



**Bilkent University  
Department of Computer Engineering**

**Senior Design Project  
T2326  
CapyNotes**

**Analysis and Requirement Report**

*22002675, Berk Temel, berk.temel@ug.bilkent.edu.tr*

*22003216, Can Ersoy, can.ersoy@ug.bilkent.edu.tr*

*21901913, İsmail Emre Deniz, emre.deniz@ug.bilkent.edu.tr*

*21901556, Kemal Kubilay Yılmaz, kubilay.yilmaz@ug.bilkent.edu.tr*

*22003609, Mustafa Burak Erkoçak, burak.erkocak@ug.bilkent.edu.tr*

*Supervisor: Fazlı Can*

*Course Instructors: Atakan Erdem, Mert Bıçakçı*

*Innovation Expert: Mustafa Sakalsız*

*Last Read By: Can Ersoy*

**08.12.2023**

## **Abstract**

Efficient note-taking during lectures, meetings, and multimedia content poses challenges for many individuals as this process affects focus and comprehension of the individual. Our project, CapyNotes, aims to address this issue by developing a note generation application designed to automate the note-taking process. Capable of accepting audio recordings or videos as input, CapyNotes aims to produce structured and comprehensive notes, allowing users to concentrate fully on the content they are listening to. The generated output note contains a summary of the recording, keywords along with their definitions. Also the application is available to create a set of flashcards from the keywords it detected.

**Keywords:** Note generation, Summary, Speech To Text, Lecture, Meeting

# Contents

<b>1 Introduction</b>	<b>5</b>
<b>2 Current System</b>	<b>5</b>
<b>3 Proposed System</b>	<b>6</b>
<b>3.1 Overview</b>	<b>6</b>
<b>3.2 Functional Requirements</b>	<b>7</b>
<b>3.3 Non-functional Requirements</b>	<b>8</b>
<b>3.3.1 Usability</b>	<b>8</b>
<b>3.3.2 Portability</b>	<b>8</b>
<b>3.3.3 Efficiency</b>	<b>8</b>
<b>3.3.4 Maintainability</b>	<b>8</b>
<b>3.3.5 Scalability</b>	<b>9</b>
<b>3.3.6 Security</b>	<b>9</b>
<b>3.4 Pseudo Requirements</b>	<b>9</b>
<b>3.5 System Models</b>	<b>9</b>
<b>3.5.1 Scenarios</b>	<b>9</b>
<b>3.5.2 Use-Case Model</b>	<b>12</b>
<b>3.5.3 Class Diagrams</b>	<b>13</b>
<b>3.5.4 Dynamic Models</b>	<b>16</b>
<b>3.5.5 User Interface Mock-Ups</b>	<b>20</b>
<b>4 Other Analysis Elements</b>	<b>24</b>
<b>4.1 Consideration of Various Factors in Engineering Design</b>	<b>24</b>
<b>4.1.1 Constraints</b>	<b>24</b>
<b>4.1.2 Standards</b>	<b>24</b>
<b>4.2 Risks and Alternatives</b>	<b>24</b>
<b>4.3 Project Plan</b>	<b>26</b>
<b>4.4 Ensuring Proper Teamwork</b>	<b>30</b>
<b>4.5 Ethics and Professional Responsibilities</b>	<b>31</b>
<b>4.6 Planning for New Knowledge and Learning Strategies</b>	<b>32</b>
<b>5 Glossary</b>	<b>32</b>
<b>6 References</b>	<b>33</b>

## List Of Figures

<b>Figure 1:</b> Comparison of CappyNotes and its competitors in the market	6
<b>Figure 2:</b> Use Case Diagram of CappyNotes	12
<b>Figure 3:</b> Class Diagram for Note Generation Service	13
<b>Figure 4:</b> Class Diagram for Notification Service	14
<b>Figure 5:</b> Class Diagram for Audio Service	15
<b>Figure 6:</b> Class Diagram for Authentication Service	16
<b>Figure 7:</b> Activity Diagram	17
<b>Figure 8:</b> Sequence Diagram for the note generation process	17
<b>Figure 9:</b> First part of Figure 8	18
<b>Figure 10:</b> Second part of Figure 8	19
<b>Figure 11:</b> Onboarding Screen	20
<b>Figure 12:</b> Login Screen	20
<b>Figure 13:</b> Register Screen	21
<b>Figure 14:</b> Folder View with Various Notes	21
<b>Figure 15:</b> Generate Note Screen	22
<b>Figure 16:</b> Manually Generate Flashcard Screen	22
<b>Figure 17:</b> Front of a Flashcard Screen	23
<b>Figure 18:</b> Back of a Flashcard Screen	23

## List Of Tables

<b>Table 1:</b> Factors that can affect analysis and design	24
<b>Table 2:</b> Risks and Alternatives	25
<b>Table 3:</b> List of work packages	26

# Analysis and Requirement Report

Project Short-Name: CapyNotes

## 1 Introduction

Taking meaningful, comprehensive notes during the lectures can be a compelling task for many individuals, often diverting their attention from the learning activity. Additionally, the notes taken can be disorganized or insufficient at times. The process of note-taking remains a challenge for students, educators and any learner. In response to this, we introduce CapyNotes, an innovative note generation application designed to supply the user with effective lecture notes. Recognizing the difficulties faced by its target audience, CapyNotes allows the users to effortlessly upload or record audio lectures, transforming them into handwritten-like notes. Going beyond simple summarization, the application goes a step further by identifying keywords, definitions, and constructing flashcard sets to reinforce understanding. CapyNotes also has the capability to generate tree schemes that outline and categorize topics within the content. In any type of setting, whether in class, business meetings, or media tracking tasks, CapyNotes adapts to diverse applications. All in all, with CapyNotes, the goal is to enable our users to focus on the essence of the content while the application takes care of the challenges of note taking.

The type of innovation that CapyNotes exhibits is primarily a process innovation as the objective of the application is to streamline the traditional note-taking process for the users. CapyNotes can be considered innovative not only because it introduces a new method for the conventional manual approach, but also by incorporating modern technologies and redefining the way they are utilized with educational content. It combines techniques such as speech-to-text conversion, text summarization, and more in order to offer a new way of note-taking.

## 2 Current System

Although CapyNotes provides a comprehensive note generation service to the users, the services it offers are neither completely unique nor unprecedented. There are a number of applications in the market provided for different sorts of platforms such as web and mobile. These applications may have a part of services that CapyNotes offer. For instance, FireFlies provides a real-time transcription of dialogues for especially online meetings. As another example, Otter serves as a meeting summarizer and live summary provider resembling FireFlies.

On the other hand, CapyNotes not only focuses on the speech-to-text and summarization process but also on the structure of the generated note. The main purpose of the CapyNotes is to make the content of the created note much remarkable and memorable. The detailed comparison of the CapyNotes with the prospective competitors in the market is given in the following figure (see Figure 1).

FEATURES	CAPYNOTES	STUDY FETCH	OTTER	FIREFLIES
Create Lecture/Meeting Notes	✓	✓	✓	✓
Notes from Already Recorded Lectures	✓	✓	✓	✓
Listen the Current Lecture	✓	✗	✓	✓
Create Keywords	✓	✓	✓	✓
Create Summary	✓	✓	✓	✓
Create Flashcards	✓	✓	✗	✗
Timestamped Note Taking	✓	✗	✗	✓
Integration with Another Applications (e.g Zoom)	✗	✗	✓	✓
Create Schemas	✓	✗	✗	✗
Chatbot	✗	✓	✓	✗

**Figure 1:** Comparison of CapyNotes and its competitors in the market

### 3 Proposed System

In this section, the proposed system will be discussed in detail.

#### 3.1 Overview

CapyNotes is a note generation application that addresses the problems related to efficient note-taking during lectures, meetings, and multimedia content. Unlike the traditional note-taking process, CapyNotes assists users by allowing them to effortlessly upload audio recordings, transforming them into structured and comprehensive notes. Instead of taking notes during lectures, users can focus on the content, while CapyNotes handles the challenges of note taking.

CapyNotes will be a cross-platform application which can be accessed and used from mobile devices, tablets, and from the web. Users can provide both audio recordings and video links and initiate the note generation process. Additionally, convenience of uploading supplementary materials such as lecture slides or images from whiteboards. The application's capabilities include automatic transcription of the audio content, which will be presented to the user alongside the corresponding audio.

Following transcription, the summarization process will take place, generating concise and well-structured notes. Keywords within the content will be identified in order to provide corresponding definitions or explanations. Users then will be presented an option to request flashcards which are automatically generated from key concepts and utilize these as study material. CappyNotes provides convenience by incorporating timestamps within the generated notes, enabling navigation to specific moments from the audio content. Upon completion of the note generation process, users may save notes in various formats such as PDF or plain text. The application's organizational features support users in managing their notes easily by allowing creation of distinct folders for different classes, contexts, or topics. Also, transcriptions and flashcard sets belonging to an audio content will be available through the specific audio listed. Lastly, the users will be able to access their notes (previously generated) in the offline mode, as well.

Overall, CappyNotes brings a new perspective to note-taking, letting users focus on content while conducting the whole process in the background. With cross-platform accessibility and powerful features, CappyNotes offers a user-centric solution for efficient and accessible note generation.

### 3.2 Functional Requirements

Outlined below are the functional requirements for our note-taking application, with specific features marked as "(Additional)" that may be incorporated if time and resources allow for further enhancements.

