Meeting Minutes Summarizer FYP

Final Report

https://github.com/wasifsamad/MeetingMinutesSummarizer

BY

Member(s):

Wasif Samad 14207

Surendar Kumar 14878

PROJECTS COMMITTEE (PC)

SUPERVISOR: ***Tahir Syed***

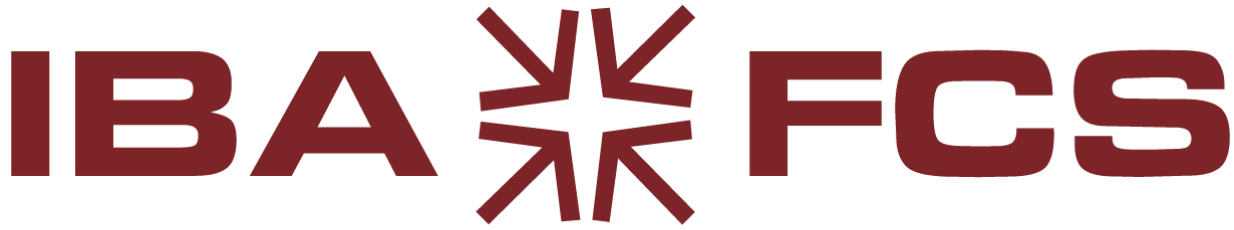
Project Committee Member: ***Sumaira Syed***

SUBMITTED TO

PROJECTS MANAGER – FYP

ON

DATE (5/6/2021)



Faculty of Computer Science

(TO BE LEFT BLANK)

CONTRIBUTIONS

The idea and project tasks were distributed among both group members.

Surendar

The first draft of the PDR was made by Surendar. In literature review, major contribution related to summarization was done. All diagram related parts in SRS were completed. Added changes recommended by committee. For the initial android project, made some screens design. Made front end HTML5 and CSS. Made 40% of summarizer.

Wasif

Provided information related to market review. Second draft and onwards were prepared by Wasif. In literature review, speech to text API and research related to diarylation was mostly done by Wasif. The SRS document first draft was prepared by Wasif. This includes everything except diagrams. For android project, made recording screen and worked to implemented speech to text. Made speech to text and 60% of summarizer.

Contents

Project Defense Report (PDR)

Meeting Minutes Summarizer

Project Defense Report

(7th semester)

BY

Surendar Kumar 14878

Wasif Samad Shaikh 14207

SUPERVISOR: Dr. Tahir Syed

SUBMITTED TO: ***Dr. Tahir Syed***

PROJECTS MANAGER

Abstract

MMS aims to increase the efficiency of users by helping them to fully focus on conversation without worrying about taking notes. Users do not lose track of conversations and participate actively in discussions. They will open the webapp on their laptop browser. When the recording stops, the audio is converted into text. MMS not just gives the text format of meeting but also presents the concise summary. It saves the summary of your meeting which you may recall later. For example, if you are sitting in a meeting and want to focus on discussions rather making notes simultaneously, just open the MMS application and let it create notes on your behalf.

PROJECT OVERVIEW

A PROJECT TITLE: Meeting Minutes Summarizer

B ACRONYM: MMS

C TAGLINE: Relax! We will take your notes.

D START DATE: | 0 | 2 | 1 | 1 | 2 | 0 | 2 | 0 |

E EXPECTED COMPLETION: | 0 | 4 | 0 | 6 | 2 | 0 | 2 | 1 |

F PROJECT BRIEF

MMS (Meeting Minutes Summarizer) will be a web application which will take the audio as an input during meeting, conversation, lecture, or any other related environment and create a concise summary. In the first part we will record the voice and convert that into the text using speech to text API. In the second part, which is the major part, we will use Natural Language Processing techniques and tools to recognize the words and associate meaning with them, we call them tokens. Those tokens then will be used to create a context. What makes our app different is that our web app will provide a summary using machine learning techniques. Finally, for developing the web app we will use MERN framework.

PROBLEM STATEMENT

Most participants during a meeting do not write a memo. It would be better to have an app that could automatically write the meeting summary.

MARKET/LITERATURE REVIEW

During the market research we found the following applications which try to solve the problem discussed above:

1) Otter Voice Meeting Notes for English***:*** Otter is a voice to text conversion app. It takes audio as an input and converts it in the text form. It is a commercial product and gives only 600 free minutes for a month. It saves the notes and has a share option, using that option one can share notes directly to his teammates. It is not an open-source application and charges the users.

2) Braina***: It offers various services. Making notes from voice recording is one of their popular service. It is commercial product and offers very few and limited services in unpaid version. It has specific requirements like it works on windows 10, 8.1, 8 and 7 only. Briana does not make summary of the meeting.***

3) Amazon Transcribe***: This is also a commercial service on cloud. It comes with the AWS cloud services. It takes a voice notes and produces the transcribes of voice notes. In order to use this service one will need to buy the cloud service. Good thing about this service is that it provides many categories in voice to text conversion like a customized service for medical audio etc. It misses the summary making feature.***

4) ListNote Pro NotePad***: It is also a paid service. It is a little different from the former services because it has an option to record the audio and convert it to text on the fly. Its interface is enriched with many editing and formatting options. It is a useful app if you want to make notes of your lecture, but it also misses the summary making feature.***

5) Deep Talk***:*** It transcribes your conversations on the fly. deepTalk will generate a transcript enriched with speaker identification and voice separation, allowing you to understand exactly who said what. ***Like all the above-mentioned services, it is also paid.*** They have built their own algorithm, SumMeet, which extracts and presents the necessary actions, decisions, and insights and creates a summary. This is all good except that it is a commercial app and charges for most of its services.

Apart from the above services there are many other services available too. 6) Reason8, 7) Penman, 8) Dragon, 9) SpeechNotes, 10) SpeechTexter and 11) Azure cloud services and 12) IBM cloud services are among them. All the above services share a common problem that they are all commercial. Some of them offer free versions but with very limited services and most of them does not have a feature for making summary.

PROJECT DETAILS

A PROPOSED SOLUTION:
In the market research, we found that there are many products available for transcribing the voice notes into the text, except DeepTalk, no service or product provided the summary of the meeting, and those two also charge heavily for providing the service of summary making. Our solution would free of cost, open-source and will include a feature for making concise summaries. We will use NLP, AI and Machine Learning techniques and tools to create this extra feature.

B PROJECT OBJECTIVES:
To create a service for professionals and students to save their efforts, and time. Increase their

productivity and efficiency. We achieve mentioned goals by creating an MMS app that summarizes their conversations and saves them, so that people can recall their meetings later.

