narrated Mode, Textbook mode.

3.5.5.26 Navigator: Audiobook Mode



Figure 35: Audiobook Mode Screen

3.5.5.27 Navigator: Narrated Mode

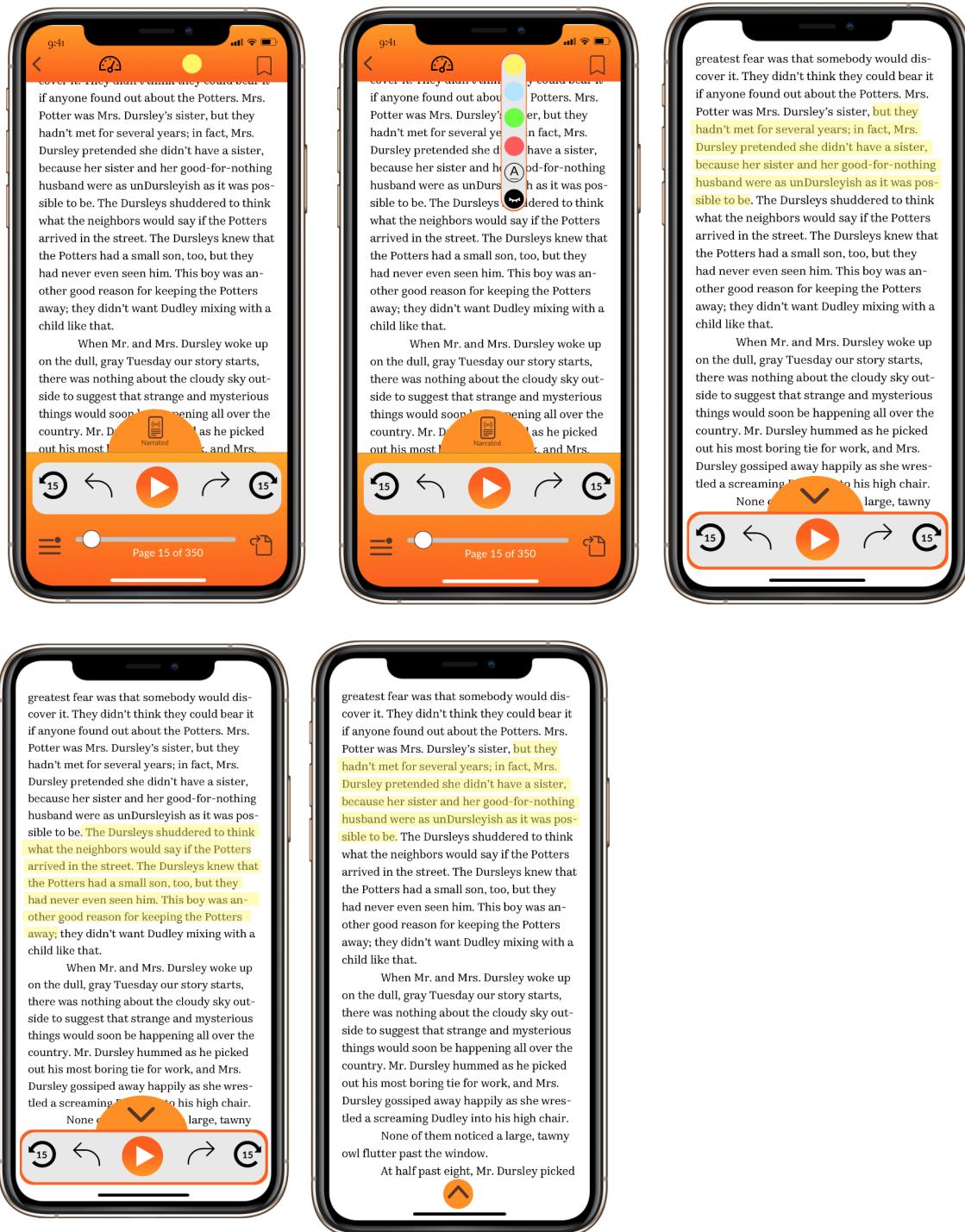


Figure 36: Narrated Mode Screen

These mockups show the Narrated Mode. Aligned with the audiobook, text will highlight the sentences along with the audiobook.

