Software Requirements Specification
Version 1.0

TalkToText Pro

Theme: AI-Powered Meeting Notes Rewriter

Project Name: TalkToText Pro

Category: Generative AI Odyssey

Contents

1.1 Background and Necessity for the Application

With the rise of remote work, online learning, and virtual collaboration, meetings have become increasingly common. Platforms such as Microsoft Teams, Google Meet, and Zoom now play an integral role in professional and academic environments. Despite this shift, the process of documenting meeting discussions, action points, and decisions remains largely manual and error-prone. Participants often struggle to take accurate notes while actively engaging in the conversation, leading to missed information, inconsistencies, and a lack of reliable documentation. As a result, organizations and teams are increasingly looking for automated solutions that can transcribe and summarize meeting content efficiently and accurately.

To address this requirement, a solution must be created that can convert speech into meeting notes. With the emergence of AI tools and technologies, utilizing them to build this solution is appropriate.

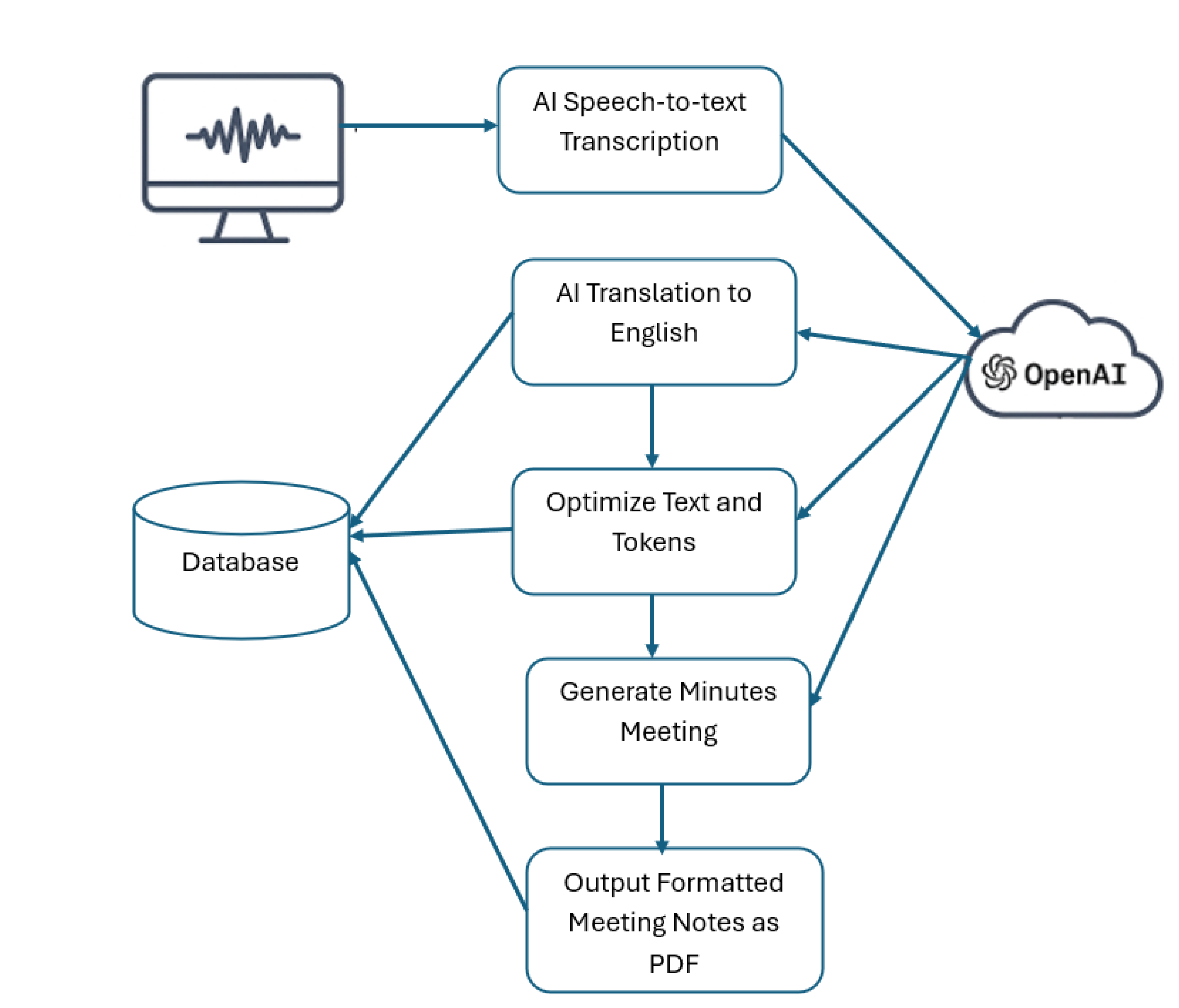
1.2 Proposed Solution

‘**TalkToText Pro**’ emerges as a timely solution to this challenge by offering an AI-powered system that converts speech into well-structured and actionable meeting notes. By combining speech-to-text technology, language translation, and advanced summarization using models such as OpenAI’s ChatGPT, the application ensures that key meeting insights are captured, organized, and easily accessible. This automation not only saves time and effort, but also improves communication clarity, enhances follow-up efficiency, and supports multilingual collaboration. As teams grow more distributed and meetings become more frequent, the necessity for such intelligent documentation tools becomes essential for maintaining productivity, accountability, and institutional memory.

TalkToText Pro offers an end-to-end AI-powered system designed to automate the process of converting meeting audio into clean, structured, and actionable notes. The application begins by accepting audio input from recorded or live online meetings conducted via platforms such as Microsoft Teams, Google Meet, or Zoom. It uses advanced AI-based speech-to-text transcription to accurately convert spoken content into written form. If the meeting is conducted in a language other than English, the system automatically translates the transcript into English using a reliable language translation engine.

Once the transcript is available, the text is cleaned and token-optimized to prepare it for summarization. The core of the solution lies in using OpenAI’s ChatGPT API, which processes the optimized transcript and generates comprehensive meeting minutes. These include an executive summary, key discussion points, action items, decisions made, and even sentiment analysis, if required. The final output is formatted for clarity and can be downloaded, shared via email, or integrated into productivity tools. All versions of the transcript and notes are stored in a database for future retrieval and reference. This ensures that no important information is lost, and all meetings are documented with consistency and precision.

