Bilkent University
Department of Computer Engineering

Senior Design Project

T2326
CapyNotes

Project Specification 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

17.11.2023

This report is submitted to the Department of Computer Engineering of Bilkent University in partial fulfillment of the requirements of the Senior Design Project course CS491/2.

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

List Of Figures

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.

In this report, a detailed description of CapyNotes, its architecture and components, constraints, professional/ethical issues, and adopted standards are provided in the first part. Proceeding with the concepts that construct the basis of our design process, functional and non-functional requirements are mentioned. Afterwards, feasibility discussions are provided, which include the sections “Market & Competitive Analysis”, “Academic Analysis”, “Technical Discussion”, “Effects of the Product”. Lastly, the report is concluded with a brief glossary of technical terms and the references section.

1.1 Description

Many individuals find it challenging to efficiently take notes during lectures, meetings, and various audio-visual content, which can hinder their focus and understanding. Our aim is to develop a note generation application that automates the note-taking process, allowing users to concentrate on the content while generating comprehensive notes.

Our note-taking application will accept audio recordings or video content as input, and it will generate structured notes as output, suitable for a wide range of applications, including education, business meetings, media tracking, and more. Along with the summary of the contents of the audio recording, CapyNotes will detect the keywords and their definitions and prepare a flashcard set. If possible, the application will create tree schemes such as the main topic and subtopics of the contents. Users can also enhance the generated notes by adding presentation slides or images from whiteboards.

1.2 High Level System Architecture & Components of Proposed Solution

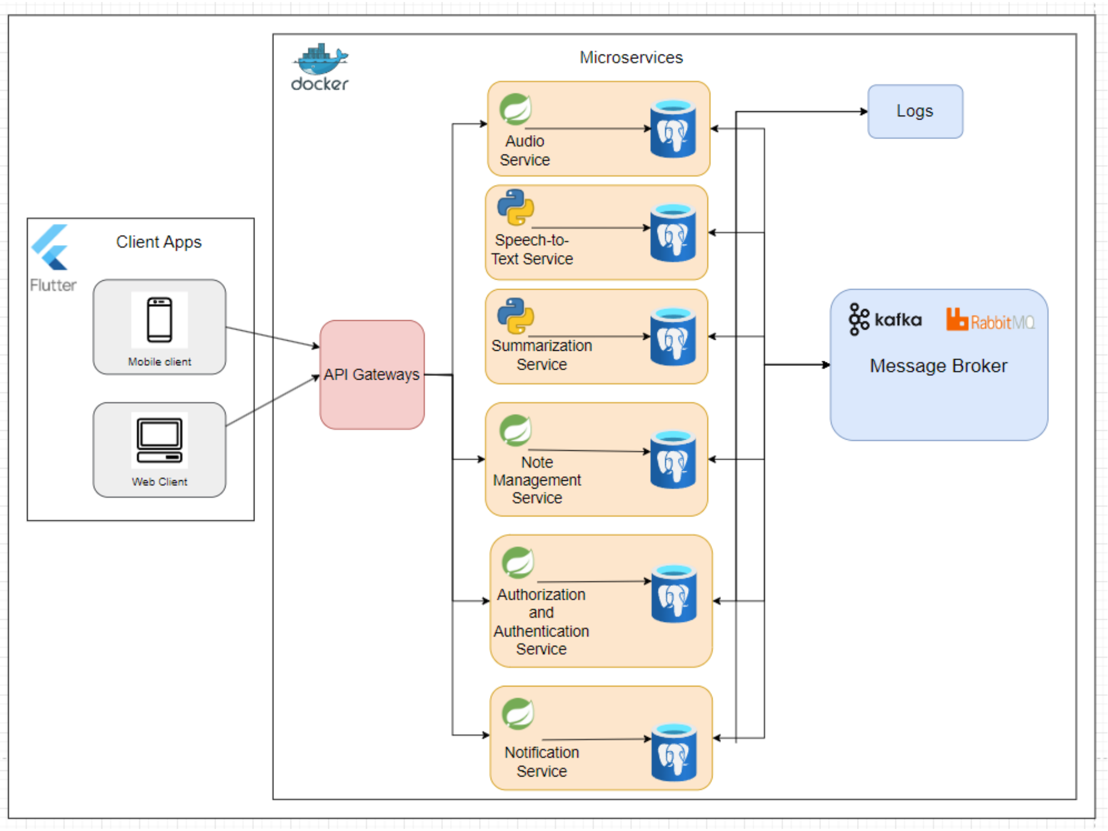


Figure 1: High Level System Architecture of CapyNotes

Client Apps

There will be a mobile application and a web application. These are the frontend applications where users interact with the system. Requests will be sent to the API Gateways.