3.5.5.28 Account Panel



Figure 37: Account Panel Screen

These screens include the user's own public page and settings. Users will be able to see their created shelves, reviews, followed authors and users. Application settings are adjusted from this panel. Users can change their general settings, preferred languages (which change the book pages of the discover panel). Users can add profile pictures and look at legal documents and versions.

4. Other Analysis Elements

4.1 Consideration of Various Factors in Engineering Design

4.1.1 Public Health

Learning a new language may be mandatory under several circumstances, such as studying or working abroad, skipping university preparation school, etc. With PAPYRUS, we aim to reduce the level of stress of language learning by creating a safe, library-like environment for language learners, which may affect their psychological state positively. Apart from that, unfortunately, PAPYRUS does not have a goal to influence public health beneficially.

4.1.2 Safety

PAPYRUS will not harm the safety of personal data since the data of users, publishers, or books will not be shared with third parties.

4.1.3 Welfare

Since language learning may help people to find new jobs, develop new relationships with others and help self-actualization, PAPYRUS will aim to affect public welfare positively.

4.1.4 Global

PAPYRUS can be used globally. Language learners from all around the world will be able to enter the application to read books written in a specific language. Also, languages in the application will be updated regularly to present more languages to learn.

4.1.5 Cultural

PAPYRUS aims to broaden the cultural perspective of people by offering the opportunity to learn a new language by reading books of that language. Since literature is a significant field to represent a culture, users will encounter new cultures and gain knowledge of the cultures as well.

4.1.6 Social

PAPYRUS aims to create communities of language learners with respect to their favorite authors, genres, books, etc. Furthermore, since people will be able to share their notes and annotations for a book, their book collections, and their flashcard decks; social interaction will be affected positively.

4.1.7 Environmental

According to Cleantech, a single hard-copy book generates about 7.5 kg of carbon dioxide, while this value is doubled for a textbook. However, an iPad

generates 130 kg of carbon dioxide in its lifetime. Therefore, having an iPad neutralizes its harm when the user reads approximately 18 e-books.[1]

Consequently, e-books are much more nature-friendly than hard-copy books. By digitalizing the libraries as we aim with PAPYRUS, readers' carbon footprints will also reduce by a significant amount.

4.1.8 Economic

Subscription plans are great for keeping users in their monthly budget, because subscription cost is always consistent and predictable, making the application more convenient to access. Users will pay for the subscription to the application. Yet, e-books will be cheaper than hard copy books. Therefore, users will not be affected negatively in terms of economic issues. Also, if the business is suitable for the mode, the subscription model can be great for the business owner. Business owners calculate the predicted revenue more easily, due to the recurrent nature of the subscription model. John Warrilow says, creator of the Value Builder System, "The more guaranteed revenue you can offer a potential acquirer, the more valuable your business is going to be, because a high percentage of the revenue of a subscription-based business is recurring, its value will be up to eight times that of a comparable business with very little recurring revenue." [2]. In that way using a subscription base model, made sense to us, we

	Effect Level	Effect
Public Health	1/10	Offering an unstressful learning environment
Safety	5/10	Protecting the data of the users, against pirating
Welfare	5/10	Language learning leads one to self-actualization
Global	7/10	Increase communication globally by teaching a new language
Cultural	9/10	Language learning broadens the cultural perspective and enriches the knowledge about other cultures
Social	8/10	presenting a social media-like e-reading platform to share thoughts on languages and books
Environmental	7/10	Using e-books, whose carbon footprint is much less than hard-copy books
Economic	6/10	Offers a less price than hard-copy books'

Table 4: Effects of the Project

4.2 Risks and Alternatives

Our project is broad, we aim to make an application that can contend with popular and complex applications such as Spotify, Kindle and Storytel. Our application will be constructed with a similar fashion of database, machine learning algorithms and user interface methods of the aforementioned applications. In our conjecture, this will be a hard and long task to do. In that way a number of problems and risks can arise that can hinder the process of the application. These are the possible risks and our solutions, we thought in advance.

- Lack of group members considering the other groups:

Our group consists of three people, two people deficient from the 5 people group limit, that puts the project at risk in case of work power.

Our solution: Not always less people means less work power. Cells divide when they become too big, because it gets harder to control the resources. Just like a cell, a concise group means that our combined potential work power can be used more efficiently, because of easier planning and controlling each member's work. Also three people can reach a consensus more easily, therefore disputes over the project's direction and design potentially will be lower.

