SmartMeetings: Product Requirements Document

Team Name: Meeting is Believing
Team Members: Maxton Ginier (Lead), Tuan Le, Jackson Li (Scribe), Sarita Phoosopha, Christina Tao

Introduction

Problem
Meetings are crucial for any line of work. They are necessary for communication between colleagues about ideas, problems, roadblocks, and more. However, repetition of discussions and materials from previous meetings is a downfall that causes present-day meetings to be inefficient and unnecessary. Other common problems in meetings are undistributed speaking time, groupthink, members absence, and run-on meetings. People may also forget what was discussed, either because they are unable to take notes, or they are unable to pay full attention to what is being said because they are too focused on taking notes.

The problem is currently being solved using audio transcription by various companies. However, the processing and analysis of the transcription is lacking. Most applications focus on having a simple transcription of the meeting afterwards, which catches absent members up when necessary, but does not include real-time information that can benefit current, as well as the future, meetings. The simple transcription also does not help absent, as well as available members, to find out the important tasks at hand, as well as questions that each member may have during the meeting.

Innovation
This project will produce a tool that makes meetings more productive and collaborative by providing a more meaningful analysis of the meeting. The application will prompt pre-meeting questions about the type of meetings (stand-up, long board meetings, etc.) and the estimated or desired meeting time of the meeting. The application will then use the responses to predict the pacing and runtime of the meeting. Additionally, audio transcriptions will be used to detect off-topic conversations, and video analysis will detect restless behaviours, such as fidgeting. Such observations will be reported both in real time during the meeting and in a summary after the meeting in order to increase

productivity in future meetings and give personal feedback to the members of the meeting. The transcriptions and video analysis will allow the tracking of meeting progress and emotional sentiments as well. Other functions include annotated transcript, note-taking, and automatic to-do list creation based on spoken keywords are employed to provide a well-rounded summary of the meeting.

Science

The technologies that will be utilized to solve the problem and accomplish such innovation are audio recognition through Watson IBM Speech-to-Text API and natural language processing through Python's Natural Language TookKit (NLTK). Furthermore, React with Materials UI hosted on Heroku will be utilized in order to provide an intuitive interface for users, whose data will be stored in a DynamoDB database. WebRTC will provide users with real-time communication, and the audio and video files will be stored on Amazon S3. Everything will be integrated together using the backend framework Ruby on Rails. Finally, Docker will be used for the container platform to ensure portability and enable developers to work on their own machines without having to worry about dependencies, libraries, and configuration files.
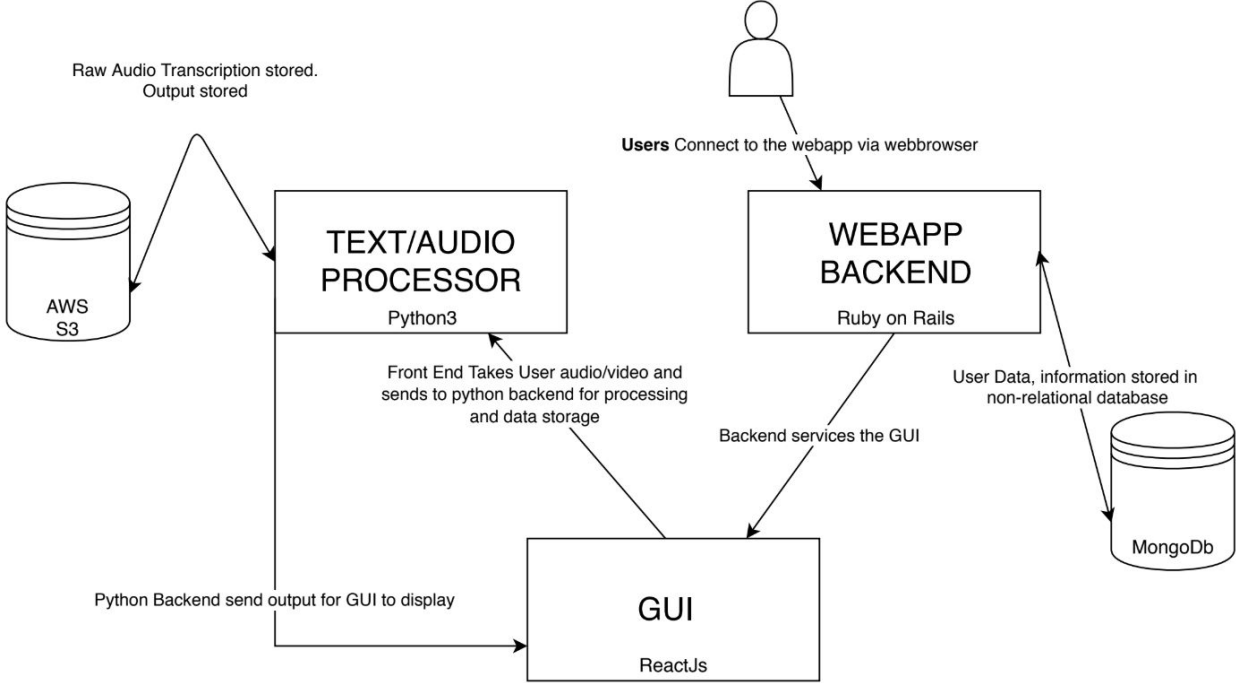
Goals and Objectives

