



STeMS Backend Services

Version 4.0

Project Name: Short-Term Memory Systems

Date: August 4, 2023

Project Leader: Jonathan Nagy

Project Members: David Babers, Mohamed Ben Lakbir, Robson De Souza, Collin Hicks, Johnny Huynh, Benny Iko, Scott McCrillis, Amol Thomare

Leader Approval: Jonathan Nagy

Date: 8/04/2023

Client Approval:

Date:

Revision History

Date	Version	Description	Author
06/02/2023	1.0	Initial revision	Amol Thomare, Robson De Souza, Collin Hicks, Benny Iko, and Mohamed BEN LAKBIR
06/07/2023	1.1	Milestone 1 resubmission	Robson De Souza Mohamed Ben Lakbir Amol Thomare
06/14/2023	2.0	Milestone 2 (Test Plan)	Mohamed BEN LAKBIR Robson De Souza Johnny Huynh Kidanu Mekonnen Jonathan Nagy
07/06/2023	3.0	Milestone 3	Team B Members
08/04/2023	4.0	Milestone 4	Team B Members

Table of Contents

Contents

1. Problem Definition	5
1.1 Brief History of the Development Project	5
1.2 Abbreviations and acronyms	5
1.3 Statement of Need	7
1.4 Catalyst for the Development Project	8
1.5 Potential Clientele	8
1.6 Scope Management Plan	8
1.6.1 <i>Project In-Scope Activities</i>	8
1.6.2 Project Out-of-Scope Activities	10
1.6.3 Success Criteria	11
1.6.4 Project Assumptions	11
1.6.5 Project Tools	11
2. Project Assignment	12
3. Risk Analysis	13
3.1 Risk matrix	13
4. Communications	17
4.1 Objective	17
4.2 Communication Methods	17
4.3 Stakeholder Analysis	17
4.4 Communication Schedule	17
4.5 Crisis Communication Plan	17
4.6 Feedback Mechanism	18
5. Change Management	19
5.1 Change Control Board (CCB)	19
5.2 Internal Change Requests	19
5.3 External Change Requests	19
5.4 Change Request Form	20
5.5 Requirements Traceability Matrix (RTM)	20
6. Organization of the Project	21
6.1 Project Model	21

6.1.1 Initiation Phase	22
6.1.2 Definition Phase	23
6.1.3 Design Phase.....	25
6.1.4 Development Phase	27
6.1.5 Implementation Phase.....	29
6.2 Project Summary	32
7. Software Test Plan	33
7.1 Purpose	33
7.2 Scope	33
7.3 Considered Requirements	33
7.3.1 To Be Tested	33
7.3.2 Not to be tested	34
7.4 Approach	34
7.5 Pass/Fail Criteria	35
7.6 Defect Severity Levels	35
7.7 Risks and Contingencies	35
7.8 Test Cases.....	36
7.8.1 Recordings API.....	36
7.8.2 Browser Extension API.....	41
7.8.3 ChatGPT Query API: Food Orders	44
7.8.4 ChatGPT Query API: STML Reminders	46
7.8.5 User Profile	48
7.8.6 Reminder Notifications	50
7.8.7 Non-Functional Requirements	51
7.8.8 Future Requirements	53
8. Overview	54
9. Works Cited	57

1. Problem Definition

The following project documentation will be included in the entire documentation package for this project/application:

Table 1. Most Recent Version of Documents

	Document	Version	Date
1	Project Management Plan (PMP)	4.0	8/04/2023
2	Software Requirements Specification (SRS)	4.0	7/24/2023
3	Technical Design Document (TDD)	3.0	7/24/2023
4	Programmers Guide (PG)	2.0	7/24/2023
5	Deployment and Operations (DevOps)	2.0	8/04/2023
6	User Guide (UG)	1.0	8/05/2023
7	Test Report (TR)	1.0	8/05/2023

1.1 Brief History of the Development Project

Building off previous capstone projects by former UMGC graduates, our team has been tasked with creating an application that assists individuals that suffer from short term memory loss (STML) where it may affect important parts of their everyday life. Utilizing the prevalence of smartphones in nearly everyone's possession, audio recordings will be the primary mode of data collection for the application. Data captured within the audio recordings will be the basis for analysis through speech-to-text services that ultimately link to an artificial intelligence (AI), in this case ChatGPT, to add value to the application's functionalities.

By leveraging AI technology, the user will have a powerful assistant with the potential to recall forgotten details from conversations, create events or notifications from a previous interaction, or make sense of heavy jargon so anyone can understand the context and meaning of a conversation.

1.2 Abbreviations and acronyms

This section covers all abbreviations and acronyms used throughout this project management plan.