The architecture for building the application will be as follows:


AI Speech-to-text Transcription
AI Translation to English
OpenAI
Optimize Text and Tokens
Database
Generate Minutes Meeting
Output Formatted Meeting Notes as PDF

Sample Architecture of the Application

The steps to create the application are as follows:

1. **Online Meeting Input** - The system receives a meeting recording or live audio stream from tools such as Google Meet or Microsoft Teams.
2. **AI Speech-to-Text Transcription** - The audio is converted to text using AI-based speech recognition APIs (example, Whisper or Google Speech-to-Text). OpenAI or other speech models can be plugged in here for accurate transcription.
3. **AI Translation to English** - If the transcript is in another language, use AI translation tools to convert it to English. AI services such as Google

Translate API or OpenAI language translation models can be used for translation.

4. **Optimize Text and Tokens** - The raw/translated text is cleaned (such as removing noise, filtering pauses, repetitions, filler words). Token optimization ensures the content fits within OpenAI API token limits, which is important for longer transcripts.
5. **Generate Meeting Minutes (OpenAI API Call)** - The optimized transcript is passed to ChatGPT via OpenAI API, prompting it to extract:
   - Summary
   - Key discussion points
   - Decisions
   - Action items
   - Sentiment (positive/negative tone)
6. **Store in Database** - All versions which include raw, translated, optimized, final notes are stored in a relational or NoSQL database (example, MongoDB).
7. **Output Formatted Meeting Notes** - Final notes can be exported as PDF or Word reports.

1.3 Purpose of the Document

The purpose of this document is to present a detailed description of the AI-powered meeting transcription and summarization system titled **'TalkToText Pro**.'

This document explains the purpose and features of the application, which uses Generative AI technologies such as speech-to-text, language translation, and Large Language Models (LLMs). These technologies work together to transform meeting audio into structured, actionable notes. It also outlines the constraints under which the system should operate. This document is intended for both stakeholders and developers of the application.

1.4 Scope of Project

The '**TalkToText Pro**' application should be able to assist users in processing meeting audio recordings from various sources such as Microsoft Teams, Google Meet, Zoom, and other conferencing platforms. It will support multiple audio formats and languages to ensure broad usability across professional and academic settings.

Key features will include AI-powered speech-to-text transcription, automatic language translation, text optimization, and the use of Generative AI models for generating structured meeting notes. These notes will include summaries, key discussion points, action items, and sentiment analysis. A user-friendly interface and secure database storage will be employed to enable users to upload, view, download, and retrieve meeting notes, ensuring an efficient, consistent, and intelligent meeting documentation experience.

