NATIONAL UNIVERSITY OF COMPUTER AND EMERGING SCIENCES, KARACHI

SOFTWARE DESIGN SPECIFICATION

Automation of Minutes of Meeting

Author:
Mehdi Raza K16-3904,
Hussain Yousuf K16-3805,
Shehryar Naeem K16-3950

Supervisor:
Dr. Muhammad Rafi

22nd December, 2019

Revision History

Version	Name of Person	Date	Description of Changes
1.0	Mehdi	30 Oct 2019	Created the Document
1.0	Hussain	2 Nov 2019	Added Non Functional Requirements
1.0	Mehdi	4 Nov 2019	Added Use Case Diagrams
1.0	Shehryar	4 Nov 2019	Added Introduction
1.0	Shehryar	7 Nov 2019	Added Overall System Description
1.0	Mehdi	8 Nov 2019	Completed Functional Requirements
1.0	Hussain	12 Nov 2019	Added External Interface Requirements
2.0	Hussain	25 Nov 2019	Edited External Interface Requirements
2.0	Hussain	26 Nov 2019	Edited Non Functional Requirements
2.0	Shehryar	27 Nov 2019	Edited Overall System Description

Distribution List

Name	Role
Dr. Muhammad Rafi	Supervisor

Document Sign-Off

Version	Sign-off Authority	Sign-off Date
1.0	Dr. Muhammad Rafi	12 Nov 2019
2.0	Dr. Muhammad Rafi	27 Nov 2019

Document Information

Category	Information
Customer	FAST-NU
Project	Automation of Minutes of Meeting
Document	Software Requirement Specification
Document Version	2.0
Status	Final Version
Author(s)	Mehdi Raza Rajani, Shehryar Naeem, Hussain Yousuf
Approver(s)	Dr. Muhammad Rafi
Issue Date	27 November 2019
Document Location	
Distribution	Advisor, Project Coordinator's Office (through Advisor)

Contents

1 Introduction

1.1 Purpose

This document presents a detailed description of the Minutes of Meeting. It will explain the purpose and features of the system, the interfaces of the system, what the system will do, the constraints under which it must operate and how the system will react to external stimuli. This system will automate the whole task of the minutes of the meetings.

1.2 Intended Audience

This document is intended for Supervisors, team members, the stakeholders and the developers of the system.

1.3 Abbreviations

MOM	Minutes of Meeting
EMA	Enterprise Meeting Assistant
DB	Database
ERD	Entity Relationship Diagram

1.4 Document Conventions

Latex document class: scrreprt

2 Overall System Description

2.1 Project Background

Every enterprise, either small or large, organize meetings daily. For instance, in a software house following the agile sort of software management methodology, daily scrum meetings are held to discuss the progress and further plan of action. The minutes of the meeting method is the way to note down the key insights of the meeting, including the previous development and proposals for an agenda. The primary purpose of this project is to not only automate this essential process of enterprise meetings but to present it in an improved manner as compared to human power. This application will tend to provide these minutes of the meeting concerning the terminologies the enterprise is built. The main focus of this application is to minimize the exploitation of human energy in doing an excessive amount of work daily for creating minutes of the meeting.

2.2 Project Scope

MOM will provide a complete automated solution for the professional world. As a system, MOM will take audio-video recordings and meeting's document as an input with some metadata supporting the meeting points. MOM will also require the agenda of the meeting and its participants. In this manner, the MOM can target generating the minutes concerning the terminologies of the meeting in a contextual way. The instruments to capture audio and video data are to be pre-installed in the meeting room. Whenever an individual starts to speak the system detects it through the gestures and starts to record. This system will be an amalgamation of relevant Computer Vision, Machine Learning, and Natural Language Processing techniques. For instance, to detect who is speaking an efficient Computer Vision Algorithm is required. Similarly, Machine Learning and Natural Language processing techniques will provide near-optimal contextual information. The three mediums of data, namely audio, video, and meeting documents, will validate the results. This data will minimize the inaccuracy of the meeting's minute generation process. MOM will provide several additional facilities that include speaker wise contribution, highlighting the key terminologies, and sharing the minutes with colleagues in one click.

2.3 Not In Scope

2.4 Project Objectives

- Transcribe audio to text
- Clustering according to agendas
- Classification of tasks
- Extracting important features from the text
- Minutes of the meeting generation

2.5 Stakeholders

People will use this system in every small to large organizations.

2.6 Operating Environment

The system will be a web portal that can be accessible from every device's browser.

2.7 System Constraints

- The system will require an active Internet.
- The system will transcribe in English language.

2.8 Assumptions and Dependencies

- The system will have an active Internet connection.
- The system depends on Google's and Amazon's speech to text API.

3 External Interface Requirements

3.1 Hardware Interfaces

Handheld devices will record meeting minutes (audio and video), the recording will be then sent to a server via web interface, the audio and video data will be converted to formats compatible with the third party API's which includes mp3, mp4, wav, FLAC etc and will be then sent to the pertaining service.