- Copyright issues regarding the publishers:

Our application is new and is developed by undergraduate students that have no name in the industry. In that way, we predict getting publishers permission to feature their books in our application can be cumbersome.

Our solution: First of all, we believe that our application will attract publishers' attention with our novel idea. However, if a publisher does not allow us to

feature their book in our application, we have a way that our users can benefit. If a user has a legal copy of a book, in his/her local device, he/she can upload the book in the application. Therefore if a book is not included in our repertoire, there is a way that users benefit from our application via uploading in the local file.

- Possibility of project being exceedingly broad:

Just like it was mentioned before, we acknowledge the fact that our project is broad. However there is a possibility that the extent of the project can be more than anticipated. That we could not implement every detail to the final product in time.

Our solution: We know that there is always a possibility of insufficient implementation due to time constraint, our completing the project is beyond our capabilities. In order to eliminate this problem, we plan the progression in advance. Also, the development of the project started back in summer of 2021, in that way the conceptual design and disputes over the high level implementation is mostly eliminated.

Risk	Likelihood	Effects on the project	B Plan
Lack of group members considering the other groups hindering the project progression	3/10	Decreased work power, insufficient for our project.	We believe that we do not need a B plan, our formation of the group will be more efficient rather than a bigger more unstable group.
Copyright issues regarding the publishers	6/10	Less featured books in the application	Uploading books from local device
Possibility of project being exceedingly broad	5/10	Project not finished as we fully realized. Buggy project. Details are missing.	Starting early, doing the works in advance.

Table 5: Risks and Alternatives

4.3 Project Plan

WP#	Date	Work Package	Leader	Participants
		Title		
WP1	16.09.2021- 11.10.2021	Initial planning	Enver Yiğitler	Enver Yiğitler, Ant Duru, Atay Kaylar
WP2	12.10.2021- 8.11.2021	Analysis	Ant Duru	Enver Yiğitler, Ant Duru, Atay Kaylar
WP3	8.11.2021- 24.12.2021	High-level design and front-end	Atay Kaylar	Enver Yiğitler, Ant Duru, Atay Kaylar
WP4	24.12.2021- 10.05.2022	Continuous bugfix	Atay Kaylar	Enver Yiğitler, Ant Duru, Atay Kaylar
WP5	24.12.2021- 10.05.2022	Low-level design and backend	Enver Yiğitler	Enver Yiğitler, Ant Duru, Atay Kaylar
WP6	24.12.2021- 10.05.2022	Database and NLP algorithms	Ant Duru	Enver Yiğitler, Ant Duru, Atay Kaylar

Table 6: Project Plan

WP 1: Initial Planning

Start date: 16.09.2021 **End date:** 11.10.2021

Leader:	<Enver Yiğitler>	Members involved:	< Ant Duru, Atay Kaylar>
----------------	------------------	--------------------------	--------------------------

Objectives: <*In this stage, we discussed our project ideas as a group and declared the e-reading platform for language learners as our senior design project. The task in this time interval is to create a base for our project, discuss fundamental features, planning the UI, planning about the data storage, and starting screen mock-ups early to have a better understanding and plan about the project. An essential part of the planning was to find a supervisor for our project. Therefore, we inspected all of the academic people's interests and decided whether the fields are compatible with our project's. Lastly, we completed our project specification report.*>

Tasks:

Task 1.1 <Discussing the ideas> : *<We discussed the project ideas with each other, added or removed features from the application according to our ideas. We constructed a conceptual version of the project.>*

Task 1.2 <Visualizing the application: <After the conceptual understanding of the project, we tried to visualize the project using barebone mock-ups>

Task 1.3 <Finding a supervisor> : *<We had to find a supervisor, so we mailed the suitable academics that can instruct us throughout the progression. We tried to find a supervisor that has a vast knowledge about NLP, in that way we can maximize the benefit from working with a supervisor. We have been accepted by Fazli Can >*

Task 1.4 <Completing the project specification report>: *<We completed project specification report, in the report we introduced our project.>*

Deliverables

D1.1: <Project specification report>

WP 2: <Analysis>

Start date: <12.10.2021> **End date:** <15.11.2021>

Leader:	< <i>Ant Duru</i> >	Members involved:	< <i>Enver Yiğitler, Atay Kaylar</i> >
----------------	---------------------	--------------------------	--