1.5 Constraints

Constraints for the '**TalkToText Pro**' project could include the requirement for accurate transcription of audio files with varying quality, background noise, multiple speakers, and accents. These factors may impact the effectiveness of speech-to-text conversion. Ensuring consistent language translation and summarization across diverse meeting contexts is also a potential challenge. Additionally, security and privacy concerns when handling sensitive meeting data will be critical. This will require strict adherence to data protection regulations and the implementation of secure data storage and processing practices throughout the application's lifecycle.

1.6 Functional Requirements

The project involves developing an AI-Powered Meeting Notes Rewriter application utilizing Generative AI, LLMs, and technologies from OpenAI. The application will transform raw meeting transcripts into clean, structured notes, highlighting key decisions, action items, and takeaways.

Some of the functional requirements are explained as follows:

i. **User Interface (UI) for Audio Input** - The application should provide a UI that allows users to upload meeting audio files in supported formats such as .mp3, .wav, or .mp4. It should also enable users to input links from online meeting platforms such as Microsoft Teams or Google Meet.

ii. **Language Selection** - The application should provide an option for users to select the original language of the uploaded audio.

iii. **Speech-to-text Transcription -** The application should transcribe the uploaded audio into text using an AI-based speech recognition service.

iv. **Language Translation** - The application should automatically translate the transcript to English if the original language is not English.

v. **Text and Token Optimization** - The application should clean and optimize the transcribed text by removing filler words, repeated content, and adjusting token count to comply with the token limits of the OpenAI API.

vi. **Meeting Notes Generation using OpenAI** - The application should use the OpenAI API to generate structured meeting notes that include a summary, key points, decisions, action items, and sentiment analysis.

vii. **Display of Processed Output** - The application should display the generated meeting notes in a readable, structured format on the UI.

viii. **Download and Export** - The application should provide options to download the meeting notes in PDF or Word format and share them via email or other integrations.

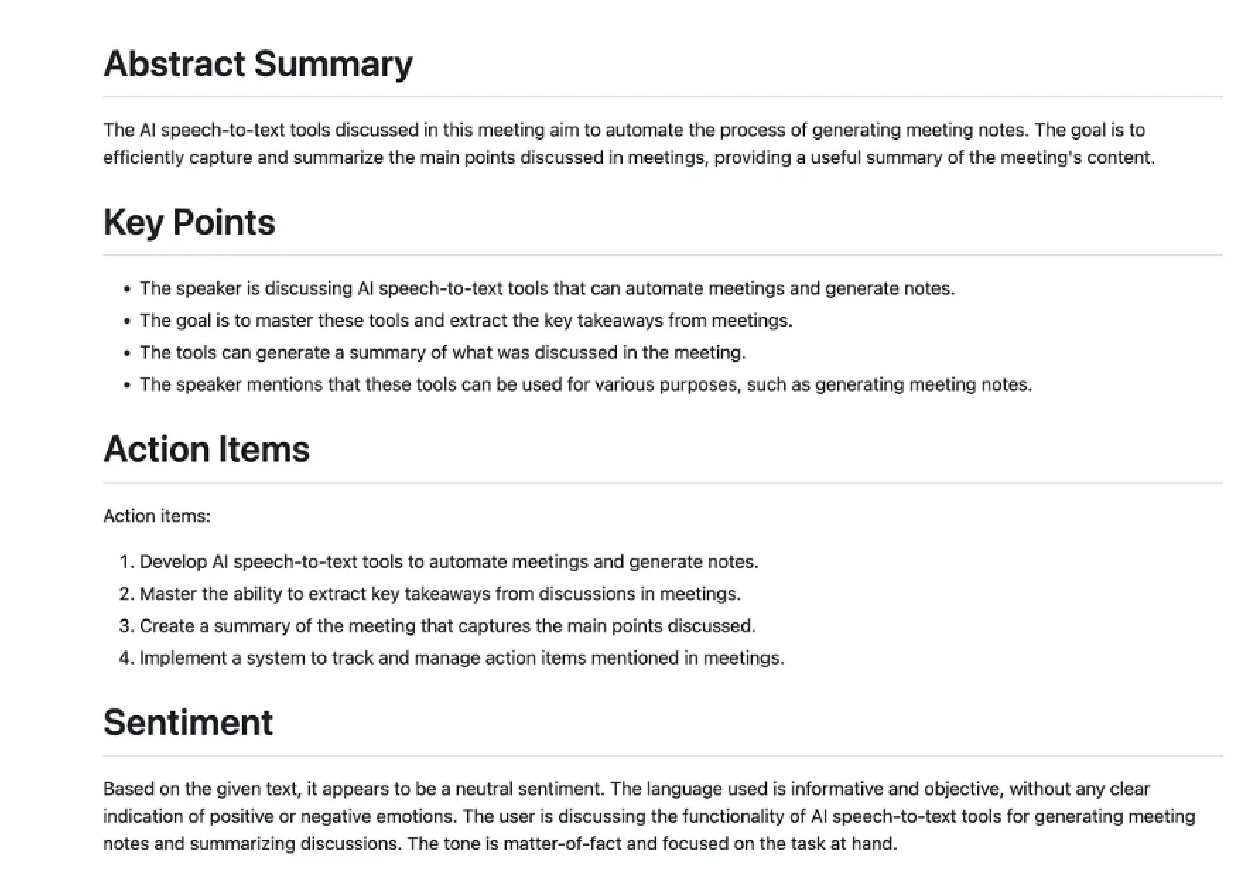
ix. **Data Storage** - The application should store the raw transcript, translated version, optimized text, and final meeting notes in a database.