- **Input Support:** The application must accept both audio recordings and video links as input, and it should allow users to upload supplementary materials such as lecture slides or images from whiteboards.
- **Transcription:** The system must be able to automatically transcribe the audio content.
- **Summarization:** Using the aforementioned transcript it should generate summarized notes with clear subsections for better organization and understanding.
- **Keyword Detection and Definitions:** The application needs to identify keywords within the content and provide automatic definitions or explanations for these keywords, enhancing the comprehensibility of the notes.
- **Flashcards:** The application should provide users with the option to manually create flashcards from keywords and key concepts. Additionally, it should automatically generate flashcards from the summaries it generates, facilitating effective study.
- **Cross-Referencing:** Users should be able to create links or references between related notes within the same domain or context, facilitating easy navigation and understanding of interconnected content.
- **Timestamps:** The application should incorporate timestamps within the generated notes to enable users to pinpoint specific moments in the audio or video content. This feature enhances navigation and retrieval of precise content segments.
- **Output Formats:** Users must have the flexibility to choose and save their notes in various formats, such as PDF, plain text, or other common formats, to cater to their preferences.
- **Organizational Features:** The application should support the organization of notes by enabling users to create distinct folders for different classes or contexts, simplifying the management of notes within these folders.
- **Search and Filters:** The application should provide efficient search and filtering options, enabling users to quickly find specific notes and content based on various

- criteria such as date, keywords, content type, or folders. This streamlines access to relevant information.
- **Offline Mode:** The application should offer an offline mode, allowing users to access and work with their notes without an internet connection. This feature ensures accessibility and usability even in environments with limited or no internet connectivity, which is crucial for students and professionals who may need to review their notes while offline.
  - **Integration with Whiteboard Content (Additional):** The application should be capable of analyzing whiteboard images and seamlessly integrating written information from these images into the notes.
  - **Media Integration (Additional):** The system should offer users the option to seamlessly integrate relevant images found on the web into their notes, enhancing visual content within the notes.

### **3.3 Non-functional Requirements**

#### **3.3.1 Usability**

Offering a user-friendly interface is a high priority since the product itself aims to minimize the users' effort in their note-taking process. In order to achieve this, a convenient user interface that the users will be able to adapt to and operate on seamlessly will be designed. A stable color palette, responsive design and understandable components such as entry fields displaying the correct format are going to be important in this regard. It should be possible for the user to reach the desired pages quickly, thus, the shortest process (in terms of click/action count) for the user will be provided.

#### **3.3.2 Portability**

The application needs to be cross-platform (Android, iOS, Web) as the users should be able to record/upload audio recordings using any device. The system will be designed and implemented in a way that it operates simultaneously on different platforms with small effort, seamlessly. In this regard, we will be utilizing Flutter [1] (cross-platform SDK) and its extensive set of tools to ensure consistent and efficient development across various targeted platforms.

#### **3.3.3 Efficiency**

Processes that take place after the audio recording is uploaded relies on the API requests. Considering this, we will target the minimum number of API requests sent throughout the note generation process. The user interface must continue functioning with satisfactory performance even when handling audio that is large in size.

#### **3.3.4 Maintainability**

Within the application's flow, data will be exposed to a consecutive set of processes such as transcription of the audio, followed by text summarization. Throughout the whole process, API configurations and model parameters may change with varying requirements and for optimization purposes. Thus, the system should be capable of handling modifications seamlessly. To address maintainability, we will try to implement clear interfaces and design an architecture that incorporates loose coupling. Adaptation

of the code into different platforms should be done with minimal effort, and Flutter supports the system in that regard.

### 3.3.5 Scalability

The application should be designed to scale easily to accommodate a growing number of users and increased data processing requirements without compromising performance and user experience. The system should handle a large number of API requests being sent and received simultaneously. To achieve scalability, the microservices architecture will be utilized as efficient resource allocation is desired.

### 3.3.6 Security

Within the application, sensitive information will be requested and collected, occasionally. User credentials and audio recordings are among the most significant of these. To ensure the data's privacy/security, user credentials must be stored in the database in an encrypted fashion. Also, access to sensitive data of another user should be limited. Appropriate access control and authentication mechanisms will be used alongside secure storage systems, for this purpose.

## 3.4 Pseudo Requirements

- Frontend of the application will be implemented in Flutter as it is a cross-platform framework [1]. We have targeted Android, iOS and Web as platforms for our application.
- Backend of CappyNotes will be implemented in Java Spring Boot [2] along with Python [3] (for speech to text and text summarization features).
- Github [4] will be used for version control and Jira [5] will be used for task management.
- We plan to collaborate with a company called Voiser [6] and use their Transcription API [7] for transcribing the audio recording provided from the user. If no collaboration happens, we will use Open AI's Whisper API [8] for this purpose.
- PostgreSQL [9] will be used to store and handle data.
- Amazon S3 [10] will be used for storing audio files and generated notes.

## 3.5 System Models

### 3.5.1 Scenarios

The scenarios for our application are listed below.

#### Scenario 1: Register / Login

**Actors:** User

**Entry Conditions:** The user is at the starting screen of the app.

**Exit Conditions:** The user accesses the main features of the app.

#### The Flow of Events:

1. The user opens the application and is greeted with a welcome screen.
2. The user selects the option to register.
3. They enter their email address and create a password, then click 'Register'.
4. The application sends a verification email to the provided email address.

5. The user verifies their account through the link in the email.
6. Post verification, the user logs in with their new credentials.
7. The application directs the user to the dashboard where they can start using its features.

### **Scenario 2: Generate Notes from an Audio Recording**

**Actors:** User

**Entry Conditions:** The user is logged in.

**Exit Conditions:** The user saves the generated notes.

**The Flow of Events:**

1. The user selects 'New Note' and chooses 'Upload Audio' or 'Record Audio'.
2. They provide the audio file of the lecture and choose whether they want to generate flashcards from the generated note.
3. The application transcribes the audio into text.
4. Post-transcription, the app automatically summarizes the content into organized notes.
5. The user reviews and optionally edits the notes for clarity.
6. The user saves the notes in their preferred format in a designated folder.
7. If chosen so, the application saves the flashcards generated from the notes.

### **Scenario 3: Review Generated Notes Using Cross-Referencing and Timestamps**

**Actors:** User

**Entry Conditions:** The user has multiple sets of notes.

**Exit Conditions:** The user successfully reviews relevant notes.

**The Flow of Events:**

1. The user opens a set of notes.
2. They notice a topic that relates to another lecture's notes.
3. Using the cross-referencing feature, they navigate through related notes.
4. The user uses timestamps to find specific points in the lecture for detailed review.
5. After reviewing and linking, they save the changes.

### **Scenario 4: Generate Flashcards**

**Actors:** User

**Entry Conditions:** The user wants to create study aids.

**Exit Conditions:** The user saves the generated flashcards.

**The Flow of Events:**

1. The user opens their notes and selects the 'Generate Flashcards' option.
2. The application provides empty cards for the user to fill in (front side and back side).
3. The user reviews and customizes the flashcards.
4. They save the flashcards for later study.

### **Scenario 5: Study Flashcards**

**Actors:** User

**Entry Conditions:** The user has previously created flashcards.

**Exit Conditions:** The user completes a study session.

**The Flow of Events:**

1. The user accesses the 'Flashcards' section.
2. They select a set of flashcards to study.
3. The application presents the flashcards in a study session format.
4. The user reviews each flashcard, flipping them to see definitions or explanations.
5. After completing the session, they exit.

### **Scenario 6: Utilizing Offline Mode and Search Functionality**

**Actors:** User

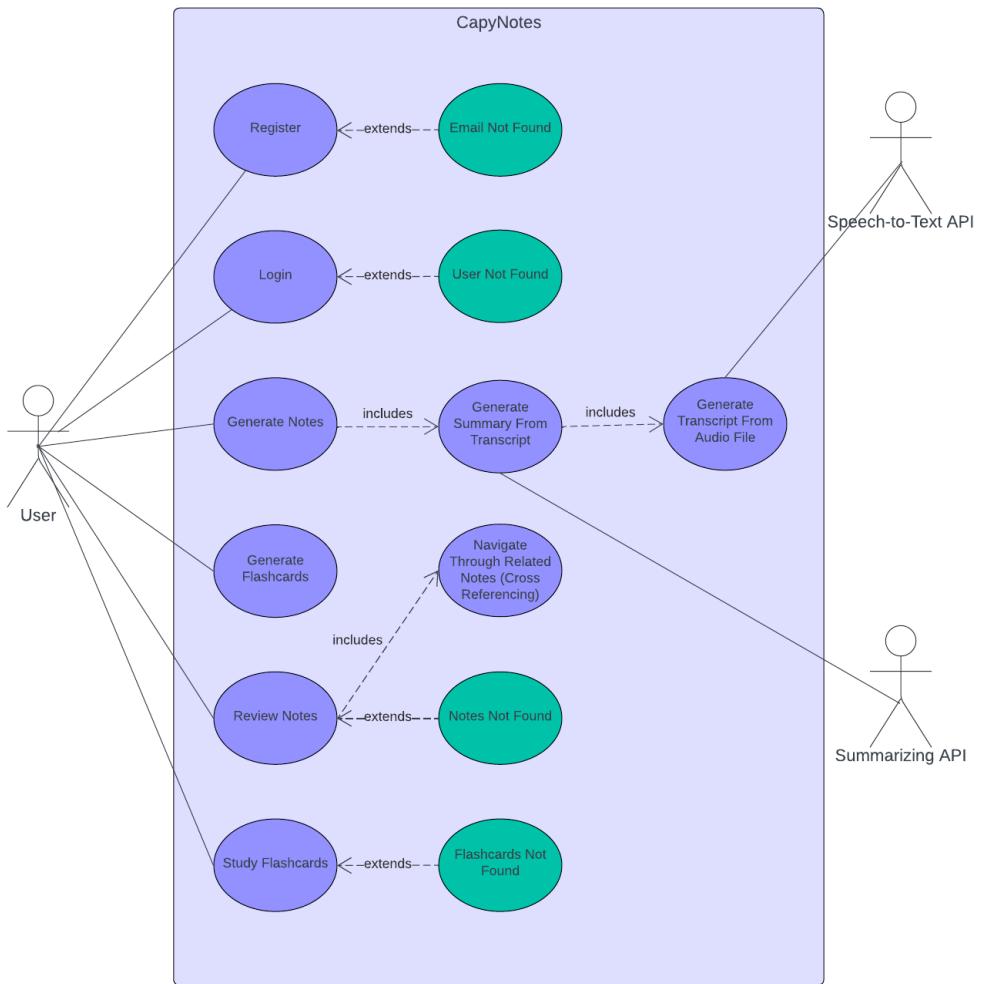
**Entry Conditions:** The user needs to access notes without internet connectivity.