C METHODOLOGY:

Below is the stepwise breakdown of MMS app:

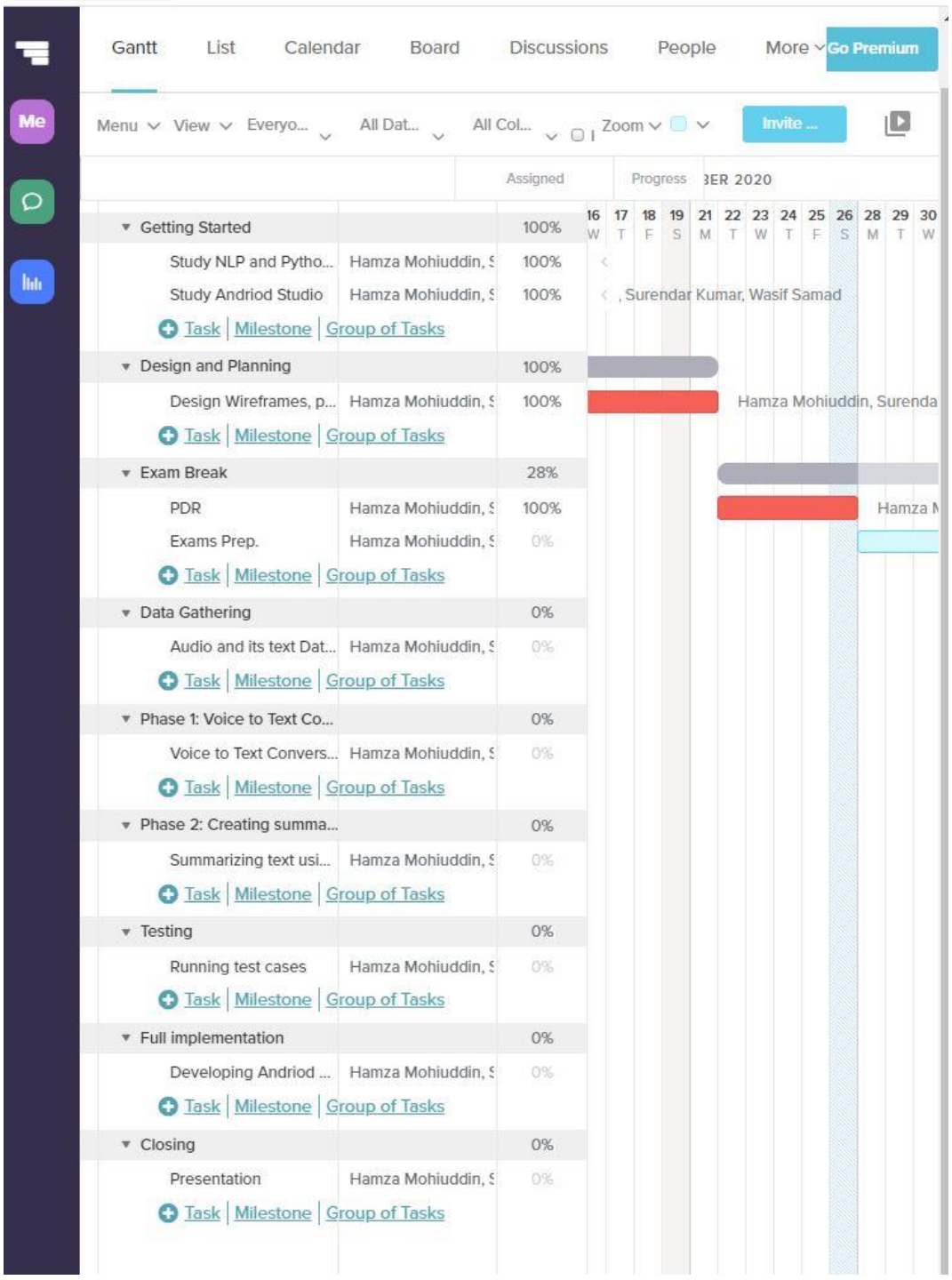
1. Literature review
2. Problem statement
3. Design
4. POC implementation
5. Phase 1: Voice to Text Conversion
6. Phase 2: Creating summaries of texts
7. Testing
8. Full implementation

D THE PRODUCT:

1. Web application
2. Software Documentation
3. Reports
4. User Manual

PROJECT MILESTONES AND DELIVERABLES

Gantt Chart Link:



WORK DIVISION

Every group member will be assigned equal workload and responsibilities. We will take the project step-wise and both of us will keep doing work together.

COSTING

We may need to purchase cloud services for understanding voice to text API's. Total cost will be around 10 to 15k, which includes app development process too. We will manage the cost ourselves.

REFERENCES

[1] Santosh K.Gaikwad, 'A review on speech recognition techniques', International Journal of Computer Applications, Volume 10– No.3, November 2010.

[2] Su Myat Mon, Hla Myo Tun, "Speech-To-Text Conversion (STT) System Using Hidden Markov Model (HMM)," IJSTR, vol. 4, June 6, 2015.

[3] Prachi Khilari1 , Prof. Bhope V. P.2, "Implementation of Speech to Text Conversion," IJIRSET, Vol. 4, Issue 7, July 2015.

[4] Deepa V.Jose, Alfateh Mustafa, Sharan R , "A Novel Model for Speech to Text Conversion," IRJES, Vol. 3, Issue 1 (January 2014), PP. 39-41.

[5] "Learn how to Build your own Speech-to-Text Model (using Python)," Analytics Vidhya. https://www.analyticsvidhya.com (accessed July 15, 2019).

[6] "Automated meeting notes for in-person and remote conversations," DeepTalk. https://deeptalk-ai.com/ (accessed Aug 20, 2020).

Software Requirements Specification

Introduction

The introduction of the Software Requirements Specification (SRS) provides an overview of the entire SRS with purpose, scope, definitions, acronyms, abbreviations, references, and overview of the SRS. The aim of this document is to gather and analyze and give an in-depth insight of the **Meeting Minutes Summarizer Application** by defining the problem statement in detail.

Purpose

The purpose of the document is to collect and analyze all assorted ideas that have come up to define the system, its requirements with respect to consumers. Also, we shall predict and sort out how we hope this product will be used to gain a better understanding of the project, outline concepts that may be developed later, and document ideas that are being considered, but may be discarded as the product develops.

In short, the purpose of this SRS document is to provide a detailed overview of our software product and goals. This document describes the project's functionality and its user interface, hardware, and software requirements. It defines how our client, team and audience see the product and its functionality. Nonetheless, it helps any designer and developer to assist in software delivery lifecycle (SDLC) processes.

Scope

Primarily, the scope pertains to the features of Meeting Minutes Summarizer. Meeting Minutes Summarizer will be a web application. It will be used in a meeting setting where there are 2-3 speakers and not all speakers will speak at the same time. Beta version will be ready by 10th May. Modules of the app will include a voice to text transcriber and a summarizer. Further functionality and details are mentioned in section 2.

Definitions, Acronyms, and Abbreviations

FAQ	Frequently Asked Questions
User	Someone who interacts with the application
MMS	Meeting Minutes Summarizer

1. References

The references are:

[1] IEEE Software Engineering Standards Committee, “IEEE Std 830-1998, IEEE Recommended Practice for Software Requirements Specifications”, October 20, 1998.
[2] Wiegers, Karl, Software Requirements (3rd Edition), Microsoft Press 2013
[3] Standardized Statement Templates – beginning to write SRS
Link: https://www.coursera.org/learn/srs-documents-requirements/lecture/1o2v0/standardized-statement-templates

Overview

The remaining sections of this document provide a general description, including characteristics of the users of this project, the product's hardware, and the functional and data requirements of the product.

Overall Description

2.1 Product Overview

MMS aims to increase the efficiency of users by helping them to fully focus on conversation without worrying about taking notes. Users do not lose track of conversations and participate actively in discussions. They will open the web app in their browser and start the recording. When the recording stops, the audio is converted into text. The text transcription will be converted to a summary. It saves the summary of your meeting which you may recall later. For example, if you are sitting in a meeting and want to focus on discussions rather than making notes simultaneously, just open

the MMS application and let it create notes on your behalf.

2.2 Similar System Information

1) Otter Voice Meeting Notes for English: Otter is a voice to text converter app. It takes audio as an input and converts it in the text form. It is a commercial product and gives only 600 free minutes for a month. It saves the notes and has a share option, using that option one can share notes directly to his teammates. It is not an open-source application.

2) Braina: It offers various services. Making notes from voice recording is one of their popular service. It is commercial product and offers very few and limited services in unpaid version. It has specific requirements like it works on windows 10, 8.1, 8 and 7 only. Briana does not make summary of the meeting.