x. **View History** - The system shall allow users to view and retrieve previously generated meeting notes from the database.

xi. **Progress Tracking UI** - The system shall show real-time processing status updates for each step (transcription, translation, optimization, generation) through a progress tracker on the UI.

In addition, the application should include a login system to enable users to manage their own meeting records and access history securely.

Following is a sample of an expected output:



Abstract Summary

The AI speech-to-text tools discussed in this meeting aim to automate the process of generating meeting notes. The goal is to efficiently capture and summarize the main points discussed in meetings, providing a useful summary of the meeting's content.

Key Points

- The speaker is discussing AI speech-to-text tools that can automate meetings and generate notes.
- The goal is to master these tools and extract the key takeaways from meetings.
- The tools can generate a summary of what was discussed in the meeting.
- The speaker mentions that these tools can be used for various purposes, such as generating meeting notes.

Action Items

Action items:

1. Develop AI speech-to-text tools to automate meetings and generate notes.
2. Master the ability to extract key takeaways from discussions in meetings.
3. Create a summary of the meeting that captures the main points discussed.
4. Implement a system to track and manage action items mentioned in meetings.

Sentiment

Based on the given text, it appears to be a neutral sentiment. The language used is informative and objective, without any clear indication of positive or negative emotions. The user is discussing the functionality of AI speech-to-text tools for generating meeting notes and summarizing discussions. The tone is matter-of-fact and focused on the task at hand.

1.7 Non-Functional Requirements

There are several non-functional requirements that should be fulfilled by the application. These are listed as follows:

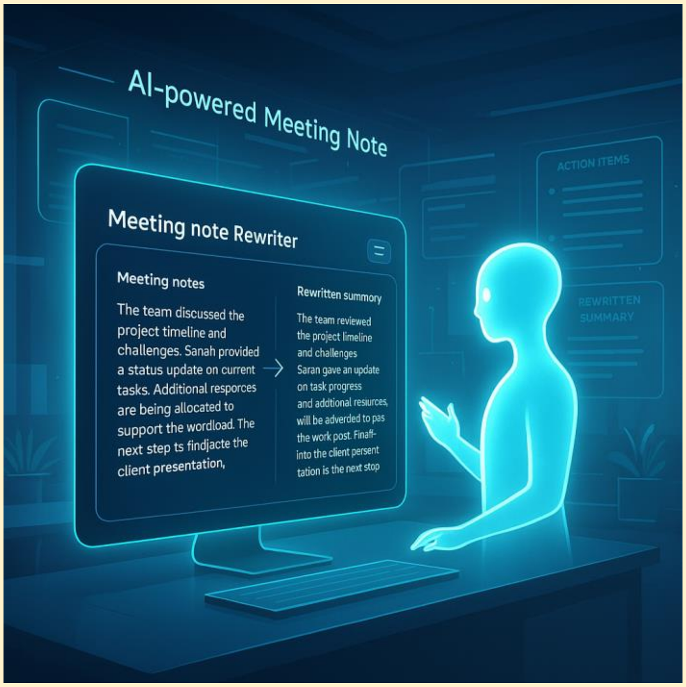
1. **Performance**: The application must process a 30-minute meeting recording and generate notes within one to two minutes.

2. **Secure**: Meeting data and summaries must be securely stored with access restrictions.

3. **Accuracy**: Transcription and summarization must achieve at least 85–90% accuracy under good audio conditions.

4. **Scalable**: The application should be scalable to handle multiple meeting uploads simultaneously.

5. **Usable**: The UI must be simple, intuitive, and mobile-responsive for both technical and non-technical users.