**Exit Conditions:** The user finds and reviews the necessary notes.

**The Flow of Events:**

1. The user opens the app in offline mode.
2. They navigate to the search bar to locate specific notes.
3. Entering keywords, the student filters the notes by date, type, or content.
4. The application displays all relevant notes.
5. The user selects and reviews the necessary notes using embedded timestamps for reference.
6. After reviewing, the student closes the app.

### 3.5.2 Use-Case Model



**Figure 2:** Use Case Diagram of CappyNotes

The use case diagram of our application describes the actions our users, and external actors can take by using our application. Our primary actor depicted as the “User” can register with a legitimate email, login with their registered account, generate notes from audio recordings, generate flashcards, review their generated notes, and study their flashcards.

The external actors of our application are the “Speech-to-text API” and the “Summarizing API”. The “Speech-to-text API” generates the transcript from the audio file to be used by the “Summarizing API”, and the “Summarizing API” generates the summary of the transcript for our application to mold it into the generated notes.

### 3.5.3 Class Diagrams

#### 3.5.3.1 Note Generation Service

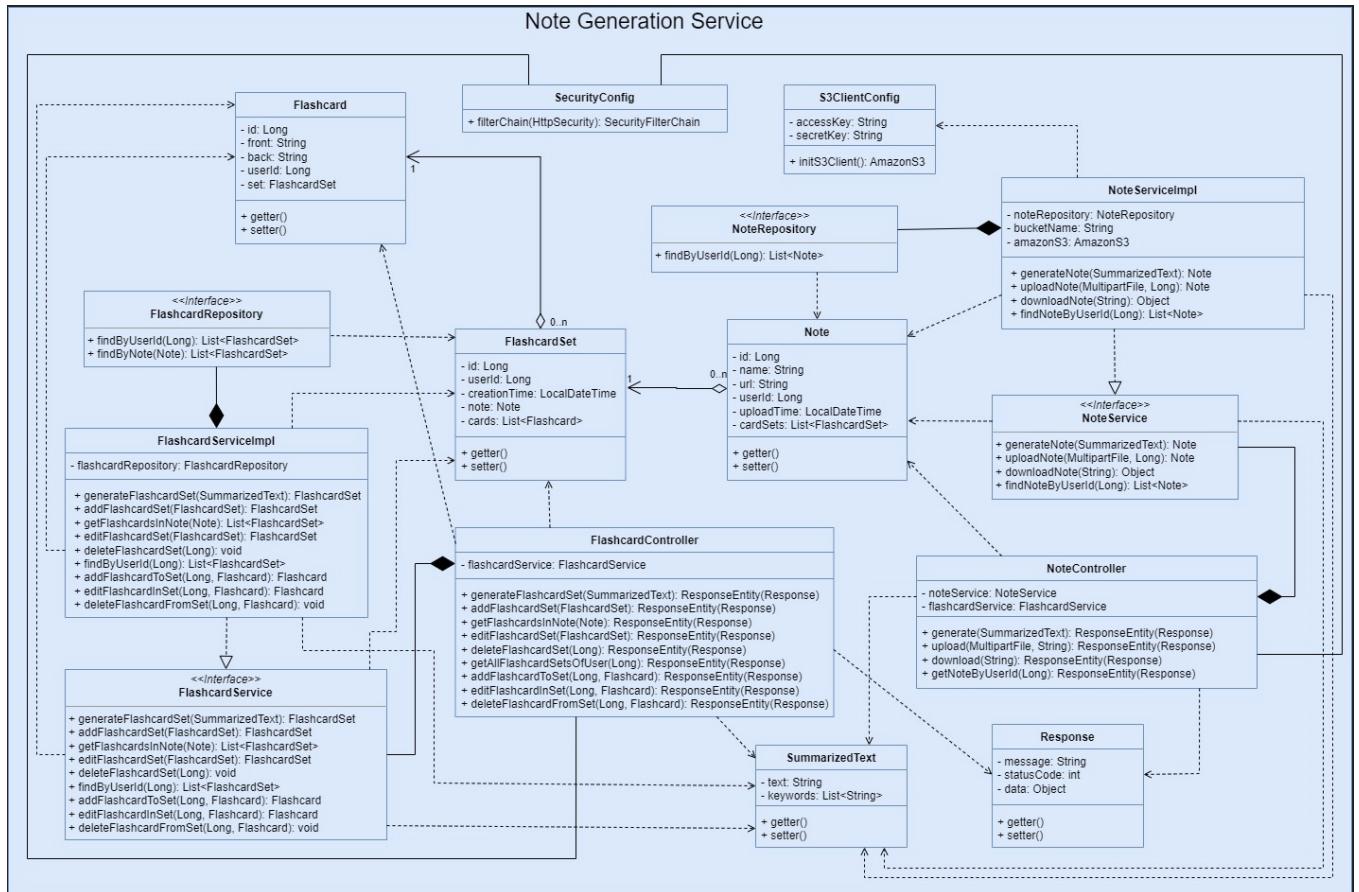
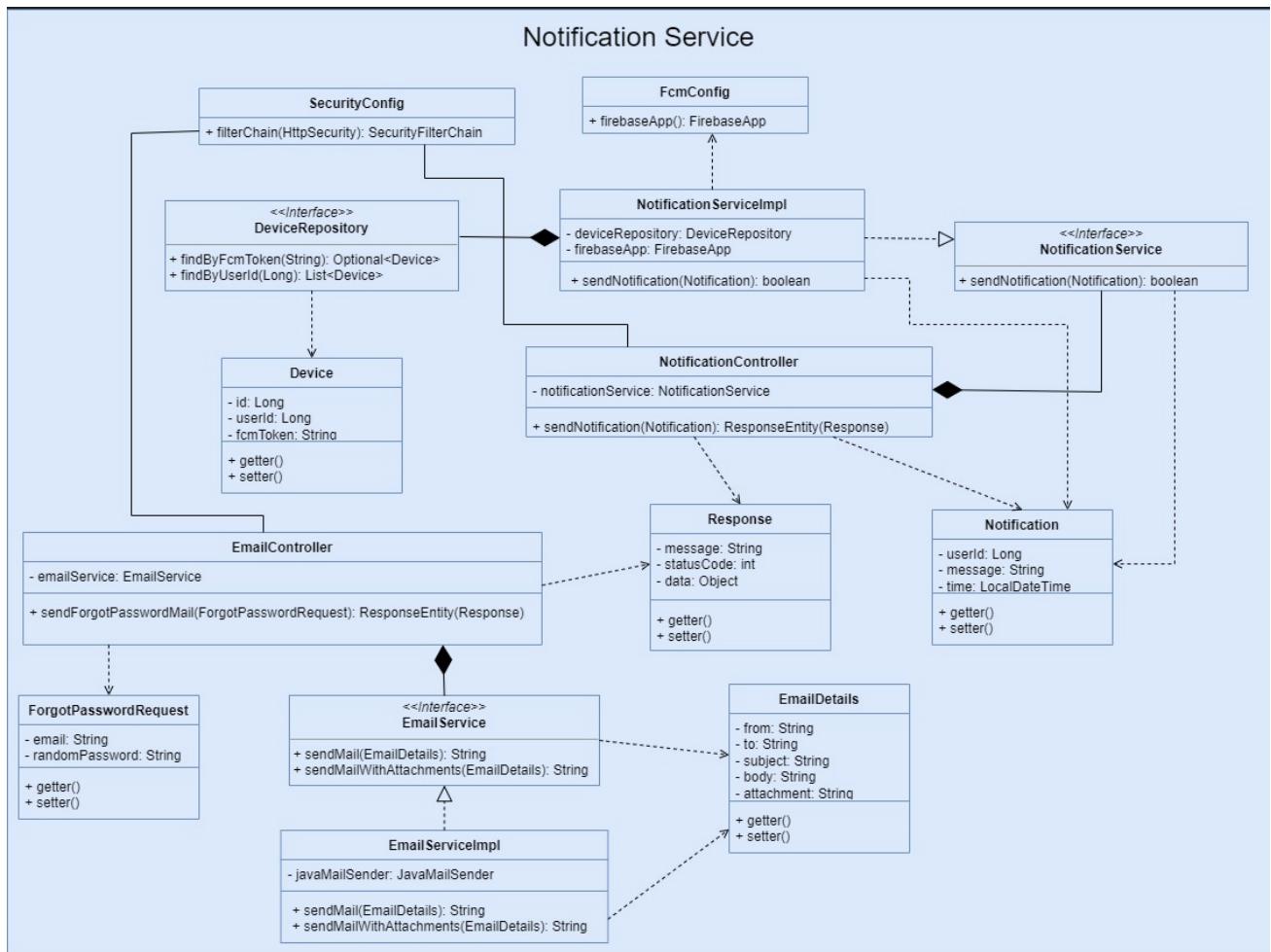


Figure 3: Class Diagram for Note Generation Service ([Click here](#) for better quality)