The minimum viable product will allow users to create an account in order to upload audio recordings of meetings to obtain a transcription and a summary of the transcription. The goal is to have a web application where a user can log in to their account, answer questions such as the topics that should be discussed and estimated time of the meeting, and then make a video call. After the meeting, the users will be able to see the total time and whether all the topics listed in the pre-meeting questionnaire were discussed. The users will also be able to navigate to a page where they can see the whole transcription, a summary of the transcription, questions asked, and tasks that were mentioned. Moreover, a reach goal is to detect emotional sentiments using video analysis. The data will allow users to evaluate the environment of the meeting for example, improving a tense meeting, or take note of how to maintain a positive meeting.

In order to achieve the goal, the team will follow the Agile software development methods, including two week sprints, daily scrums, and continuous re-analysis of the requirements. All tasks will be kept track of on Trello, and Burndown Charts will be created to keep track of weekly progress.

System Architecture Overview

High Level Diagram

Raw Audio Transcription stored.
Output stored

Users Connect to the webapp via webbrowser

AWS
S3

TEXT/AUDIO
PROCESSOR

Python3

WEBAPP
BACKEND

Ruby on Rails

Front End Takes User audio/video and
sends to python backend for processing
and data storage

User Data, information stored in
non-relational database

Backend services the GUI

MongoDb

Python Backend send output for GUI to display

GUI

ReactJs

Requirements

User Stories

1. As a user, I can input an audio file to the application so that I can obtain a summary of the audio without relying on real-time technologies.
   a. Acceptance: Text file output upon Audio file input. Makes sense when it's translated.
   b. https://github.com/maxtong9/smart_meetings/pull/23
   c. https://github.com/maxtong9/smart_meetings/pull/13
2. As a developer, I can pull a Docker image to my own machine without having to set up my own local environment.
   a. Acceptance: Developer can pull the Docker image from Docker Hub and run the container without having to download other dependencies or libraries.
   b. https://github.com/maxtong9/smart_meetings/pull/22
3. As a user, I can see a summary of the meeting transcription to view the summary of the meeting without having to review everything that was said.
   a. Acceptance: Input is a transcription, output is a summary of the transcription that is shorter than the raw transcription
   b. https://github.com/maxtong9/smart_meetings/pull/16
   c. https://github.com/maxtong9/smart_meetings/pull/20
   d. https://github.com/maxtong9/smart_meetings/commit/93f235c7f9e3184c3b6f62502279f96d47627900
4. As a developer, I can download and upload files to Amazon S3 in order to access files and push files to storage.
   a. Acceptance: File pushed to S3 by one member and downloaded by another member.
   b. https://trello.com/c/QTazkX0b/32-setup-s3-storage-for-team-members
5. As a user, I can view questions that were asked during the meeting so I do not have to write them down.
   a. Acceptance: User can view questions during the meeting at a later time on the website.
   b. https://github.com/maxtong9/smart_meetings/pull/21
6. As a user, I can create a video call so that I am able to start a meeting.

a. Acceptance: A logged in user can create a call room through the user interface and join it.

7. As a user, I can view in-depth text analysis, such as long pauses or divergence from the main topic, in order to improve the efficiency of future meetings.
   a. Acceptance: User can view text analysis on the website at a later time.

8. As a user, I can make an account in order to access meeting transcription features.
   a. Acceptance: The user can select a "create an account" option to fill in basic information such as name, email, and password.
   b. The user can see their profile page after account creation.

9. As a user, I can view my account details including transcriptions, audio, etc. so that I can see my saved data.
   a. Acceptance: From the profile page, the user can see lists of meetings that they can navigate to to access transcriptions.

10. As a user, I can record my meeting (audio and video) so I can review the meeting at a later time.
    a. Acceptance: User can see recordings in their data and view them through the website.

Current Functions, Issues and Tasks

https://github.com/maxtong9/smart_meetings
https://trello.com/b/JEuyqKwx/smartmeetings

Appendices

Technologies Employed
- GUI: React w/ Material UI
- RTC: WebRTC
- User Database: MongoDB
- Storage: Amazon S3
- Backend Framework: Ruby on Rails
- Web Hosting: Heroku
- Audio Transcription: IBM Watson

- Audio/Text Processing: nltk (Python)
- Container Platform: Docker