API Gateways

The API Gateways get requests from the user interface and direct these requests to the correct microservice depending on the request type. The API Gateways will be used for load balancing as well.

Microservices

Each microservice is responsible for a specific functionality. They perform tasks internally and interact with other microservices through Message Brokers like RabbitMQ or Apache Kafka if necessary.

Audio Service

This service is responsible for handling audio recording and uploading from the client applications. It stores the uploaded recordings in its own database. Uploaded audios are sent to Speech-to-Text Service to be transcribed.

Speech-to-Text Service

This service receives audio files from Audio Service and gives transcription of the audio. Then, the transcribed text is sent to the Summarization Service.

Summarization Service

It gets the transcription of the audio recording and summarizes the text into a shorter version. It also identifies important keywords and their definitions found in the text. All this data is sent to the Note Generation Service.

Note Generation Service

This service handles many tasks like drawing schemas based on keywords, converting summarized text into PDF, creating flashcards, making a connection between different notes and timestamping the notes and audio.

Authorization and Authentication Service

This service handles user registration, authentication and authorization. It ensures that users are only able to access their own audios, PDFs and flashcards.

Notification Service

This service is used to send notifications and emails to users when necessary (for example when the note generation is completed).

Message Broker

Either Apache Kafka or RabbitMQ will be used. It will be used for communication between microservices. All the data sending and receiving between the services will be done through the Message Broker.

1.3 Constraints

In this section, various constraints of our project will be discussed.

1.3.1 Implementation Constraints

- 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 CapyNotes 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 Google Cloud Speech To Text API[8] for that purpose.
- PostgreSQL[9] will be used to store and handle data.

1.3.2 Economic Constraints

- All frameworks and libraries that will be used are free.
- Voiser's Transcription API is a paid service. However, we would like to collaborate with them. Google Cloud Speech To Text API is also a paid service.
- Google Play Store and Apple Store require a fee for application publishing.
- Server hosting will require a fee.
- Domain name will require a fee.

1.3.3 Ethical Constraints

- User's private information will not be shared with third party applications.
- Since the user will upload the audio recordings, the application should warn them and get their confirmation regarding the privacy of the audio recordings they upload. They must verify that all uploaded recordings are allowed to be used and to be processed by third parties.

1.3.4 Environmental Constraints

- As the users will be able to provide audio recordings that are recorded by their phone or other unprofessional equipment, we might have to transcribe low quality audio, which might result in poor accuracy.

1.3.5 Language Constraints

- The user interface of our application will support both Turkish and English.
- Currently, we plan to support only English audio recordings. However, we plan to add more language support in the future.

1.4 Professional and Ethical Issues

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 [10] [11].

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.

1.5 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.

2 Design Requirements

2.1 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.

2.2 Non-Functional Requirements

Non-functional requirements are discussed under six categories that are most significant for CapyNotes.

2.2.1 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.

2.2.2 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 (cross-platform SDK) and its extensive set of tools to ensure consistent and efficient development across various targeted platforms.

2.2.3 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.

2.2.4 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.

2.2.5 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.