### 3.5.3.2 Notification Service



**Figure 4:** Class Diagram for Notification Service ([Click here](#) for better quality)

### 3.5.3.3 Audio Service

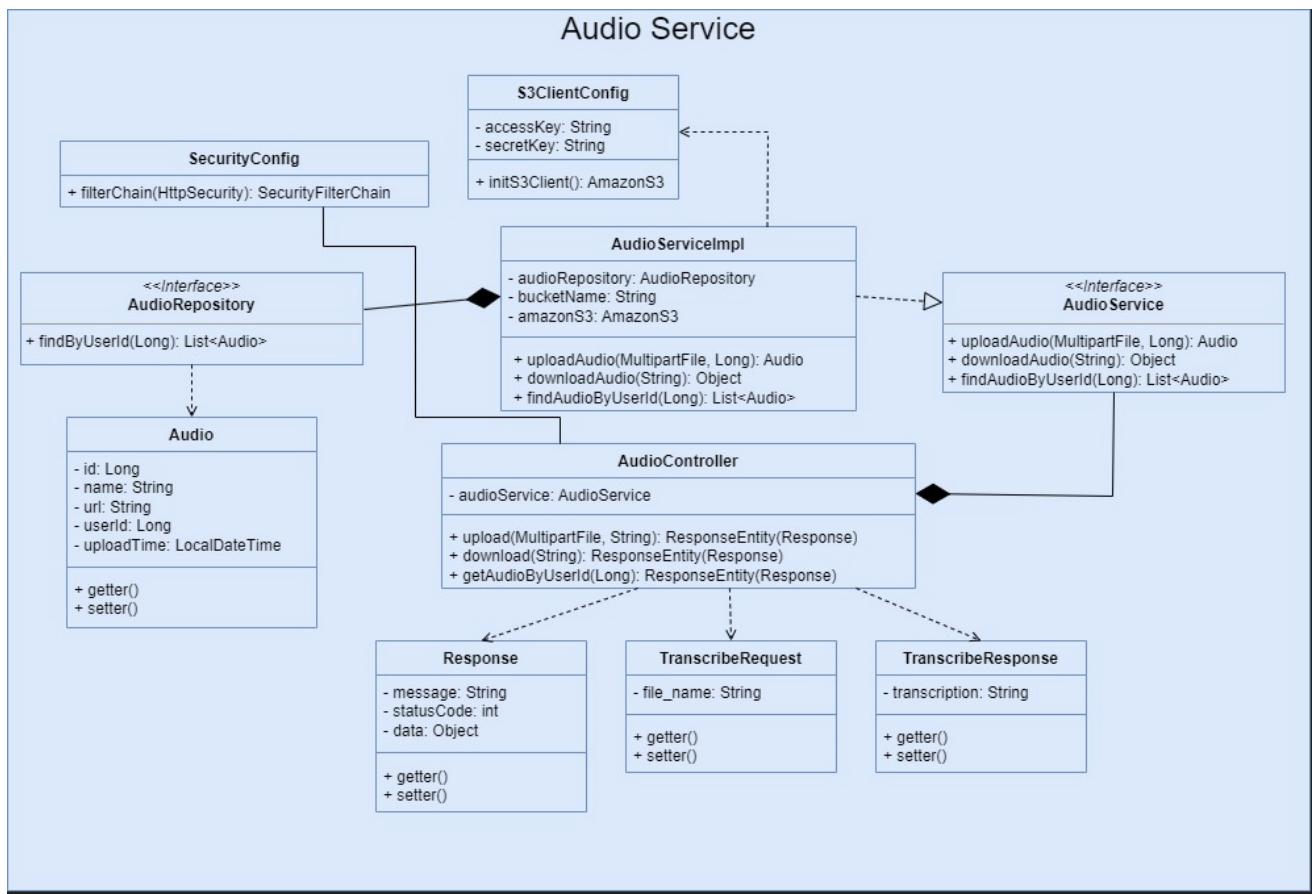
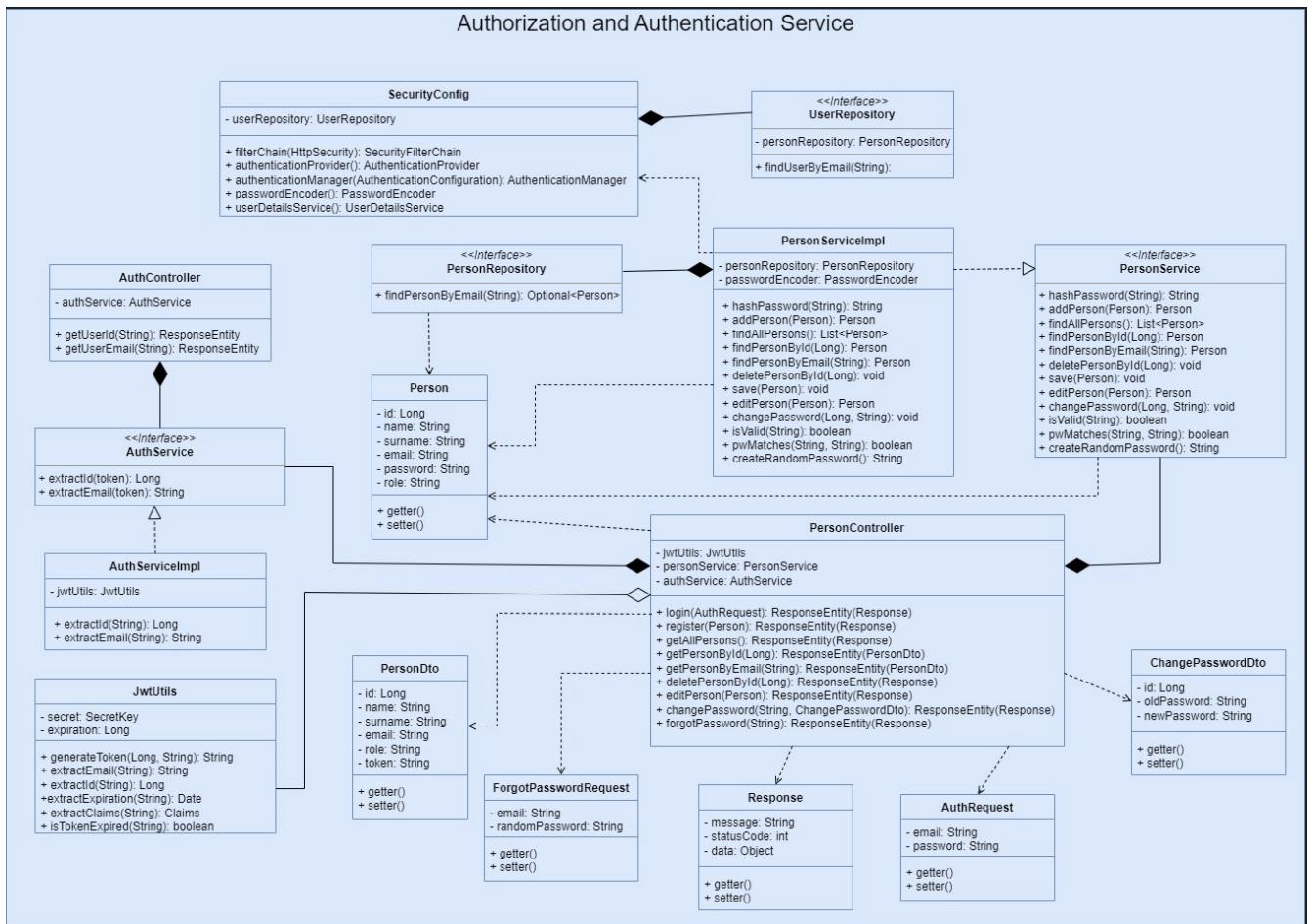


Figure 5: Class Diagram for Audio Service ([Click here](#) for better quality)

### 3.5.3.4 Authentication Service



**Figure 6:** Class Diagram for Authentication Service ([Click here](#) for better quality)

### 3.5.4 Dynamic Models

#### 3.5.4.1 Activity Diagram

Provided below, is the activity diagram constructed for the main use case of CappyNotes: Note generation. The activities (or actions) include *upload recording*, *record audio*, *enter video link*, *save note* (User), *transcribe speech*, *timestamping* (Speech-to-text Service), *summarize text*, *identify keywords*, *generate tree schemes*, *attach definitions* (Summarizing Service), and *generate note* (Note generating Service). The diagram contains all the steps included in note generation, from the user providing the audio input to the generation and saving of the note.