Table 2. Abbreviations and acronyms throughout project plan

Abbreviation	Description
AI	Artificial Intelligence

API	Application Programming Interface
ASR	Automatic Speech Recognition
BA	Business Analyst
BESie	Browser Extension Service
CCB	Change Control Board
DOG	Deployment and Operation Guide
FR	Functional Requirement
iOS	iPhone Operating System
IT	Information Technology
LLM	Large Language Model
LTM	Long-Term Memory
MCI	Mild Cognitive Impairment
NFR	Non-Functional Requirement
RACI	Responsible, Accountable, Consulted, Informed
OS	Operating System
PG	Programmer Guide
PM	Project Manager
PII	Personally Identifiable Information
PP	Project Plan
QA	Quality Assurance
SDLC	Software Development Life Cycle
SME	Subject Matter Expert
SRS	Software Requirement Specifications
STeMS	Short-Term Memory System
STM	Short-Term Memory
STML	Short-Term Memory loss
STP	Software Test Plan

STT	Speech-To-Text
TDD	Technical Design Document
TL	Team Lead
TM	Team Member
TR	Test Report
UAT	User Acceptance Testing
UG	User Guide
UI	User Interface
UMGC	University of Maryland Global Campus

1.3 Statement of Need

Short-term memory (STM), short-term storage, indicates different systems of memory involved in the retention of pieces of information for a relatively short time. It usually lasts up to thirty seconds, while long-term memory (LTM) may hold an indefinite amount of information (Marco & Khalili, 2022). Short-term memory loss (STML), or short-term memory impairment, occurs when someone cannot retain information in the short term, or forgets information he or she has just received. It is easy to forget new information if someone is inattentive, distracted, or heedless. Yet, people must be aware that mild forgetfulness is also a typical part of aging. Despite that, when they fail to recall vital information frequently enough to interfere with their ability to function in daily life, they may be suffering short-term memory loss related to a specific health condition.

According to Healthline (2021), the common circumstances allied with STML include: aging, neurodegenerative diseases, head injury, infection, tumors, stroke, cardiac arrest, medications, sleep deprivation, epilepsy, and chronic pain conditions. STML signs for affected persons include:

- Frequently asking someone's name or for the same information.
- Being confused about what they are doing, who they are with, and what time or day it is.
- Struggling to recall or understand information that has been just received (directions).
- Overlooking recent experiences or events.
- Failing to recall where they put things.
- Forgetting information that they lately heard.

To support those that undergo STML or diagnose cognitive decline. At UMG, the IT department proposes the development of a mobile application that meets the need.

1.4 Catalyst for the Development Project

There are many people who suffer from short-term memory loss due to a variety of reasons. These people must deal with losing details of their life at any given moment if they do not spend the time to commit it to their long-term memory properly. The team was hired by the client to create an app that can help the users record and iterate information back to the app users to assist with the users who suffer from this condition.

1.5 Potential Clientele

Anyone who suffers from memory loss, whether it be short-term, long-term, or another medical condition, such as Alzheimer's disease or dementia. Aside from these groups which would clearly benefit the most, other user groups, such as busy individuals that may not have the time to dictate notes from conversations or unorganized individuals that cannot interpret their own notes due to sloppy handwriting or hastily taken notes, could have potential benefits.

1.6 Scope Management Plan

This section of the project plan document describes how the scope of the project will be defined. The parties responsible for successful completion of the project will be included as well as the roles they will fulfill.

1.6.1 Project In-Scope Activities

This section defines the boundaries of the responsibilities of the team during this project. To avoid scope creeps, this list is intended to be used as a basis for the tasks and intermediate steps necessary to build a successful product. The final product will be able to:

- Create / Read / Update / Delete / Retrieve metadata for each recording
- Search through recording data
- Create / Read / Update / Delete Retrieve list of recordings
- Analyze recording data and convert audio to text using Speech-to-Text
- Create / Retrieve a short code for the Browser Extension to identify ConvoBuddy application from local storage
- From a web browser, text field names will be collected and sent in a query to ChatGPT to fill out the information based on an analysis of the recorded data
- Based on a textual analysis of the recordings, a query will be generated and sent to ChatGPT to create a food order
- Based on a textual analysis of the recordings, a query will be generated and sent to ChatGPT to determine if audio contained any relevant reminders
- Create and send an arbitrary query to ChatGPT to provide a tailored response based on an analysis of a group or all recordings stored on the local device
- Create / Read / Update / Delete / Retrieve user profile information
- Create reminders on the local device triggered by time-based events
- Display relevant profile information on the user interface