Objectives: <In this stage, we finished our mock-ups first to navigate our perspective better through the project. We declared our classes and partly our methods. We began trying to get fluent in Swift language. We created our analysis report through discussions about use-cases and diagrams. Furtherly, we discussed the features of our application again. We did research about libraries and frameworks, which will help us for the dataset, dataset integration with the backend and server-side programming such as Vapor, MongoDB, SQLite.>

Tasks:

Task 2.1 <Creating mock-up using Figma> : <Firstly, we started designing the UI. We believe that interface and user experience is an essential part of the project, so we put serious attention to the interface.>

Task 2.2 <Learning the Swift language> : <We will compose our project with using Swift language, although we are familiar with learning other languages, we should be fluent to make this project properly.>

Task 2.3<Analysis report> : <Our first challenging report, analysis report, is due in this work package, in that way we put most of our effort into completing the report, we practice group meeting two to four times a week in this time span.>

Task 2.4<Continuous research> : <*Every step of the way, we have not stop strategizing about the upcoming tasks. We tried to grasp the understanding of implementation of the project in the backend, and most important NLP algorithms, with researching and reading article.*>

Deliverables

D2.1: <*Analysis report*>

WP 3: <*High-level design and front-end* >

Start date: <8.11.2021> **End date:** <24.12.2021>

Leader:	< <i>Atay Kaylar</i> >	Members involved:	< <i>Enver Yiğitler, Ant Duru</i> >
----------------	------------------------	--------------------------	-------------------------------------

Objectives: <*In this stage, we will start on our high-level design report, creating subsystem decomposition and discuss more about our architecture.*

Implementation of the frontend will be started. Therefore, learning Swift programming language as well as the libraries and the frameworks should be at a

sufficient level at this stage. If the work is done before planned, integration of the NLP algorithms to our application will start.>

Tasks:

Task 3.1 <Discussing the high-level implementation> : *<In this stage, after continuous discussion about the implementation about the project, we should have a good understanding of the implementation. In that way, we should be finalizing it with further critical discussion.>*

Task 3.2 <Finalizing the learning of Swift> : *<We should be completing our learning period of Swift and start the implementation of front-end>*

Task 3.3<Front-end implementation> : *<Previously mentioned, we will start front-end >*

Task 3.4<High-level design report>: *<High-level design report will be due in this work package; therefore, we will be working on it in this time span with our knowledge of previous discussions and research.>*

Task 3.5<Early implementation of backend>: *<If the previous tasks are finished before the deadline, we plan to start with implementation of NLP algorithms into the application, which is practiced on a server>*

Deliverables

D3.1: *<High-level design report>*

WP 4: <Continuous bugfix>

Start date: <24.12.2021> **End date:** <10.05.2022>

Leader:	< Atay Kaylar >	Members involved:	< Enver Yiğitler, Ant Duru >
----------------	-----------------	--------------------------	------------------------------

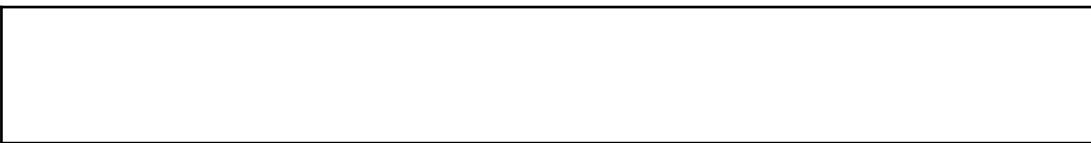
Objectives: < After finishing the frontend and during the backend, database integration and NLP model utilization, a process of bug fixing continuously should be done. Therefore, each group member will spare some time to look at bugs and errors through the second semester:>

Tasks:

Task 4.1 <Bug fixing> : <Back-end and front-end components should be starting to come together in this stage, due to complex nature of the project this process will produce bugs. We expect too many errors and bugs. End of the fall semester and start of the spring semester are expected to be filled with continuous bug fixing>

...