3.2 Software Interfaces

The external software components include the following:

- Google and Amazon APIs for transcription and annotation of speech data.
- Ubuntu 18.04 LTS as an operating system
- Nodejs 12.13 LTS as runtime environment, server side
- MongoDB 4.2 as database
- Python 3.7 for data pre-processing and NLP

The server will cater all the functional requirements by communicating with third party services, local services, and database and then passing the relevant results back to the application.

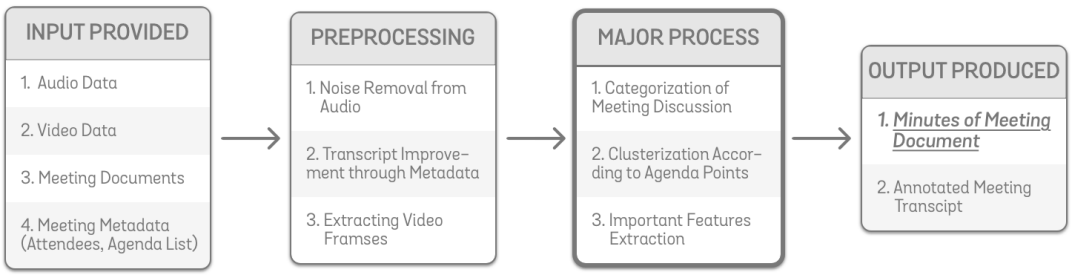
The application and server will communicate via RESTful apis using the JSON standard, which ensures interoperability between all components. Services of the third party may be invoked via HTTP queries, and application data will be sent in character format UTF-8 in JSON except if the data is binary.

3.3 Communications Interfaces

All communications will be done via HTTP/HTTPS protocol, which ensures data confidentiality and integrity. The transfer rate will depend solely upon the client's internet connection, the choice of protocol has no effect on the overall responsiveness of the application.

4 Functional Requirements

4.1 Functional Hierarchy



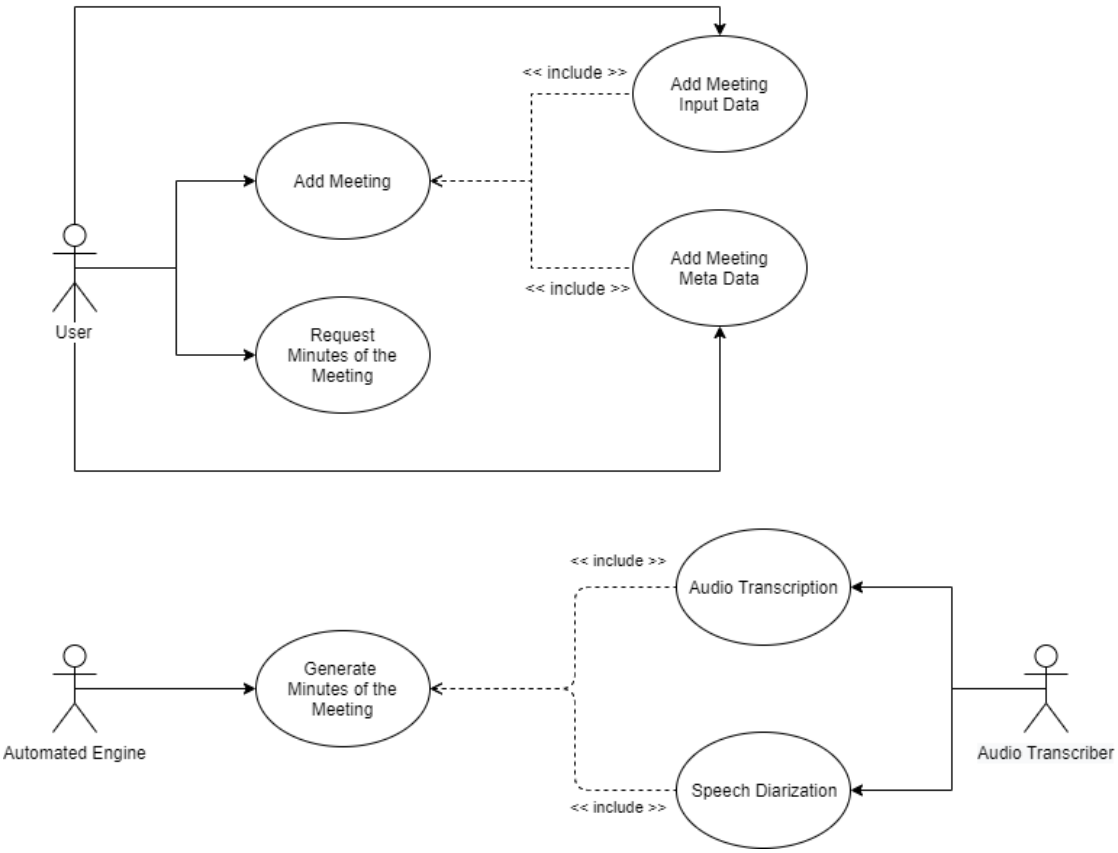
First, the user will log in to the website (meeting agent). Then it will add the meeting and provide all the meeting's input data which includes audio recording, video recording (optional), the document presented in the meeting (optional) and some metadata of meeting that includes the date, time, venue of the meeting, agenda list of meeting and attendees of the meeting.