The team discussed the project timeline and challenges. Sanah provided a status update on current tasks. Additional resporces are being allocated to support the wordload. The next step ts findjacte the client presentation,

The team reviewed the project limeline and challenges Saran gave an update on task progress and addtional resiurces, will be adverded to pas the work post. Finaff- into the client persent tation is the next stop

These are the bare minimum expectations from the project. It is a must to implement the FUNCTIONAL and NON-FUNCTIONAL requirements given in this SRS.

Once they are complete, you can use your own creativity and imagination to add more features if required.

1.8 Interface Requirements

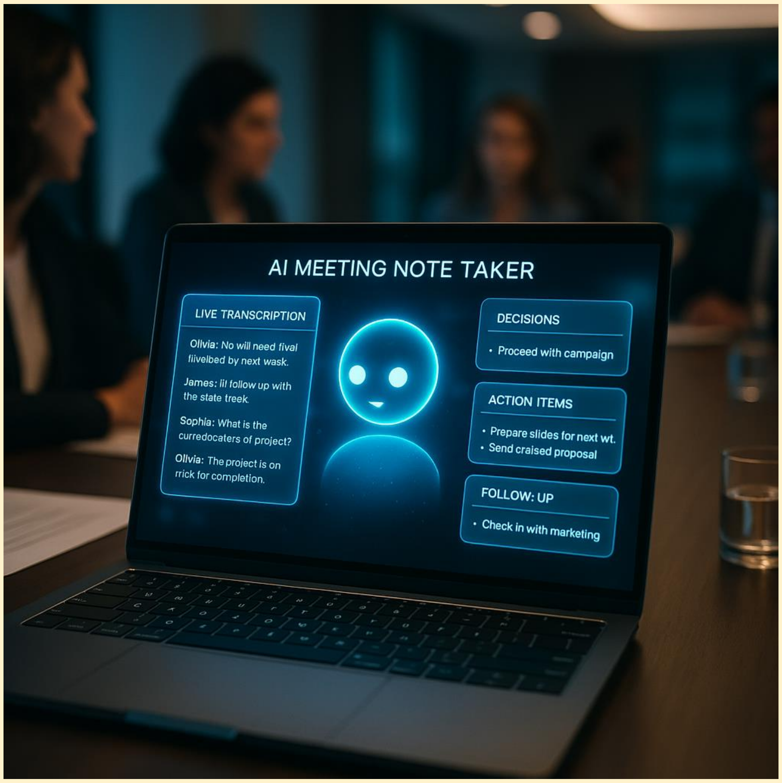
1.8.1 Hardware

Intel Core i5/i7 Processor or higher
8 GB RAM or higher
Color SVGA
500 GB Hard Disk space Mouse
Keyboard

1.8.2 Software

Technologies to be used:

1. **Frontend**: HTML5 or any other frontend programming languages
2. **Backend**: Flask/Django
3. **Data Store**: PDF, MongoDB
4. **Programming/IDE**: Python, Jupyter Notebook, Anaconda, Google Colab
5. **Libraries**: Tensorflow, NLTK, Keras, OpenAI API, Python libraries, and transformers


AI MEETING NOTE TAKER
LIVE TRANSCRIPTION
Olivia: No will need fival fiivelbed by next wask.
James: lil follow up with the state treek.
Sophia: What is the curredocaters of project?
Olivia: The project is on rrick for completion.
DECISIONS
• Proceed with campaign
ACTION ITEMS
• Prepare slides for next wt.
• Send craised proposal
FOLLOW: UP
• Check in with marketing

1.9 Project Deliverables

You will design and build the project and submit it along with a complete project report that includes:

- Problem Definition
- Design specifications
- Diagrams such as User Flow Diagram/User Journey Map
- Detailed steps to execute the project
- Test Data Used in the Project
- Project Installation Instructions
- Link of GitHub for accessing the uploaded project code (Link should have public access)
- Link of published blog

The source code, including .ipynb files for Jupyter Notebook and Google Colab, should be shared via GitHub. Appropriate access permissions should be granted to users to allow testing for Jupyter Notebook and Google Colab. The consolidated project must be submitted on GitHub with a ReadMe.doc file listing assumptions (if any) made at your end.

Kindly provide the GitHub URL where the project has been uploaded for sharing. The repository on GitHub should have public access. Documentation is a very important part of the project; hence, all crucial aspects of the project must be documented properly. Ensure that documentation is complete and comprehensive.

Do NOT copy content or code from GPTs or other AI tools, although you are permitted to use images generated by AI tools for any visual representation purposes. It is mandatory to mention such tools used in case you add any AI generated images.

Submit a video (.mp4 file) demonstrating the working of the application, including all the functionalities of the project. This is MANDATORY.

*~~~ End of Document ~~~*