3) Amazon Transcribe: This is also a commercial service on cloud. It comes with the AWS cloud services. It takes a voice notes and produces the transcribes of voice notes. To use this service one will need to buy the cloud service. Good thing about this service is that it provides many categories in voice to text conversion like a customized service for medical audio etc. It misses the summary making feature.

4) ListNote Pro NotePad: It is also a paid service. It is a little different from the former services because it has an option to record the audio and convert it to text on the fly. Its interface is enriched with many editing and formatting options. It is a useful app if you want to make notes of your lecture, but it also misses the summary making feature.

5) Deep Talk: It transcribes your conversations on the fly. deepTalk will generate a transcript enriched with speaker identification and voice separation, allowing you to understand exactly who said what. Like all the above-mentioned services, it is also paid. They have built their own algorithm, SumMeet, which extracts and presents the necessary actions, decisions, and insights and creates a summary. Deep talk predicts the intensity of the conversation as low, medium, or high. Deep talk automatically adds topics after the conversation has ended.

Apart from the above services there are many other services available too. 6) Reason8, 7) Penman, 8) Dragon, 9) SpeechNotes, 10) SpeechTexter and

11) Azure cloud services and 12) IBM cloud services are among them.

All the above services share a common feature that they are all commercial. Some of them offer free versions but with very limited services and limited conversation time. Furthermore, most of them do not have a feature for making summary.

Feature/Name	Android App	Creates Transcription	Speaker Separation	Creates Summary	Free
Meeting Minutes Summarizer	No	Yes	Yes	Yes	Yes
Otter	Yes	Yes	No	No	Limited options
Braina	Yes	Yes	No	No	Limited options
Amazon Transcribe	No	Yes	No	No	No
ListNote	Yes	Yes	Yes	No	No
Deeptalk	No	Yes	Yes	Yes	Limited options

2.3 User Characteristics

MMS is intended to be used by people participating in a meeting. A typical user should have some experience using a browser on a laptop. The number of participants supported are 2 or 3, and not all speakers should speak at the same time.

2.4 User Problem Statement

It is difficult to participate in a meeting and write meeting minutes at the same time.

Specific Requirements

The specific requirements are –

Functionality

Introduction –

This subsection contains the requirements for MMS. Features then refined into use case diagrams and to sequence diagram to best capture the functional requirements of the system.

The web app shall record live audio conversation.

After tapping record icon, the user will start the recording.

The recording can only be started by clicking the record icon on home page.

The app shall provide transcriptions of the audio conversation recorded.

As the recording progress transcriptions on the screen.

The app shall provide a summary of the meeting audio conversation.

After user clicks stop icon summary of the meeting will be shown.

The app shall allow the user to edit voice transcription.

User can edit voice transcription by editing in the transcription box.

The app shall allow the user to recreate summary after editing transcriptions.

After editing transcriptions, the user can click summarize and a new summary will be created.

The app shall allow the user to delete a meeting.

Meeting details, summary, as well as transcriptions for that meeting will be deleted.

Usability

Graphical User Interface

The system shall provide a uniform look and feel for the web app.

The interface shall consist of a bar to create a new meeting, and icons to pause or stop one if the user wants to end meeting.

Reliability & Availability

Back-end Internal Computers

The app shall use JavaScript for backend development. Other than that, DeepAffects speech to text and summarization libraries are made use of.

Local Storage

Meeting summary and transcriptions should be saved locally.

Performance

The product shall be based on web and will be run from a web server.

The product shall start up in less than 10 seconds (using Chrome Dev tools).

Supportability

The app shall be built using JavaScript version ECMA 2020, and HTML5. All latest browsers will support this.

Design Constraints

Standard Development Tools

The app shall be made using HTML5, CSS and java script as the programming language. DeepAffects API are used for speech to text and summarization.

Interfaces

Hardware Interfaces

All we browsers with JavaScript ECMA 2020 supported. Microphone access is required.

Communications Interfaces

The application shall utilize the networking hardware of the user's device through network communications services provided by the operating system.

Legal, Copyright, and Other Notices

IBA GNU

Response Time

The application should show summary in less than 3 seconds of when summarize is clicked.

Workload

The application must support at least 2 different speakers at the time of launch.

Reliability

Application should provide feedback for each action. The page should change after an action. Or the icon pressed should change.

Use Cases

Use Case Name:	Create new meeting
Actors:	End User
Description:	The user will be able to start a new meeting recording
Trigger:	Tap the voice record icon
Preconditions:	1. A web browser with JavaScript supported should be available. 2. End user is at the homepage with no meeting in progress
Postconditions:	A summary as well as transcriptions are created and saved.
Normal Flow:	1.0 User taps record button and speaks. 2.0 User clicks stop the meeting stops and it ends. 3.0 Summary is shown.

Alternative Flows:
[Alternative Flow 1 – Not in Network]

Exceptions:	2E. Microphone access not given. -2E1. Displays an error message.

Use Case Name:	View all meetings
Actors:	End User
Description:	The user will be able to browse through all the previous meetings
Trigger:	Tap view meetings
Preconditions:	The user is not currently recording a meeting.
Postconditions:	A webpage where all previous meetings are shown
Normal Flow:	1.0 User scrolls to where the meetings sections
Alternative Flows:	2.1. User has no meetings. 2.2 User is shown the message you have no meetings would you like to record one.

Use Case Name:	Edit transcriptions
Actors:	End User
Description:	The user will be able to edit transcriptions
Trigger:	User clicks in text box
Preconditions:	1.The user is at the page with transcriptions showing 2. End user no meeting in progress
Postconditions:	Transcriptions are edited
Normal Flow:	1.0 Edit the transcriptions. 2.0 Click save.

Alternative Flows:
[Alternative Flow 1 – Not in Network]

Exceptions:	2E. User deletes all transcription . -2E1. Displays an error message .

Use Case Name:	Recreate summary
Actors:	End User
Description:	The user will be able to recreate the summary
Trigger:	Tap the create summary option
Preconditions:	1.User has edited the transcriptions 2. End user has no meeting in progress
Postconditions:	A summary as well as transcriptions are created.
Normal Flow:	1.0 User taps create summary. 2.0 A new summary is created.