Deliverables



WP 5: <Low-level design and back-end>

Start date: <24.12.2021> **End date:** <10.05.2022>

Leader:	< <i>Enver Yiğitler</i> >	Members involved:	< <i>Ant Duru, Atay Kaylar</i> >
----------------	---------------------------	--------------------------	----------------------------------

Objectives: < *Swift language can also be used for the server side of the implementation. Therefore, after the base of the frontend is done, implementation of the backend will start immediately. The reason for the prioritization of the frontend is that we value the visual side of the application when it comes to create a fast, usable e-reading platform. With the beginning of the second semester, low-level design will be handled as well, and the report will be written.*>

Tasks:

Task 5.1 <Presentation and Demo for fall semester> : <Along the implementation, we should create a demo of our project, which will be presented at the end of the fall semester, for a demonstration version a working project, a sufficient version of the project should be created with proper back end and front-end implementation.>

Task 5.2 <Finalizing the front-end> : <This work package constitutes a very long-time interval; in that way our bulk of the front-end implementation should be finalized.>

Task 5.3 <Further-implementation of back-end> : <Back-end implementation will further improve in this work package. We will put most of our effort into back-end, from now on.>

Task 5.4 <Low-level design report> : <We expect low-level design report in this time span, in that way we will constitute a low-level design report according to previous tasks. >

Deliverables

D5.1: <Presentation materials>

D5.2: <Demonstration version of the project>

D5.3: <Low-level design report>

WP 6: <Database and NLP algorithms>

Start date: <24.12.2021> **End date:** <10.05.2021>

Leader:	< <i>Ant Duru</i> >	Members involved:	< <i>Enver Yiğitler, Atay Kaylar</i> >
----------------	---------------------	--------------------------	--

Objectives: < *NLP algorithms are the uniqueness of our application considering the applications in the market right now. Therefore, we want to integrate the algorithms to our project flawlessly. We will analyze the selected algorithms in terms of their metrics through reading the papers and discussing them as a group. Since the reading experience is the key to our project, algorithms such as cross alignment, force alignment, etc., readers should not encounter any problems regarding the translation. Furthermore, the application holds a huge amount of data considering user information, publisher information, books, collections, audiobooks, voice files of narrations, annotations, notes, decks and flashcards. Thus, we need to provide an adequate database system to hold this data without any problem. We will use the selected frameworks to integrate our database to the backend of the application. Therefore, integrating back-end and front-end together to finalize the project.*>

Tasks:

Task 6.1 <Finalizing algorithm implementation> : *<The application is about language learning; therefore, the alignment of the text should be working flawlessly. It is the backbone of the application. Therefor we should apply the NLP algorithms flawlessly.>*

Task 6.2 <Creating a good database> : *<Our application's database system is crucial, most of the important operations happen in the database when users navigate and make activities in the application. Therefore, the integration of the database should be robust and stable.>*

Task 6.3 <Finalizing the application> : *<In this stage, front-end and back-end should be finished and integrated with each other. Our project should be finalized>*

Task 6.4 <Final report> : *<After the finalizing of the project we should document our project for the final report.>*

Task 6.53 <Presenting the application> : *<As the final act of senior design project, we will present our project.>*