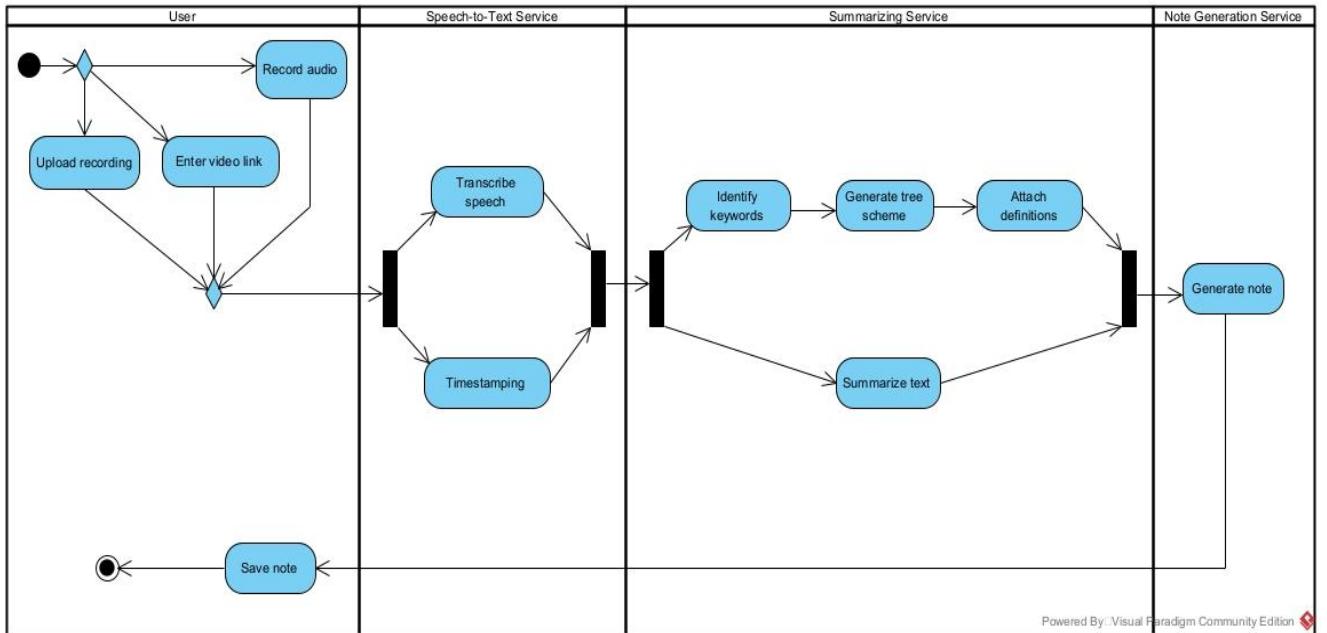


Figure 7: Activity Diagram

### 3.5.4.2 The Sequence Diagram

In the following sequence diagram, a note generation process in CapyNotes is depicted. This process includes transcript, summary, flashcard, chapter and chart generations. Firstly, the transcript from the provided audio by the user is generated. After that, the summary is generated by utilizing the recently formed transcript. Based on the summary; flashcards, chapters(subsummaries) and charts are generated.

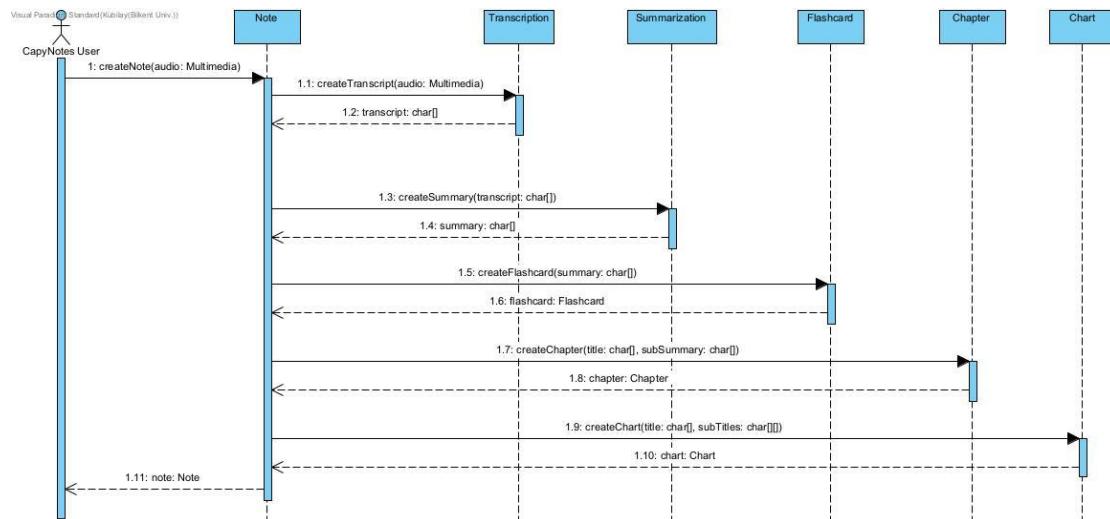
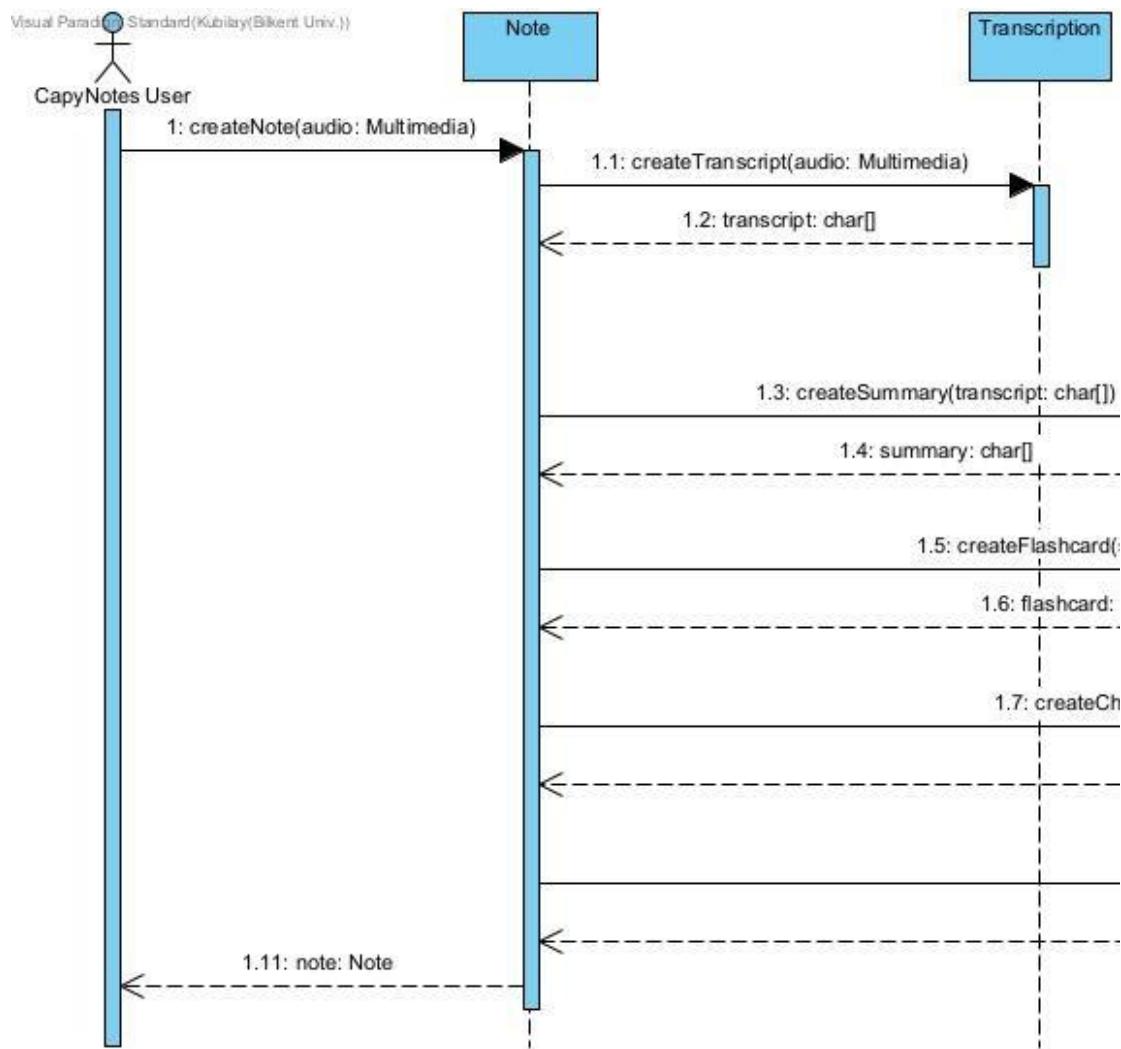
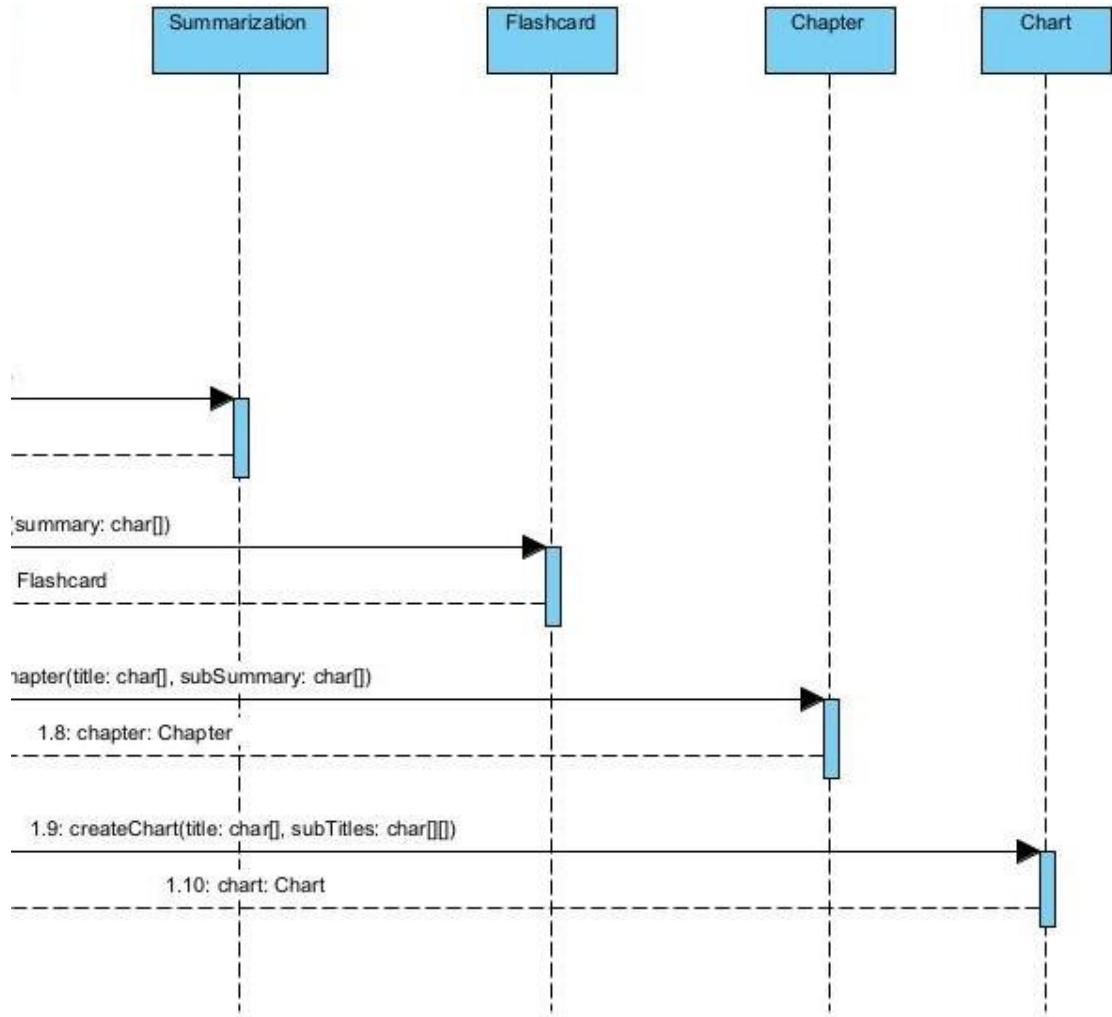