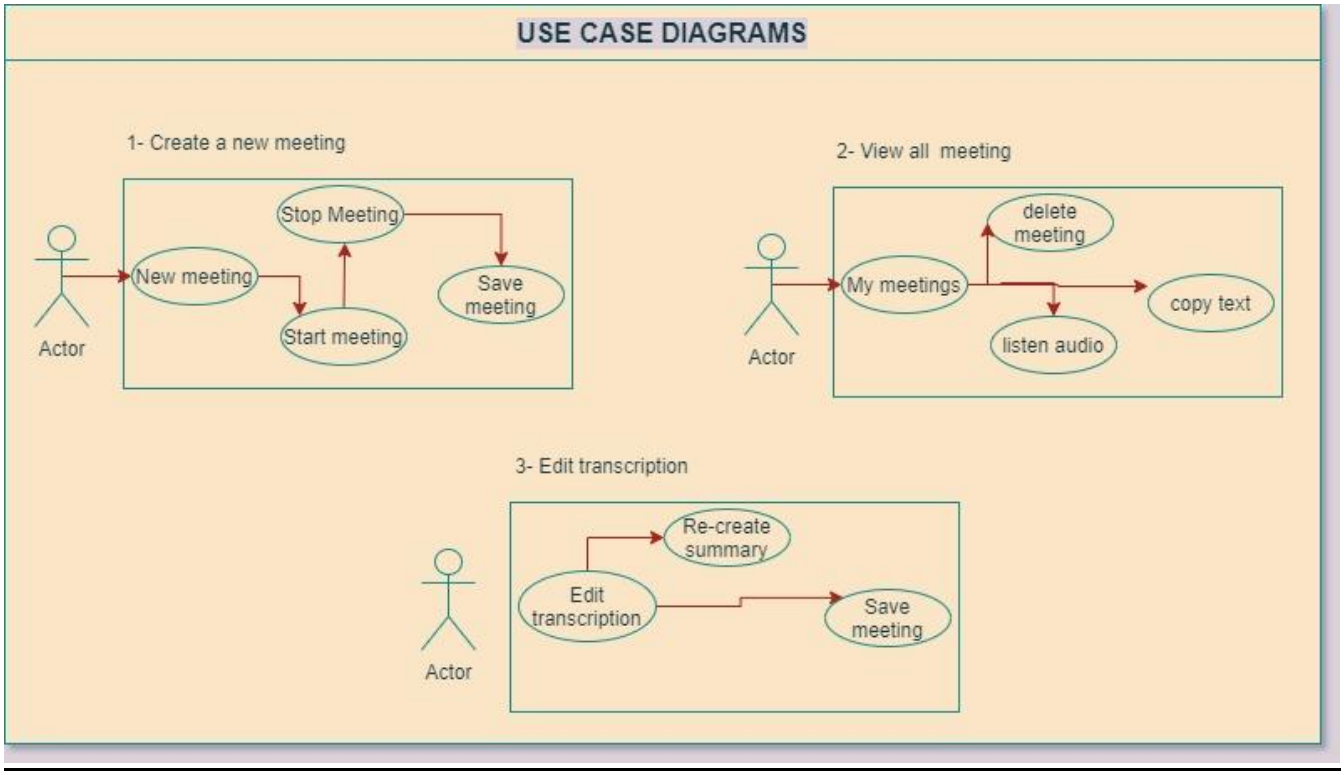
Alternative Flows:
[Alternative Flow 1 – Not in Network]

Exceptions:

Use Case Name:	Delete a meeting
Actors:	End User
Description:	The user will be able to delete a meeting
Trigger:	Tap view meetings
Preconditions:	The user is not currently recording a meeting. The user has at least one meeting
Postconditions:	Meeting will be deleted
Normal Flow:	1.0 User taps delete button on meeting page. 2.0 User is redirected to home page.

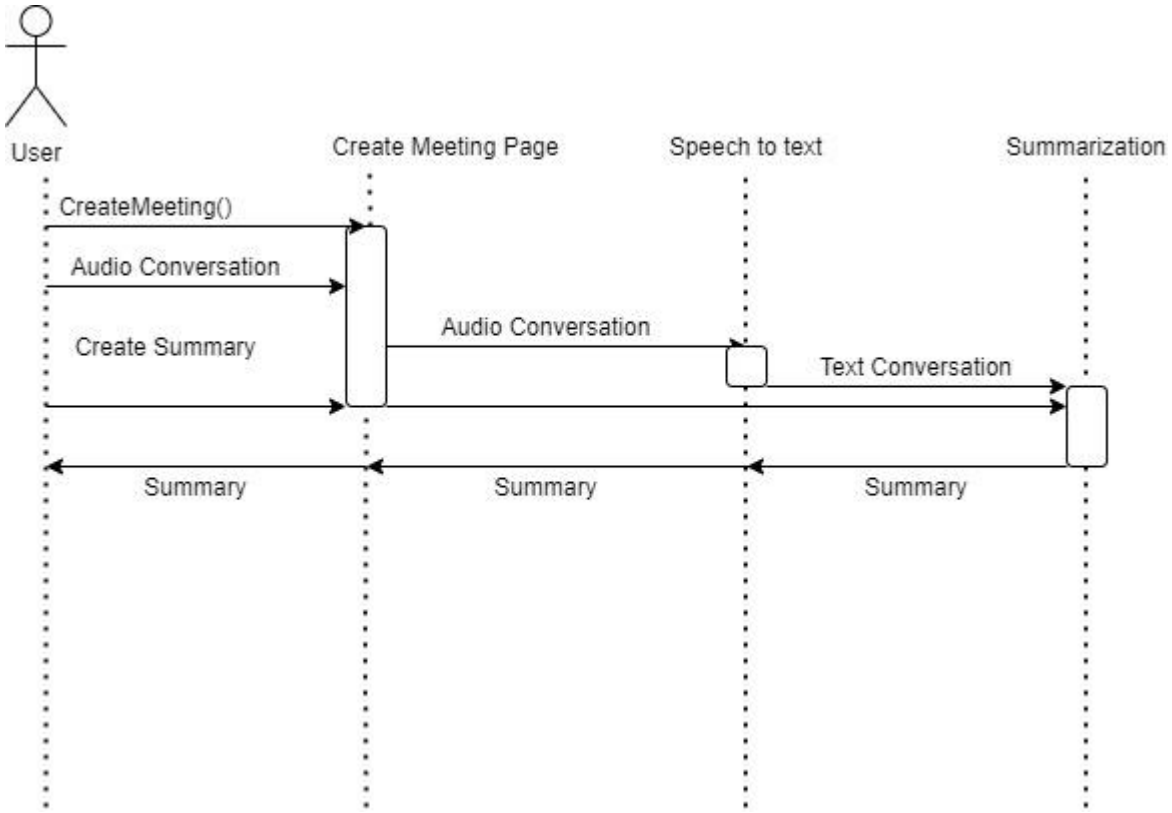
Alternative Flows:

Use Case Diagrams

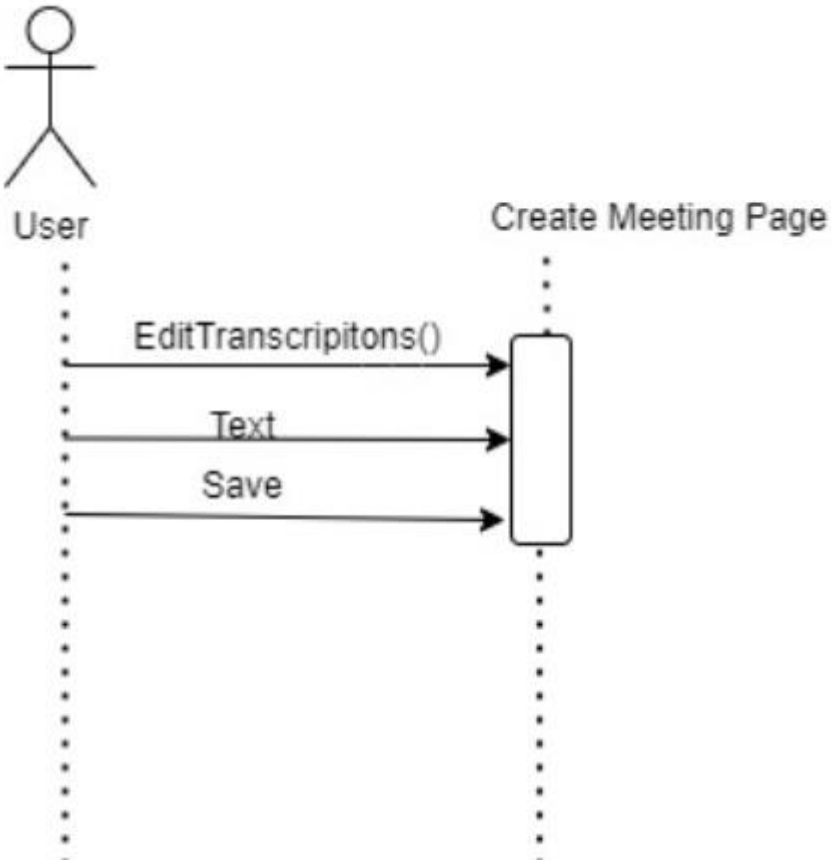


Sequence Diagram

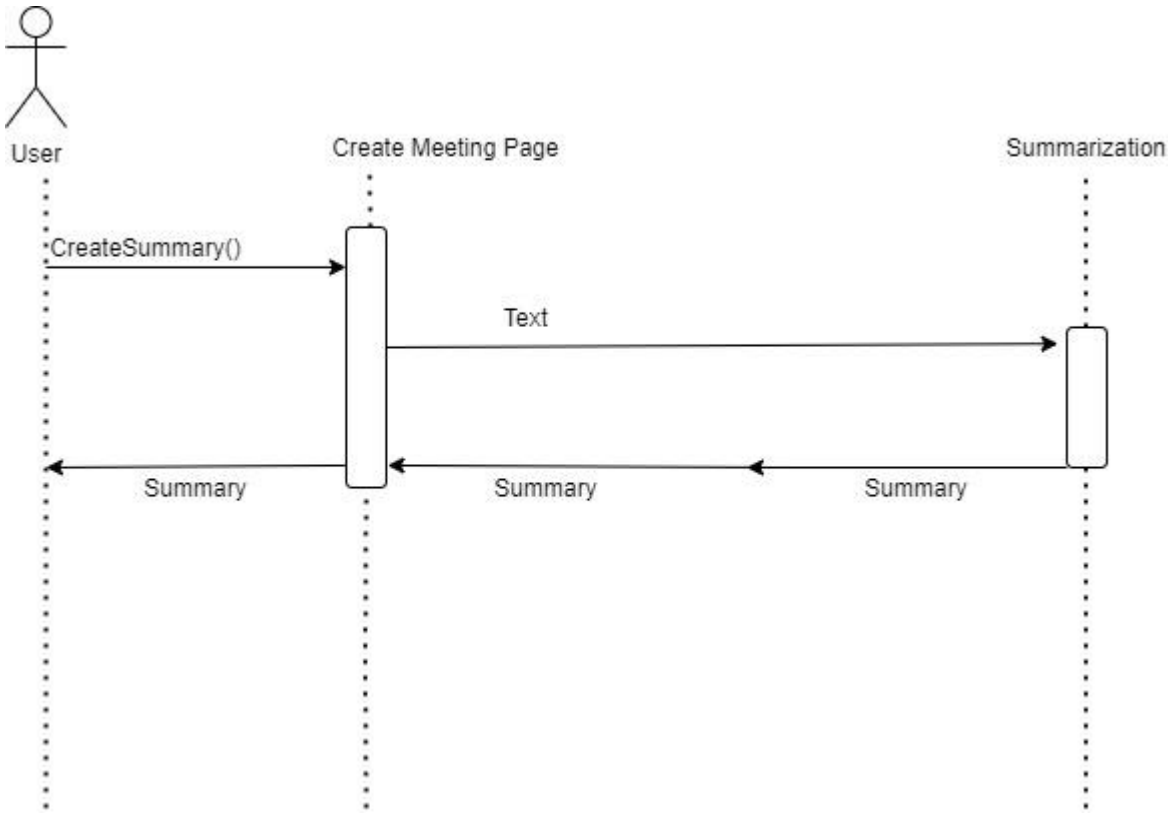
Create Meeting



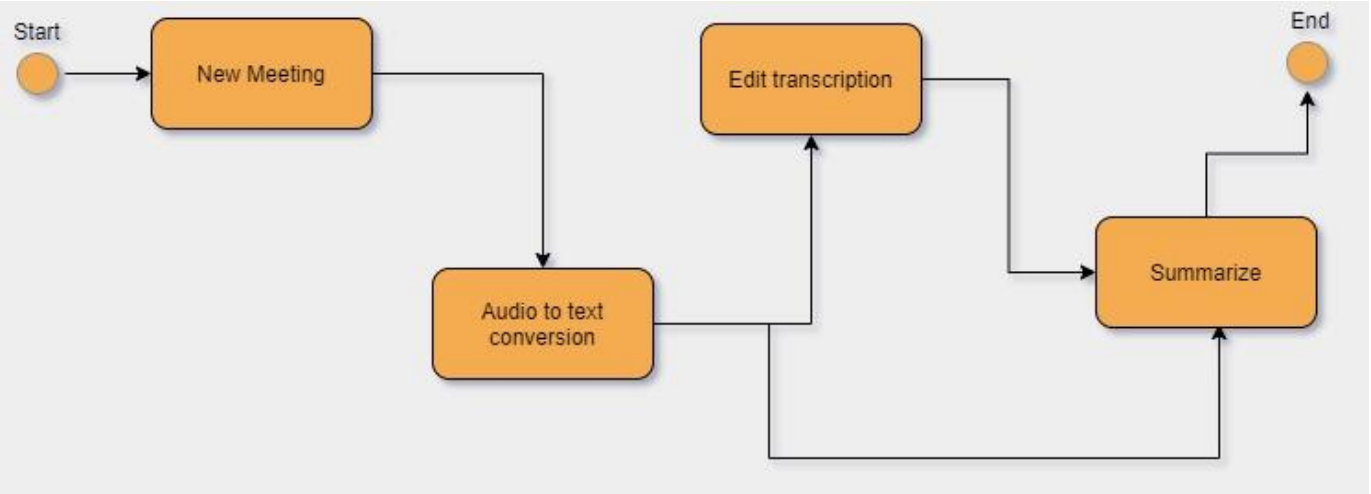
Edit Transcriptions



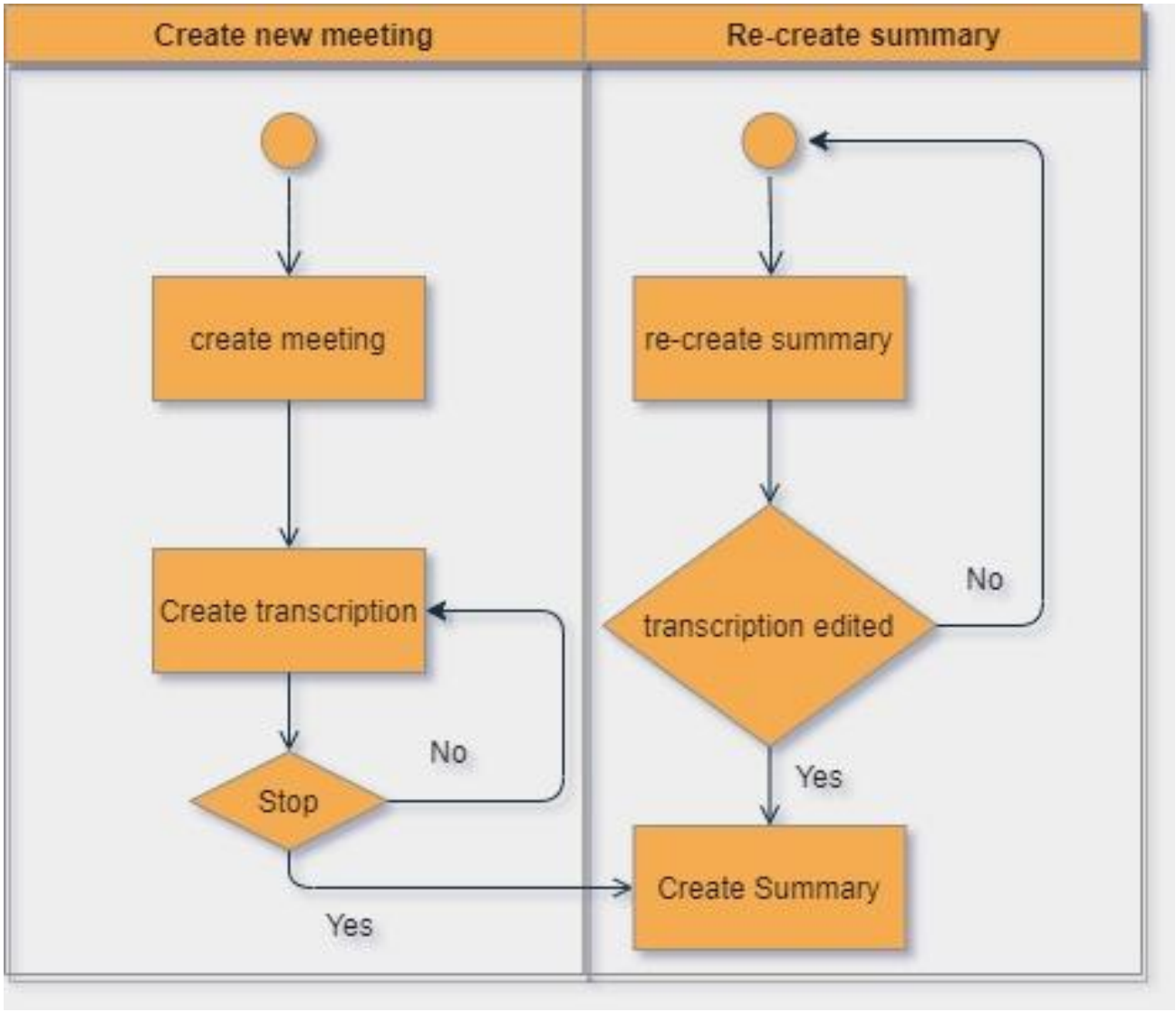
Recreate Meeting



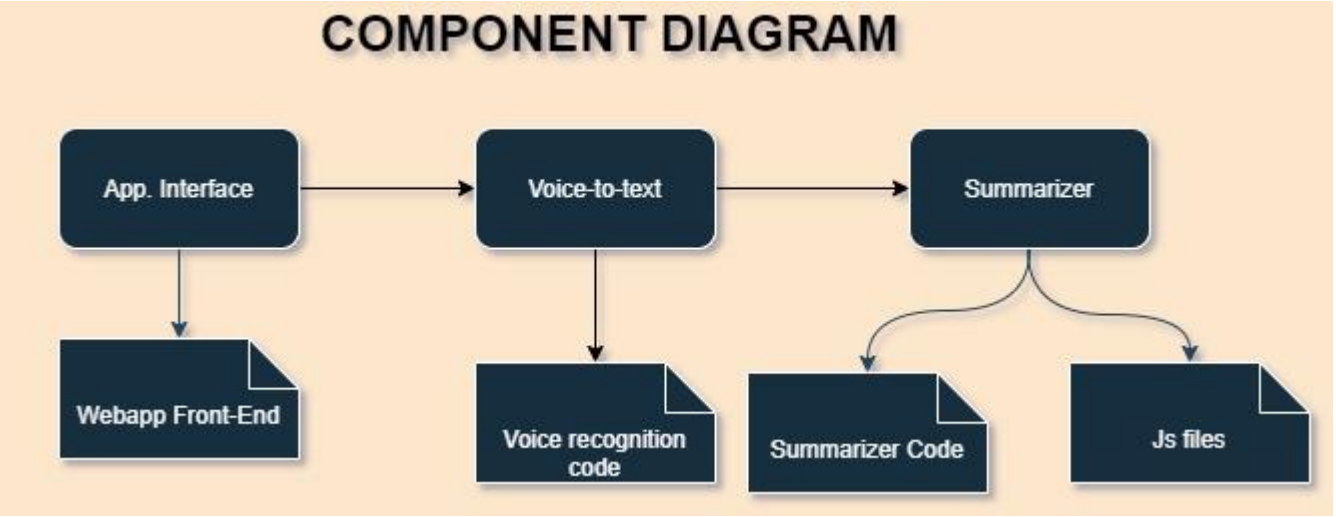
State Chart Diagram



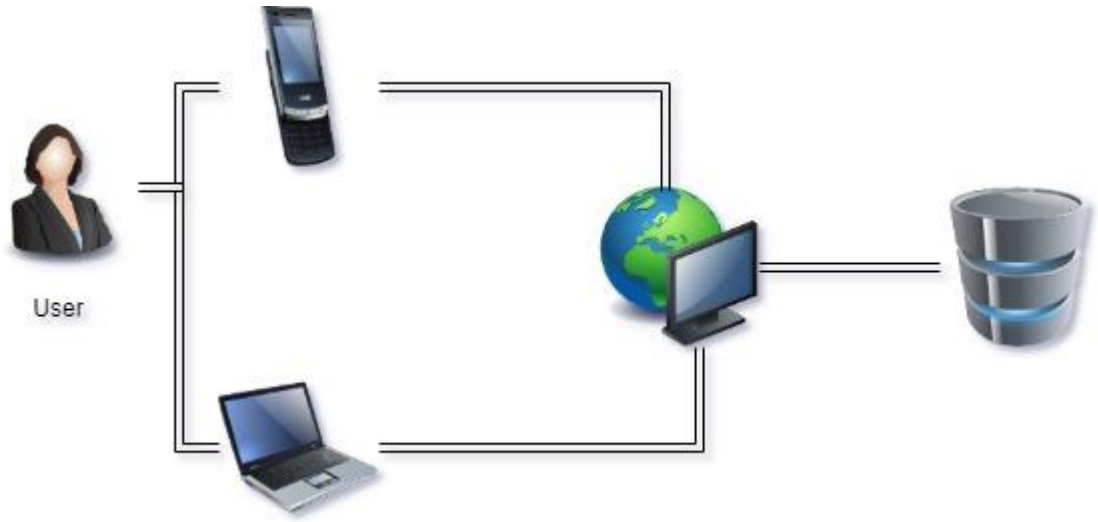
Activity Diagrams



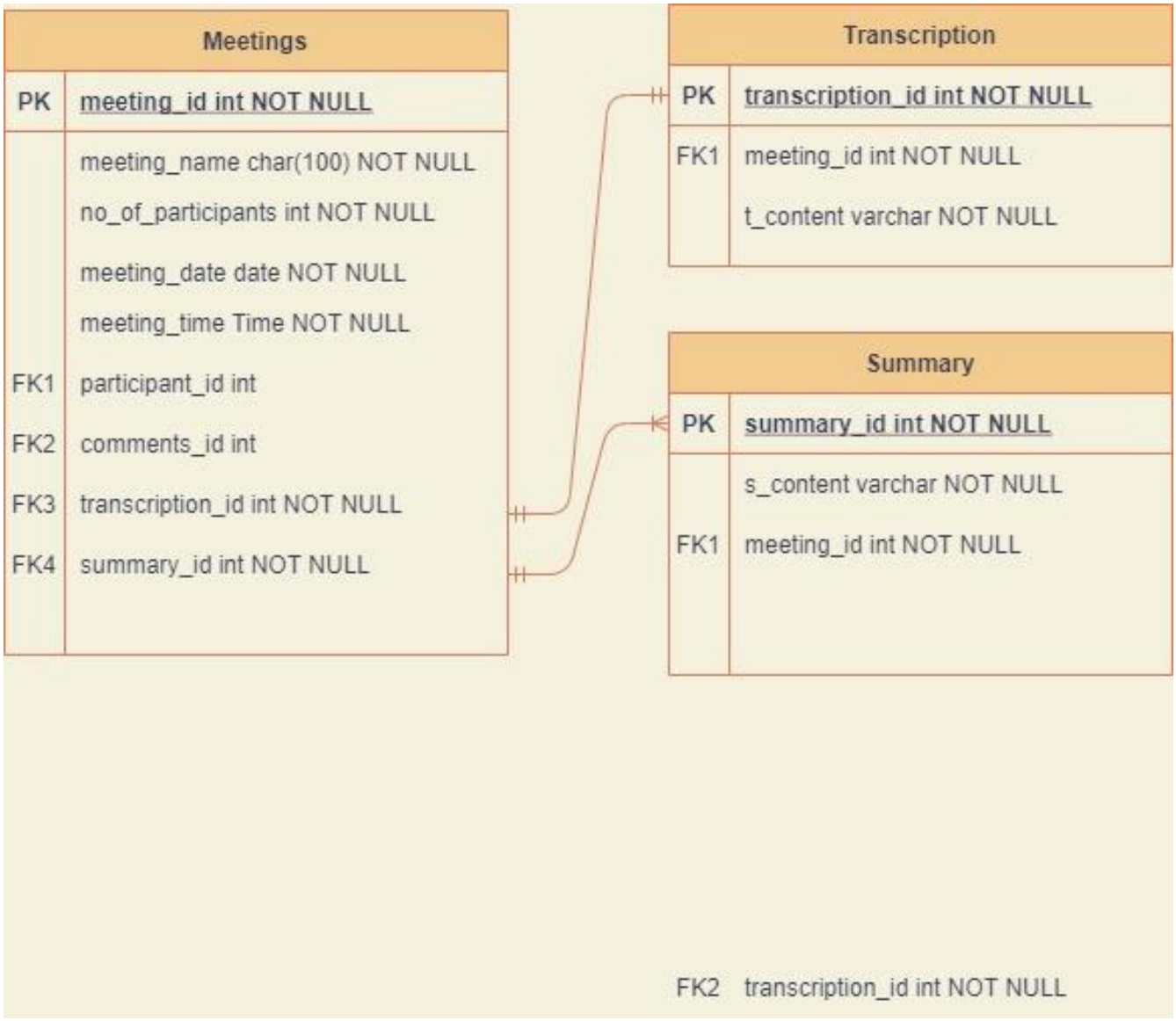
Component Diagram



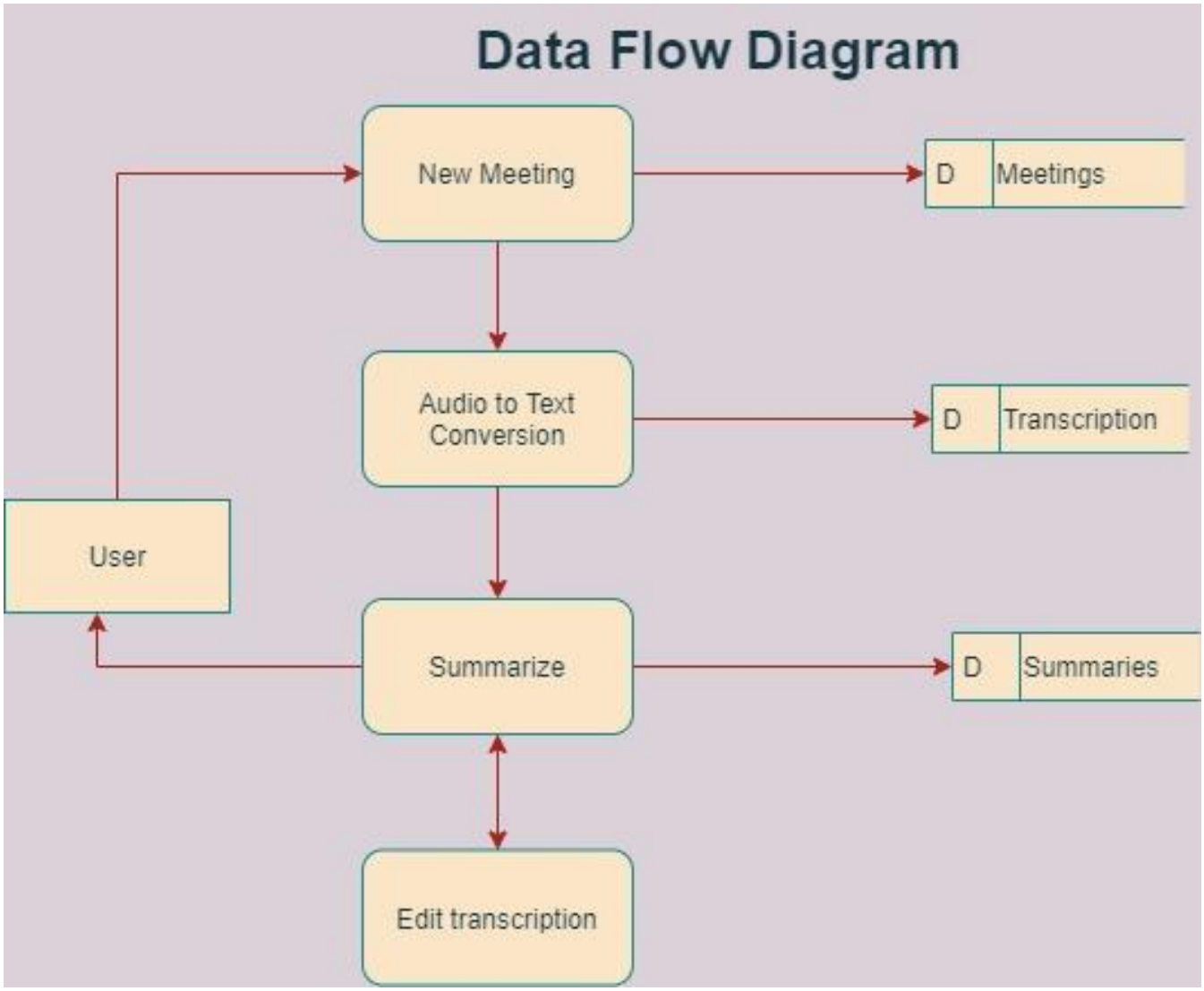
Deployment Diagram



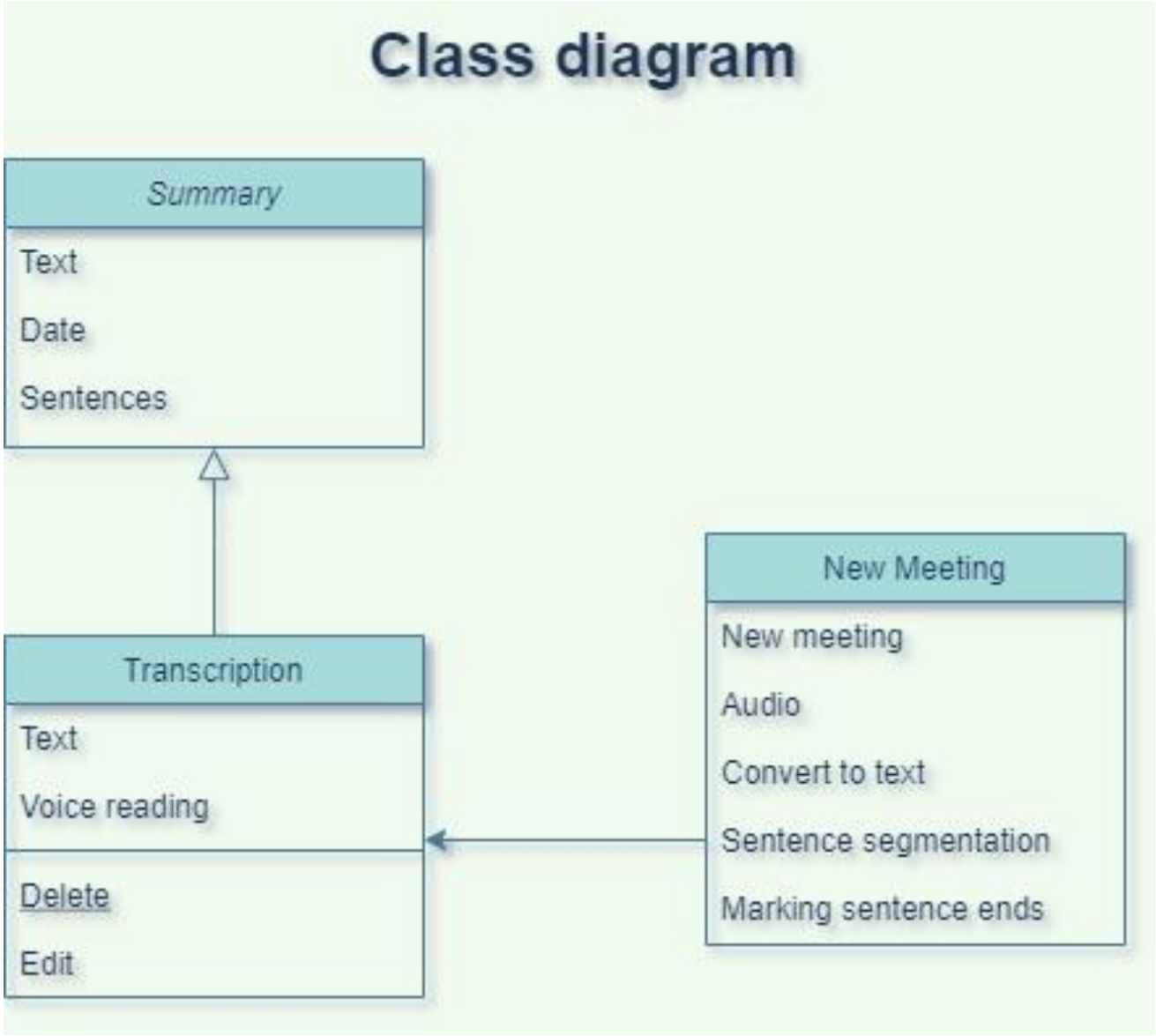
Database Model - (ERD Diagram)



Data Flow Model



Class Model


Class diagram
Summary
Text
Date
Sentences
Transcription
Text
Voice reading
Delete
Edit
New Meeting
New meeting
Audio
Convert to text
Sentence segmentation
Marking sentence ends

System Design

1. System Architecture and System Design

1.1. Architectural Style

Call and return architecture.

Each click on the button creates a call to a method in the backend. The backend returns the output and prints it on the screen.

1.2. Network Protocol

Using Live Server to run the webapp on localhost.

1.3. Hardware/Software Requirements

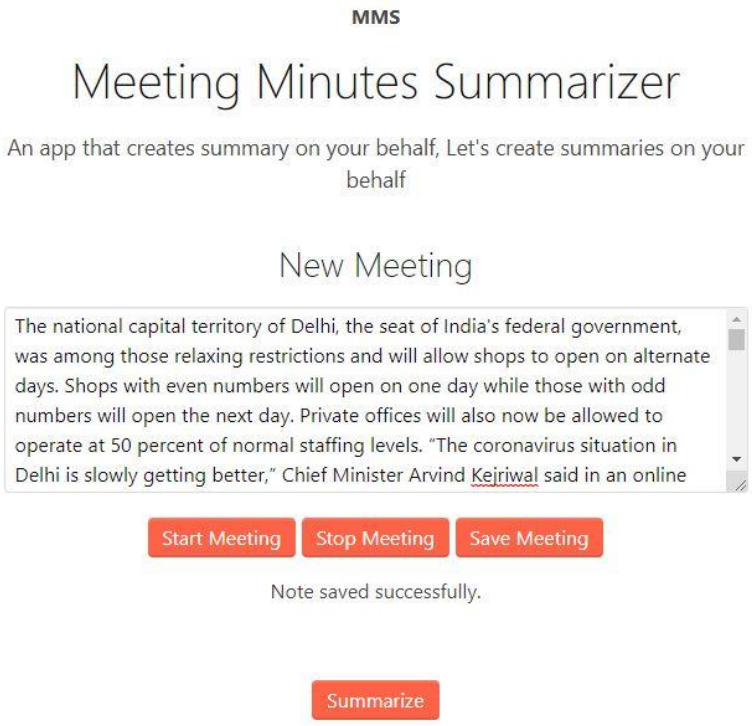