Then the user will request minutes of the meeting document.

In response, an automated engine will start working. First, the data will be pre-processed and the audio transcriber will send an annotated transcript (speech to text and speech diarization) to the engine. Then the transcript will be categorized according to the meeting's discussion and clustered according to the meeting's agenda. Then important features will be extracted and formatted to get a first version of the Minutes of Meeting document.

Then the user will proofread the document and if he finds some discrepancy so it will make the changes and will finally get the Minutes of the Meeting Document.

4.2 Use Cases



The description of the Use Case Diagram is presented in the following subsection.

4.2.1 Use Case: Add Meeting

Use Case 1	**Add Meeting**
Use case Id:	User1
Actors:	User
Feature:	User wants to request the minutes of meeting document.
Pre-conditions:	• User is already registered. • User is signed in the system.

Main Success Scenario:

1. The user selects the date and time of the meeting.
   {Will be represented on screen.}
2. The user enters the venue of the meeting.
   {Will be represented on screen.}
3. The user selects the attendees from the list.
   {Will be represented on screen.}
4. The user will add the agenda list for the meeting.
   {Will be represented on screen.}
5. The user will add the attached document which will be presented in the meeting.
   {The progress bar will be shown to track the uploading document. Afterward, it will be represented on screen.}
6. The user can add the video of the meeting.
   {This is an optional step}
7. The user will add the audio of the meeting.
   {The progress bar will be shown to track the uploading audio. Afterward, it will be represented on screen.}

Alternate Scenarios:

2.a Venue not present:

- Venue will be added to database.

3.a Attendee name is present:

- User can select it.

3.b Attendee name is not present:

- Name will be added to database.
- User can select it.

Post Conditions:	• Users can now generate the request for generating the minutes of the meeting.

Use Case Cross referenced	User2

4.2.2 Use Case: Request Generation of Minutes of Meeting

Use Case 1	**Request Generation of Minutes of Meeting**
Use case Id:	User2
Actors:	User
Feature:	User wants to request the minutes of meeting document.
Pre-conditions:	• User is already registered. • User is signed in the system. • The meeting is added with all input and meta-data.
Main Success Scenario: 1. The user will request the generation of minutes of the meeting. {Activity Indicator will be presented, to show that the process is running in the background. Afterward, the document will be shown on screen.}	
Post Conditions:	• Automation Engine will start processing and generate the document.
Use Case Cross referenced	• User1 • Server2

4.2.3 Use Case: Generate Minutes of the Meeting

Use Case 1	**Generate Minutes of the Meeting**
Use case Id:	Server1
Actors:	• Server (Automation Engine) • Audio Transcriber • User
Feature:	Automation Engine will generate Minutes of Meeting automatically.
Pre-conditions:	• The user has initiated document generation request of the legitimate meeting record.

Main Success Scenario:

1. The server will ask audio transcribers to do speech to text and speech diarization.
   {The server will wait until the annotated transcript is generated. Pass the resultant transcription for further processing. The annotated transcript will also be shown to user.}
2. The server will then perform categorization and clustering according to the meeting's agenda.
   {The server will wait until the categorization is performed. Then it will pass the categorized data to be clustered according to agenda points. Then the clustered data will be passed for further processing.}
3. The server will then extract important features from each agenda's data.
   {The server will until important features are extracted. Later it will pass for the final step.}
4. The formatted document will be generated.
   {The document will be shown to user.}
5. The user will check it confirm the document.
   {System will accommodate the changes done by user and show the final copy of the Minutes of Meeting document.}

Alternate Scenarios:

5.a No changes made by User:

- Add the document to database.
- Show the final version of document to user.

5.b User finds some discrepancy:

- Allow user to make changes.
- Add the document to database.
- Show the final version of document to user.

Use Case Cross referenced User2

5 Non-functional Requirements

5.1 Performance Requirements

The web server should be able to handle multiple users in parallel, also the response time should not be too long and the server should provide periodic notification regarding the progress. The system expects that the user will have a standard Internet connection. Since the application depends on third parties, also the system assumes that third party services will also be consistent.

5.2 Safety Requirements

The web server and database are expected to have enough capabilities to prevent data breaches and to provide periodic backups. User profiles shall not be disclosed to any third party, and minimal pertaining information of the user will be kept.

5.3 Security Requirements

The application should communicate with a remote server using a secure socket protocol. This ensures the confidentiality, integrity, and nonrepudiability of the payload. Furthermore, all data on the database shall be encrypted and third parties would not be allowed to persist or distribute data.

5.4 User Documentation

The application should include elaborate and intuitive controls that will enable the user to use the application without expert knowledge of the domain. However, instructions should be included in the application for completeness's sake.