Figure 8: Sequence Diagram for the note generation process



**Figure 9:** First part of Figure 8



**Figure 10:** Second part of Figure 8

### 3.5.5 User Interface Mock-Ups

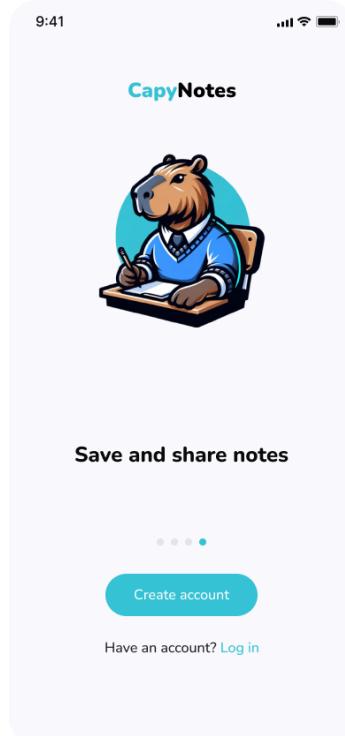


Figure 11: Onboarding Screen

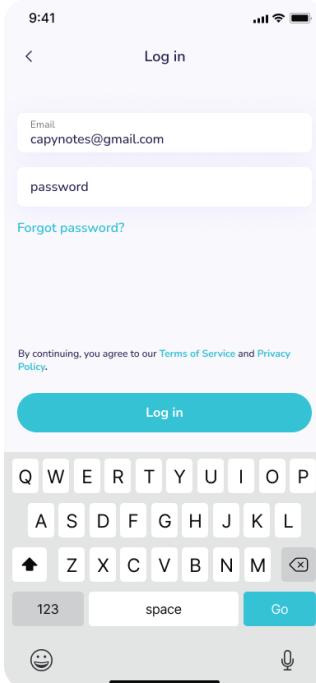
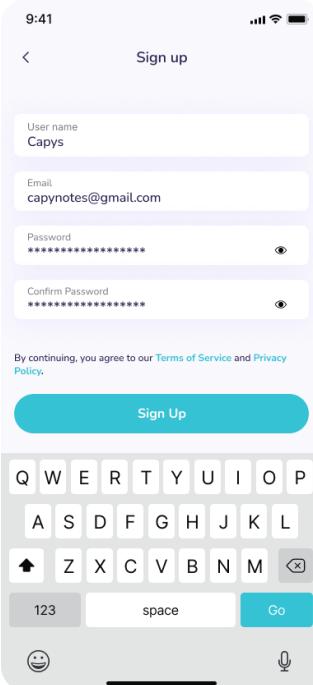
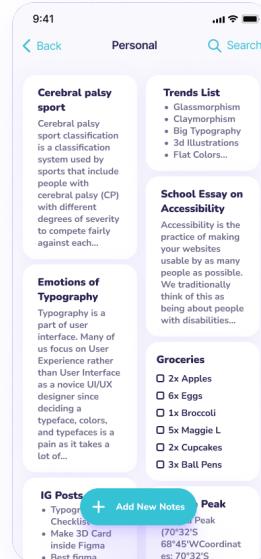


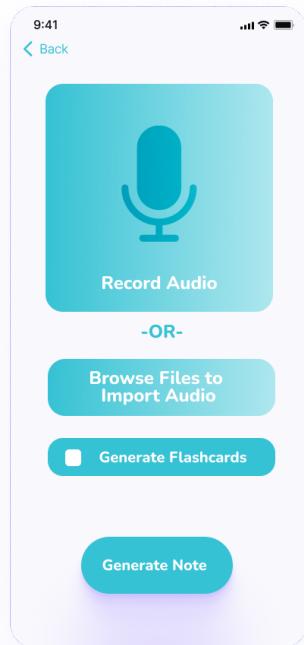
Figure 12: Login Screen



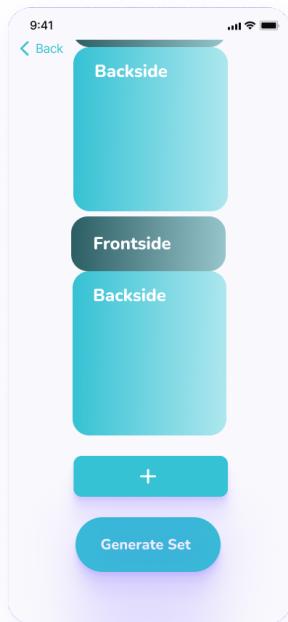
**Figure 13:** Register Screen



**Figure 14:** Folder View with Various Notes



**Figure 15:** Generate Note Screen



**Figure 16:** Manually Generate Flashcard Screen

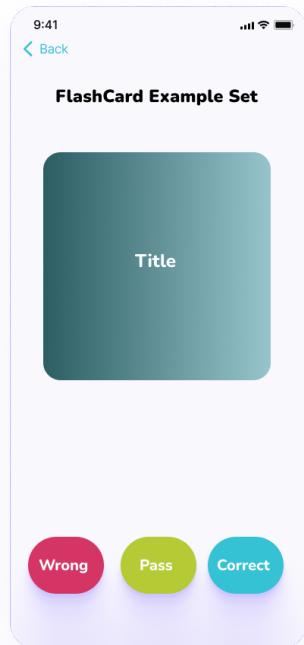


Figure 17: Front of a Flashcard Screen



Figure 18: Back of a Flashcard Screen

## 4 Other Analysis Elements

### 4.1 Consideration of Various Factors in Engineering Design

CapyNotes utilizes various types of technologies while providing a comprehensive automated note generation application. These technologies include speech-to-text APIs, summarization models, a cross platform application development framework and a database management system. Optimizing the compatibility between diverse technologies is an engineering problem to address. The engineering design for this problem should be planned so that the whole process in the application is smooth and fast. Moreover, the application ought to be user-friendly across all platforms it is provided on, ensuring a high-quality user experience for users when switching between platforms.

#### 4.1.1 Constraints

**Table 1:** Factors that can affect analysis and design.

Factors	Effect level	Effect
Global factors	Low	May affect student's habit of following lectures around the world or may not.
Cultural factors	None	This note-taking application is unlikely to have an impact on any cultural factors.
Social factors	Medium	More people might be encouraged to actively participate in the meetings they are attending instead of spending time over taking notes about topics discussed.
Environmental factors	Medium	It is estimated that the services used by the application have a similar amount of carbon footprint to the current digital note-taking methods.
Economic factors	High	The various services from different suppliers may lead to high-rate pricing policies.

#### 4.1.2 Standards

Throughout the analysis, design and development processes we have followed international engineering standards to ensure the product we develop is professional. For example, this report follows IEEE 12207 standard. We are also applying these standards to prepare ourselves for the professional work environment that lies ahead of us.

### 4.2 Risks and Alternatives

#### Risks:

CapyNotes, due to its nature and the problems it addresses, contains some risks.

1. The algorithms and APIs that will be utilized for the speech-to-text transcription and summarization purposes may occasionally produce results that unprecedently fall short of the targeted accuracy/correctness. This is due to the fact that, whichever method is chosen, there will always be an error margin involved.
2. An audio recording might be uploaded to the system for the note generation process without the consent of the speaker. Here, the underlying risk is the potential usage and processing of the audio (and the speaker's voice) by the third parties, which requires ethical consideration.
3. While the application provides flexibility and usability by offering the user the option to record an audio using their phone or other unprofessional equipment, this might mean that the note generation process could be initiated by transcribing low quality audio. This might result in poor accuracy and undesired outcomes.

**Alternatives:**

The main objective of the application is to provide the user with notes generated from the uploaded audio recording. However, the risks mentioned above should be addressed within the flow of the application, in some way. Regarding this, as the designers and developers of such an application, we have to make sure of the following:

1. Sufficient effort is put into research regarding the accuracy and fault tolerance of the underlying algorithms and APIs. Tradeoffs between options are assessed carefully (key metrics to consider: Word Error Rate [11], Signal-to-Noise Ratio [12], Accuracy Across Accents [13]).
2. Important information is provided to the user prior to usage. Consent regarding the privacy of the audio is taken to verify that the recording is allowed to be used and processed.
3. Users are informed about the possible outcomes of providing low quality audio.

Additionally, all development activities must be conducted considering factors such as time efficiency, cost efficiency, privacy, and safety.