Deliverables

D6.1: *<Final report>*

D6.2: *<Final version of the project>*

D6.3: <Presentation of the project>

4.3.7 Gantt Chart

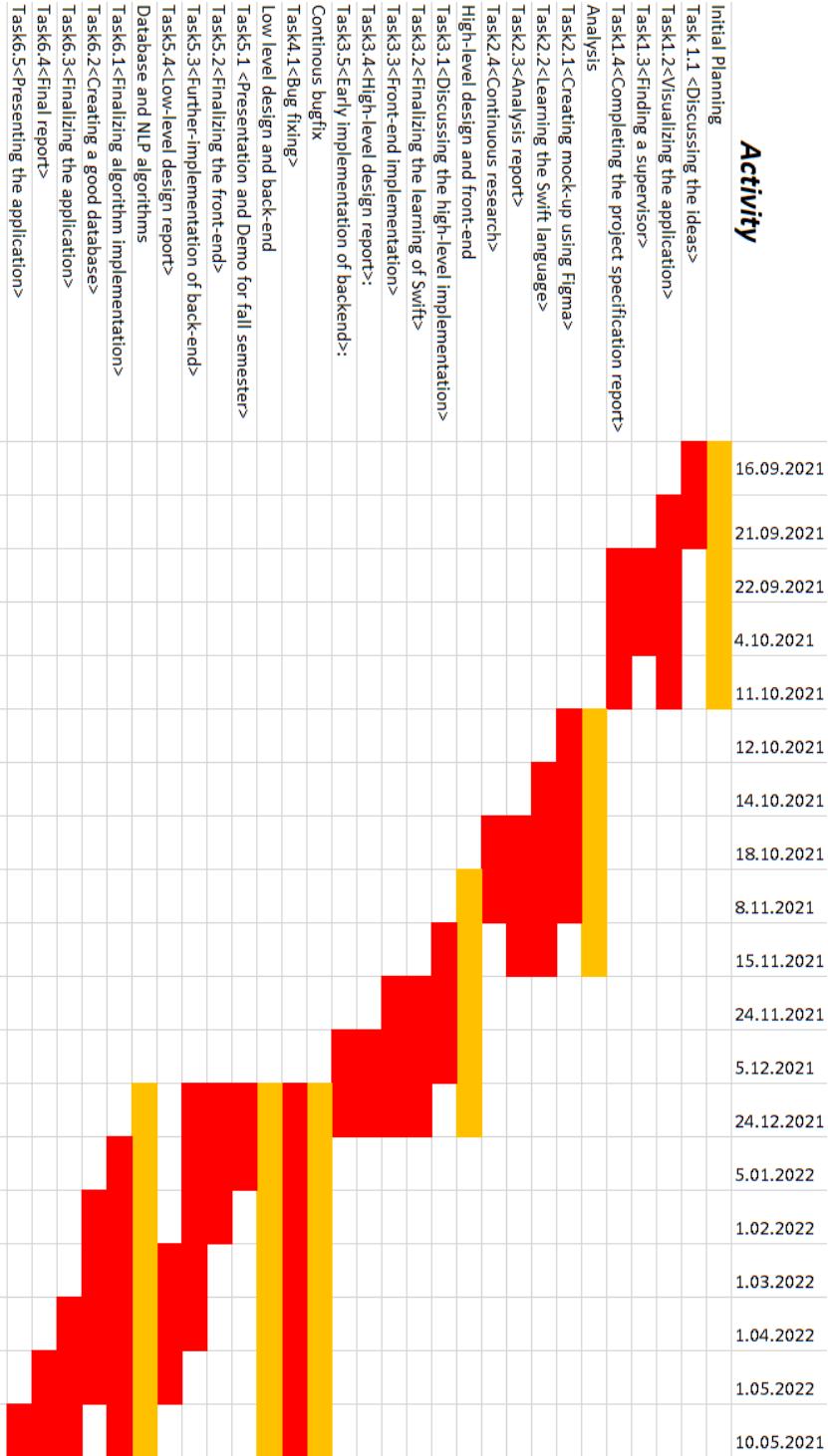


Figure 38: Gantt Chart

4.4 Ensuring Proper Teamwork

To have a fair and logical process we will divide or work into parts and people. Every one of us will be a leader of a part but everyone will participate in every part. Each work will be reviewed by everyone since we are only three people. The parts that are divided are Frontend, Backend, Database, NLP algorithms, Documentation, and APIs. Everyone should be sufficient for any part since they will do reviews for the parts whose leader is not them as well. We expect everyone to have equal contribution to both implementation and design processes.

To control the milestones, we will be in touch every time possible. There will be three meetings a week. If a leader struggles in his work, other team members will be expected to help them. If a reviewer, or a participant struggles in their work, the leader of that part will provide resources to them to improve their understanding of the concept.

4.5 Ethics and Professional Responsibilities

- The repository containing the code of the project will be private.
- All books used to demonstrate our project will have a public domain. No book without a permission will be used. In other words, the project is against pirating.
- The decisions regarding the project will be taken as a group.
- Everyone will attend regular meetings and contribute to the project as expected.

- KVKK (Kişisel Verileri Koruma Kanunu) will be applied. User information and publisher information will not be shared with third parties.

4.6 Planning for New Knowledge and Learning Strategies

In our project, there are several concepts that will be used. The list of the concepts is below.

- Backend Programming
- Frontend Programming
- iOS Application Programming
- Database Systems
- Natural Language Processing

Several frameworks and libraries should be utilized to implement our project. Each of these concepts require different techniques and knowledge. Therefore, online learning methods such as YouTube videos and Coursera courses will be used to master each concept.

5. References

- [1]Rich, Emma. "The environmental impact of Amazon's Kindle" CleanTech Group, LLC, 2009,
https://gato-docs.its.txstate.edu/jcr:4646e321-9a29-41e5-880d-4c5ffe69e03e/thoughts_ereaders.pdf
- [2]Longanecker, Chuck. "Why You Should Use a Subscription Business Model." *Entrepreneur*, Entrepreneur, 19 Mar. 2015, www.entrepreneur.com/article/243573.