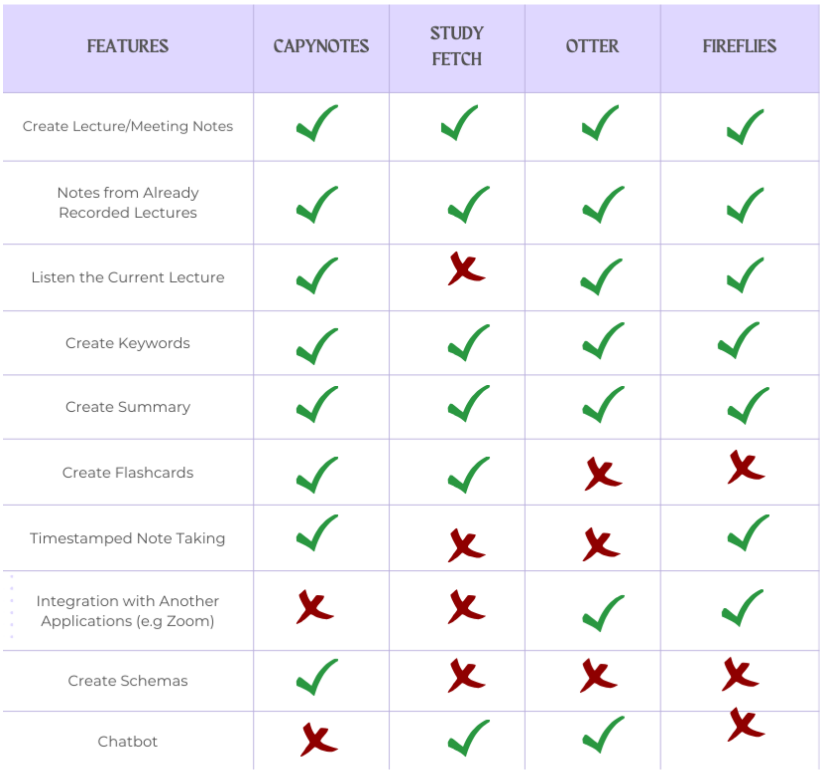
2.2.6 Security

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 Feasibility Discussions

3.1 Market & Competitive Analysis

Before deciding on a definite scope of CapyNotes, many companies, products and projects from the market have been put under the spotlight. In order to conduct a comprehensive competitive analysis, three of them are chosen based on how close their target audience is to CapyNotes' is and which features they provide. A note taking application especially used for lecture and meeting notes -Study Fetch- and two video conference note taking applications are chosen.



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 2: Comparison table of CapyNotes with similar apps

3.2 Academic Analysis

Since the start of the development process of CapyNotes; many academic papers, computer science related forums and various AI technologies have been inspected. The purpose of these researches was to create a substantial academic background for the technologies that can be used in the application and foresee prospective effects of the project on the users and the society.

3.2.1 Technical Discussion

Although it is planned to use available technologies via API's for most of the AI needed processes of the application, in some parts, it is desired to use models for some services such as speech-to-text and summarization. During research, it is determined that some platforms such as HuggingFace might be useful to find high quality models [12]. Even though the models will not be written from scratch, it is planned to fine-tune the pre-trained models, as needed. Furthermore, in our research, significance of usage of hidden markov models in summarization models is noticed [13]. It will be an important criterion while choosing a suitable model for the summarization process.

3.2.2 Effects of the Product

In the absence of an automated note taking application, many students especially in college prefer using tablets and computers to take notes during the lectures. This leads to student - technological device interaction throughout the lecture. A study has shown that active usage of technological devices, especially tablets during the lectures might cause focusing problems for students [14]. Considering the existence of this problem, it is believed that just tapping the “start recording” button, which is one interaction in one lecture, may increase the focus span of the students. On the other hand, a vast majority of studies have demonstrated that writing plays a crucial role in learning [15]. Due to this, active usage of CapyNotes may cause students to abandon their writing habits.

4 Glossary

API: Application Programming Interface

AI: Artificial Intelligence

SDK: Software Development Kit

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

5 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] “Speech To Text” Google Cloud Speech To Text. https://cloud.google.com/speech-to-text?hl=tr

[9]“What is PostgreSQL?”. PostgreSQL. https://www.postgresql.org/about/

[10]“Kişisel Verilerin Korunması Kanunu”. Kişisel Verilerin Korunması Kanunu. https://www.mevzuat.gov.tr/mevzuat?MevzuatNo=6698&MevzuatTur=1&MevzuatTertip=5

[11] “GDPR”. General Data Protection Regulation. https://gdpr-info.eu

[12] “Hugging Face”. Hugging Face. https://huggingface.co/

[13] J. M. Conroy and D. P. O’leary, “Text summarization via Hidden Markov models,” Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval, 2001. doi:10.1145/383952.384042

[14] E. Soykan, “Views of Students’, teachers’ and parents’ on the tablet computer usage in education,” Cypriot Journal of Educational Sciences, vol. 10, no. 3, p. 228, 2015. doi:10.18844/cjes.v1i1.68

[15] F. Alharbi, “Writing for learning to improve students’ comprehension at the college level,” English Language Teaching, vol. 8, no. 5, 2015. doi:10.5539/elt.v8n5p222