**Table 2:** Risks and Alternatives

Risk	Likelihood	Effect on the project	B Plan Summary
Algorithm/API accuracy shortfall	Low	High	Research algorithm accuracy; consider tradeoffs using metrics.
Privacy concerns with audio uploads	Medium	Low	Obtain user consent, inform about privacy, implement safeguards.
Low-quality audio upload	Medium	Medium	Educate users on outcomes, implement measures for accuracy.

### 4.3 Project Plan

**Table 3:** List of work packages

WP#	Work package title	Leader	Members involved
WP1	UI Design	İsmail Emre Deniz	Can Ersoy, İsmail Emre Deniz
WP2	Project Website	Can Ersoy	Can Ersoy, İsmail Emre Deniz
WP3	Authentication Frontend	Can Ersoy	Can Ersoy, İsmail Emre Deniz
WP4	Authentication Backend	Berk Temel	Berk Temel, Kemal Kubilay Yılmaz
WP5	Speech To Text Service	Mustafa Burak Erkoçak	Mustafa Burak Erkoçak, Kemal Kubilay Yılmaz
WP6	Get Audio From User	Can Ersoy	Can Ersoy, İsmail Emre Deniz
WP7	Notification Feature's Frontend	İsmail Emre Deniz	Can Ersoy, İsmail Emre Deniz
WP8	Notification Feature's Backend	Berk Temel	Berk Temel, Mustafa Burak Erkoçak
WP9	Summarization Feature Backend	Kemal Kubilay Yılmaz	Kemal Kubilay Yılmaz, Mustafa Burak Erkoçak
WP10	Flashcards Feature Frontend	Can Ersoy	Can Ersoy, İsmail Emre Deniz
WP11	Flashcard Feature Backend	Berk Temel	Berk Temel, Kemal Kubilay Yılmaz
WP12	Caching Past Notes	İsmail Emre Deniz	Can Ersoy, İsmail Emre Deniz
WP13	Tree Scheme Generation	Kemal Kubilay Yılmaz	Can Ersoy, İsmail Emre Deniz, Kemal Kubilay Yılmaz, Mustafa Burak Erkoçak
WP14	Note PDF Generation	Mustafa Burak Erkoçak	Kemal Kubilay Yılmaz, Mustafa Burak Erkoçak, Berk Temel

<b>WP 1: UI Design</b>
<b>Start date:</b> 12.11.2023 <b>End date:</b> 19.11.2023
<b>Leader:</b> İsmail Emre Deniz <b>Members involved:</b> Can Ersoy
<b>Objectives:</b> Designing the User Interface of the app
<b>Tasks:</b>
<b>Task 1.1 Determining the flow of the app:</b> Making decisions about how the app should function and the navigation between the pages
<b>Task 1.2 Figma Design:</b> Designing the User Interface in Figma
...
<b>Deliverables</b>
<b>D1.1:</b> User Interface Design in Figma

<b>WP 2: Project Website</b>		
<b>Start date:</b> 13.10.2023 <b>End date:</b> 01.05.2023		
Leader:	Can Ersoy	Members involved:
<b>Objectives:</b> Having a functional website with all of the reports		
<b>Tasks:</b> <b>Task 2.1 Website Creation :</b> Implementation of the website in Flutter <b>Task 2.2 Maintenance of the Website :</b> Updating the website when new reports are delivered		
<b>Deliverables</b> <b>D2.1:</b> Working website with reports		
<b>WP3 : Authentication Frontend</b>		
<b>Start date:</b> 1.11.2023 <b>End date:</b> 30.11.2023		
Leader:	Can Ersoy	Members involved:
<b>Objectives:</b> Authentication screens (Login, Register, Forgot Password, Change Password) implementation		
<b>Tasks:</b> <b>Task 3.1 Login Screen Implementation :</b> Completely implementing Login Screen and connecting it to backend API. <b>Task 3.2 Register Screen Implementation :</b> Completely implementing Register Screen and connecting it to backend API. <b>Task 3.3 Forgot Password Screen Implementation :</b> Completely implementing Forgot Password Screen and connecting it to backend API. <b>Task 3.4 Change Password Screen Implementation :</b> Completely implementing Change Password Screen and connecting it to backend API.		
<b>Deliverables</b> <b>D3.1:</b> Login Screen <b>D3.2:</b> Register Screen <b>D3.3:</b> Forgot Password Screen <b>D3.4:</b> Change Password Screen		
<b>WP4 : Authentication Backend</b>		
<b>Start date:</b> 1.11.2023 <b>End date:</b> 28.11.2023		
Leader:	Berk Temel	Members involved:
<b>Objectives:</b> Authentication Service implementation		
<b>Tasks:</b> <b>Task 4.1 Login Implementation :</b> Completely implementing backend of login feature. <b>Task 4.2 Register Implementation :</b> Completely implementing backend of register feature. <b>Task 4.3 Forgot Password Implementation :</b> Completely implementing backend of forgot password feature. <b>Task 4.4 Change Password Implementation :</b> Completely implementing backend of change password feature.		
<b>Deliverables</b> <b>D4.1:</b> Backend of login <b>D4.2:</b> Backend of register <b>D4.3:</b> Backend of forgot password <b>D4.4:</b> Backend of change password		

<b>WP5 : Speech To Text Service</b>
<b>Start date:</b> 20.11.2023 <b>End date:</b> 15.12.2023
<b>Leader:</b> Mustafa Burak Erkoçak <b>Members involved:</b> Kemal Kubilay Yılmaz
<b>Objectives:</b> Researching and testing various speech to text API's and comparing the results. Deciding on the suitable alternative and implementing a Flask service.
<b>Tasks:</b>
<b>Task 5.1 Research of Speech To Text APIs:</b> Researching and testing various speech to text API's and comparing the results.
<b>Task 5.2 Implementation of a Flask service :</b> Implementing a backend service which takes an audio file as an input and outputs the transcription of it.
<b>Deliverables</b>
<b>D5.1:</b> Backend service for transcription of the audio (with Flask)
<b>WP6 : Get Audio From User</b>
<b>Start date:</b> 01.12.2023 <b>End date:</b> 01.01.2024
<b>Leader:</b> Can Ersoy <b>Members involved:</b> İsmail Emre Deniz
<b>Objectives:</b> Implementing both audio file picking and instant recording features
<b>Tasks:</b>
<b>Task 6.1 Implementing Audio File Picking Feature :</b> Allow the user to pick an already recorded audio from their device and send it to backend.
<b>Task 6.2 Implementing Instant Audio Recording Feature:</b> Allow the user to record an audio instantly from their device and send it to backend.
<b>Deliverables</b>
<b>D6.1:</b> Frontend with working audio file picking feature
<b>D6.2:</b> Frontend with working audio recording feature
<b>WP7 : Notification Feature's Frontend</b>
<b>Start date:</b> 07.01.2024 <b>End date:</b> 21.01.2024
<b>Leader:</b> İsmail Emre Deniz <b>Members involved:</b> Can Ersoy
<b>Objectives:</b> Researching and configuring the application to receive notifications from server
<b>Tasks:</b>
<b>Task 7.1 Researching Push Notifications:</b> Researching how to implement push notifications with Flutter and Firebase
<b>Task 7.2 Configuring The App:</b> Implementing push notifications and integrating the app with Firebase accordingly
<b>Deliverables</b>
<b>D7.1:</b> Application can successfully receive push notifications from the server
<b>WP8 : Notification Feature's Backend</b>
<b>Start date:</b> 07.01.2024 <b>End date:</b> 21.01.2024
<b>Leader:</b> Berk Temel <b>Members involved:</b> Mustafa Burak Erkoçak
<b>Objectives:</b> Researching and implementing backend to send notifications to users
<b>Tasks:</b>
<b>Task 8.1 Researching Push Notifications:</b> Researching how to implement push notifications with Java Spring Boot and Firebase
<b>Task 8.2 Implementing Notification Service:</b> Implementing Notification Service so it can send push notifications when required

<b>Deliverables</b>
<b>D8.1:</b> Notification Service can successfully send push notifications
<b>WP9 :</b> Summarization Feature Backend
<b>Start date:</b> 10.12.2023 <b>End date:</b> 01.03.2024
<b>Leader:</b> Kemal Kubilay Yilmaz <b>Members involved:</b> Mustafa Burak Erkoçak
<b>Objectives:</b> Researching and testing various summarization models and comparing the results. Fine tuning model to make it more suitable for our purpose. Implementing a Flask service.
<b>Tasks:</b>
<b>Task 9.1 Research of Summarization Models :</b> Researching and testing various summarization models.
<b>Task 9.2 Implementation of a Flask Service :</b> Implementing a backend service which takes a text as an input and outputs the summary of it as well as identifying keywords and definitions.
<b>Deliverables</b>
<b>D9.1:</b> Backend service for summarization and keyword identification of the text (with Flask)
<b>WP10 :</b> Flashcards Feature Frontend
<b>Start date:</b> 07.01.2023 <b>End date:</b> 28.01.2024
<b>Leader:</b> Can Ersoy <b>Members involved:</b> Ismail Emre Deniz
<b>Objectives:</b> Allow the user to create manual flashcard sets, implement flashcard set practice screens
<b>Tasks:</b>
<b>Task 10.1 Manual Flashcard Set Creation Screens:</b> Implement Create Flashcard Set Screen and let the user create flashcard sets manually.
<b>Task 10.2 Implementation of Flashcard Practice Screen:</b> Completing implementing flashcard set practice screens
<b>Deliverables</b>
<b>D10.1:</b> Create Flashcard Set Screen with connection to backend.
<b>D10.2:</b> Flashcard Set Practice Screen with connection to backend.
<b>WP11 :</b> Flashcards Feature Backend
<b>Start date:</b> 07.01.2024 <b>End date:</b> 28.01.2024
<b>Leader:</b> Berk Temel <b>Members involved:</b> Kemal Kubilay Yilmaz
<b>Objectives:</b> Implementing Flashcards Service in backend such that manual flashcard sets can be created along with generating them automatically from the keywords and definitions found by the summary service.
<b>Tasks:</b>
<b>Task 11.1 Implementation of manually creating flashcard sets:</b> Completing the implementation of generating flashcard sets manually.
<b>Task 11.2 &lt;write the name of the task/subtask&gt; :</b> Completing the implementation of generating flashcard sets from the keywords and definitions found by the summary service.
<b>Deliverables</b>
<b>D11.1:</b> Flashcard Service for manually creating sets
<b>D11.2:</b> Flashcard Service for automatically creating sets
<b>WP12 :</b> Caching Past Notes