Any system with a browser supporting JavaScript, HTML5, and CSS. This includes all systems created after 2015, and all popular browsers.

2. Algorithms and Data Structures

Basic OOP concepts used including functions. Array is used to store the text segments, that is the transcriptions.

3. User Interface Design and Implementation

The user interface is a simple single page interface. The page is created on HTML5 and CSS. A snapshot is shown as below.



4. Design of Tests

Test case 1

Test case description: Testing loading time

Expected Output: Loading time should be less than 3 seconds.

Actual Output: Of 20 loading all were measured by human and loaded in less than 3 seconds.

Test case 2

Test case description: Creating transcription.

Expected Output: A transcription on the screen.

Actual Output: A transcription on the screen.

Test Case 3

Test case description: Creating a summary of the transcription.

Precondition: There should be at least 5 sentences in the transcriptions.

Expected Output: a summary of the top 5 sentences is created.

Actual Output: a summary of the top 5 sentences is created.

Test case 4:

Test case description: Creating a meeting of 2 people.

Expected Output: Transcription and summary of their conversation.

Actual Output: Transcriptions and summary of the conversation are presented on screen.

Test case 5:

Test case description: saving multiple meetings.

Expected Output: Meetings are saved with the newest meeting summary showing first.

Actual Output: The summaries of all meetings are shown on My meeting section with the latest showing first.

User Manual

Open the 2 .js files and html and css files in one project. Make a separate directory for them. Open them in Visual Studio or local browser, if using Visual Studio you should Install Live server and open the html file with that.

Project Management and Plan

Around Jan to first week of February we did market and literature review. After that we created SRS document and a prototype for app by March. Then we started making the app. In end of April and May our other courses load increased so we had to stop. For various reasons including poor android development skills and not being able to implement summarizer for android forced us to redo our project as a web app. Till mid may to June 5 we worked on the web app. And lastly also created the final report.

Current Status and Future Work

Currently our web app is focused to run on local host. It creates transcriptions, summary, and saves it. It uses extractive summarizer.

Future people can use a cloud server to make it accessible from the web. Other than that people can research in abstractive summarization to produce an abstractive summary. Also, people can gather data related to a specific type of meeting and make the app more accurate in those cases.

This web app was made by our thought-out requirements. A better app can be made by using various clients' requirements.

People could build an app that loads our webpage, making the summary available on app. Other than that, users can create an entirely new app with the same idea but more pages and interaction.

Lastly, people can work in doing summarization using audio which will output audio conversation.

References

[1] IEEE Software Engineering Standards Committee, "IEEE Std 830-1998, IEEE Recommended Practice for Software Requirements Specifications", October 20, 1998.
[2] DeepAffects API Documentation, "https://docs.deepaffects.com /docs/speech- to-text-api.html", 2020.
[3] Wiegers, Karl, Software Requirements (3rd Edition), Microsoft Press 2013
[4] Standardized Statement Templates – beginning to write SRS

Link: https://www.coursera.org/learn/srs-documents-requirements/lecture/1o2v0/standardized-statement-templates

The End