<b>Start date:</b> 01.02.2024	<b>End date:</b> 28.02.2024
<b>Leader:</b> İsmail Emre Deniz	<b>Members involved:</b> Can Ersoy
<b>Objectives:</b> Researching and implementing caching feature in frontend so that the user can access generated notes while they are offline.	
<b>Tasks:</b>	
<p><b>Task 12.1 Researching caching or alternatives:</b> Conducting research about how to allow users to access already downloaded notes while they are offline.</p> <p><b>Task 12.2 Implementing offline access:</b> Configuring the app according to the decided method (caching etc.).</p>	
<b>Deliverables</b>	
<p><b>D12.1:</b> Application is configured so that the users can access downloaded notes while they are offline.</p>	
<b>WP13 : Tree Scheme Generation</b>	
<b>Start date:</b> 01.03.2024	
<b>End date:</b> 01.05.2024	
<b>Leader:</b> Kemal Kubilay Yılmaz	<b>Members involved:</b> Can Ersoy, İsmail Emre Deniz, Mustafa Burak Erkoçak
<b>Objectives:</b> Implementing tree scheme generation based on the given keywords and summary.	
<b>Tasks:</b>	
<p><b>Task 13.1 Researching Ways to Generate Tree Scheme :</b> Researching how tree schemes can be generated in a PDF.</p> <p><b>Task 13.2 Implementing Tree Scheme Generation :</b> Implementing the found ways to generate tree schemes.</p>	
<b>Deliverables</b>	
<p><b>D13.1:</b> A method whose functionality is generating tree schemes based on the keywords.</p>	
<b>WP14 : Note PDF Generation</b>	
<b>Start date:</b> 11.12.2023	
<b>End date:</b> 01.05.2024	
<b>Leader:</b> Mustafa Burak Erkoçak	<b>Members involved:</b> Kemal Kubilay Yılmaz, Berk Temel
<b>Objectives:</b> Generating a well organized PDF with summarized text and generated tree schemes.	
<b>Tasks:</b>	
<p><b>Task 14.1 Researching Libraries Used for PDF Generation :</b> Researching existing libraries that can be used for generating PDFs.</p> <p><b>Task 14.2 Implementing PDF Generation :</b> Implementing PDF generation with summarized text, keywords and definitions, and generated tree schemes using the researched libraries.</p>	
<b>Deliverables</b>	
<p><b>D14.1:</b> Service for PDF generation</p>	

#### 4.4 Ensuring Proper Teamwork

As a project team of 5, ensuring proper teamwork is crucial for the success of the project. The priority is to have consistent and clear communication channels so that each member can be on the same page regarding the state of the project and the workload. It

is also important to receive contributions from all of the members. To ensure this, the members are separated into two teams, which are the frontend team and the backend team. Even though this separation doesn't clearly define the set of tasks each team undertakes, it could be stated that the focus of the frontend team is to implement the user interface, while the backend team deals with the implementation of required services. Also, it is worth mentioning that in the later stages of the project, members may assist the other team, if required. The teams are provided below:

**Frontend Team:**

- Can Ersoy
- İsmail Emre Deniz

**Backend Team:**

- Berk Temel
- Kemal Kubilay Yılmaz
- Mustafa Burak Erkoçak

As mentioned previously, having proper communication channels have implications for the success of the project. Thus, before anything else, these channels were determined. For scheduling meetings and asynchronous communication, WhatsApp (messaging application) is being used. For conducting online meetings and discussions, Discord platform will be used. Online meetings will be scheduled upon requests, with the attendance of members that are concerned with the meeting topic. Weekly face-to-face meetings will be held with the participation of all team members. Topics such as the status of the project, incoming deadlines, task distribution, implementation details and future plans will be discussed in detail. JIRA (task management software) is also a crucial component of our communication as each member can track the To Do/In Progress/Testing/Done tasks of the current sprint and active tasks assigned to each team member.

Overall, division of responsibilities into frontend and backend teams, coupled with proper communication channels sets up the 'proper teamwork' within the team. Ideal separation of workloads ensures abstraction and efficient teamwork. On the other hand, utilization of communication and task management platforms enhances the member's grasp of the project throughout its lifecycle.

#### **4.5 Ethics and Professional Responsibilities**

CapyNotes will collect users' information and preferences throughout different stages in order to offer secure authentication and to provide the optimal experience for each user. Such information will consist of user credentials, device information, session data, language preferences and most importantly, audio recordings uploaded by the user. This might result in ethical issues that need to be addressed. First of all, private user information will not be shared with third party applications. Additionally, specific data will be collected only after the user consents to it. That is, within the context of CapyNotes, the users will be warned and asked their confirmation whether the uploaded audio recordings are suited to be used and processed by third parties. All issues mentioned must be handled in compliance with the Turkish law (specifically, The Law on the

Protection of Personal Data) and non-local data protection and privacy regulations such as GDPR [14] [15].

The product's target audience consists of students, instructors and lifelong learners. In this respect, it is important to design the application in a way that is accessible by all of these groups.

#### **4.6 Planning for New Knowledge and Learning Strategies**

In the process of implementing CapyNotes while optimizing its functionality and efficiency, our development strategy is shaped according to a wide range of technologies/tools being used. With this being the situation, the importance of acquiring new skills becomes even more significant. The frontend of CapyNotes will be implemented using Flutter, as it has the capability to provide a consistent user experience throughout different platforms. The frontend team puts significant effort to improve their knowledge on Flutter using sources from the Web while actively implementing the frontend components of the application. The backend frameworks include Java Spring Boot and Flask (Python), and they have different roles in the project. The backend team is involved in continuous research regarding various implementation details such as exploring optimal APIs/algorithms for speech-to-text, text summarization features, and evaluating the scalability of the Java Spring Boot framework. Also, simulating scenarios regarding the use of different APIs like Voicer and Whisper Speech-to-Text before the actual implementation is an important learning strategy we adopted. Lastly, even though each team member had previous experience with database management via PostgreSQL, we try to get involved in the database design and implementation processes altogether for collaborative improvement.

Regarding the planning for learning strategies of technical tools, we try to reflect the progress of our learning onto our task management procedures, as well. That is, we utilize the JIRA Software in keeping track of our tasks, including the learning/research activities. Specifically, our JIRA tasks are labeled according to their category, and one of them is the [Research] category. We strongly believe that keeping track of what knowledge we have obtained and what we should learn/improve in the future will significantly boost our performance.

### **5 Glossary**

**API:** Application Programming Interface

**SDK:** Software Development Kit

**Speech to Text:** The general process of transcribing an audio recording to text.

## 6 References

- [1]“Multi-Platform”. Flutter. <https://flutter.dev/multi-platform>.
- [2]“Why Spring”. Java Spring Boot. <https://spring.io/why-spring>
- [3]“About Python”. Python. <https://www.python.org/about/>
- [4]“Github”. Github. <https://github.com/>
- [5]“Jira Software - Features”. Jira. <https://www.atlassian.com/software/jira/features>
- [6]“Voiser”. Voiser. <https://voiser.net/>
- [7]“Voiser Deşifre”. Voiser. <https://voiser.net/desifre>
- [8]“Introducing Whisper” Whisper. <https://openai.com/research/whisper>
- [9]“What is PostgreSQL?”. PostgreSQL. <https://www.postgresql.org/about/>
- [10]“Cloud Object Storage”. Amazon Web Services.  
[https://aws.amazon.com/s3/?nc1=h\\_ls](https://aws.amazon.com/s3/?nc1=h_ls)
- [11]“Comparing Speech Recognition Systems (Microsoft API, Google API And CMU Sphinx)”. Bohouta, Gamal & Képuska, Veton.Bohouta Gamal .  
[https://www.researchgate.net/publication/314938892\\_Comparing\\_Speech\\_Recognition\\_Systems\\_Microsoft\\_API\\_Google\\_API\\_And\\_CMU\\_Sphinx](https://www.researchgate.net/publication/314938892_Comparing_Speech_Recognition_Systems_Microsoft_API_Google_API_And_CMU_Sphinx)
- [12]“Sound localization in noise: The effect of signal-to-noise ratio”. Michael D. Good, Robert H. Gilkey.  
<https://pubs.aip.org/asa/jasa/article-abstract/99/2/1108/696623/Sound-localization-in-noise-The-effect-of-signal?redirectedFrom=fulltext>
- [13] “Speech Perception and Generalization Across Talkers and Accents”. Kodi Weatherholtz, T. Florian Jaeger.  
<https://oxfordre.com/linguistics/display/10.1093/acrefore/9780199384655.001.0001/acfore-9780199384655-e-95>
- [14]“Kişisel Verilerin Korunması Kanunu”. Kişisel Verilerin Korunması Kanunu.  
<https://www.mevzuat.gov.tr/mevzuat?MevzuatNo=6698&MevzuatTur=1&MevzuatTertip=5>
- [15] “GDPR”. General Data Protection Regulation. <https://gdpr-info.eu>