Figure 1. RACI (Responsible, Accountable, Consulted, and Informed) Chart

	PROJECT TEAM A Members												
	Mir Assadullah, Professor	Roy Gordon, Advisor	Robert Wilson, DevOps	David Babers, PM	Jonathan Nagy, TL	Amol Thomare, TM	Scott McCrillis, TM	Robson De Souza, TM	Mohamed Ben Lakbir, TM	Johnny Huynh, TM	Kidanu Mekonnen, TM	Benny Iko, TM	Collin Hicks, TM
PROJECT TASKS	CLIENT	MENTORS		PROJECT TEAM B Members									
Milestone 1													
Project Plan	C	C	C	A	A	R	A	R	R	A	R	R	A
Software Requirements Specification (SRS)	C	C	C	R	R	A	R	A	A	R	A	A	R
Presentation 1	C	C	C	R	R	R	R	R	R	R	R	R	R
Milestone 2													
Technical Design Document (TDD)	I,C	C	C										
Software Test Plan (STP)	I,C	C	C										
Updated Milestone 1 Documents	I,C	C	C										
Presentation 2	I,C	C	C										
Milestone 3													
Programmer's Guide (PG)	I,C	C	C										
Deployment and Operations Run Book/Guide (DOG)	I,C	C	C										

Initial Software	I,C	C	C										
Initial Testing	I,C	C	C										
Updated Milestone 1 and 2 Documents	I,C	C	C										
Presentation 3	I,C	C	C										
Milestone 4													
User Guide (UG)	I,C	C	C										
Test Report (TR)	I,C	C	C										
Updated Milestone 1, 2, and 3 Documents	I,C	C	C										
Finished Software	I,C	C	C										
Presentation 4	I,C	C	C										

Legend

R	Responsible
A	Accountable
C	Consulted
I	Informed

1.6.2 Project Out-of-Scope Activities

This section covers the activities that are not included in the project scope thus the team is not responsible for it. From the project “STeMS” perspective, the out of-scope activities include and are not limited to the following:.

- Usage of another private Large Language Model (LLM) over ChatGPT.
- Usage of a diarization software to work with application.
- Additional functionality such as calling emergency services, providing suggestions based on recent transcriptions, and etc.

1.6.3 Success Criteria

At project completion, success criteria are all parameters by which a project will be evaluated to decide whether it has been successful based on stakeholders. Success criteria are vital and should be considered beforehand and focused on all the time. In the studying project, success criteria consist of:

- Accomplish all milestones and deliverables on time.
- Deliver all project documentation to stakeholders.
- Deploy a final product that covers the high-level requirements.
- Stay within budget.
- Make sure communication with stakeholders is frequent throughout SDLC.
- Deliver a working application.
- Assure positive relationships with all stakeholders.

1.6.4 Project Assumptions

It is impossible to plan any project without few assumptions. Project assumptions include any factor impacting the project, that is considered to be true or certain. For STeMS success, parameters that are presumed to be true are:

- The stakeholders are represented by the Professor Mir Assadullah.
- Team size remains the same throughout SDLC.
- The application must be developed in Flutter and Dart.js as a mobile application to use for both Android and iOS smartphones.
- The application shall be designed to meet the need for STML patients.
- The hourly wage for developers is \$43/hr and \$48/hr for the lead
- The hourly wage for BA is \$40/hr
- The hourly wage for the technical writer is \$500/hr
- The hourly wage for QA is \$40/hr
- The hourly wage for Team B would be \$437/hr

1.6.5 Project Tools

This section enlists the necessary tools to monitor and build a successful STeMS project. It includes:

Table 3. Risk matrix diagram (Tucci, 2023)

Product/Tool	Version
Microsoft Office	Office 365 Apps for Enterprise
Microsoft Teams	Office 365 Apps for Enterprise
Trello	June 6, 2023 (current online version)
GitHub	June 15, 2023 (current online version)
Microsoft Project	Professional 2021
Flutter	3.10.5 (current latest)
Dart	3.0.5 (packaged with Flutter SDK 3.10.5)
VS Code	1.79 (current latest stable)

2. Project Assignment

The deliverable for this project will be a fully functioning mobile application. This application will be able to transcribe vocal inputs given by the user and any nearby person and send this transcription to ChatGPT. From there, the user can ask inquiries about the vocal inputs and the application will query ChatGPT for the answers based on the transcription. The vocal transcription can be triggered by manual input, location-based proximity, timed triggers, and scenario based. Additionally, the transcriptions can not only be used for direct queries like previously stated but also for use in creating follow on events and simplifying the transcription for easier understanding. The personal assistant will also be able to use these transcriptions to create suggestions based on the vocal inputs such as creating contacts, putting events in calendar, and having an appointment for future recordings. The application will also have a web-based plug-in to query from previous transcriptions using the internet. These transcriptions will be stored within the application until deleted by the user. This application will not be able to come to conclusions using any outside knowledge, not be able to transcribe any vocal inputs unless one of the previous triggers has been set up in the app, and not be able to answer any questions about the transcriptions unless it is queried either afterwards or through a command in the app. In addition, the transcription of the vocal inputs will not be deleted from the app until specifically instructed to by the user. Also, the application shall be utilizing American English as the language of vocal inputs and text.

3. Risk Analysis

Taking stock of project risk is critical to project success. Analysis of project risks provides an opportunity to anticipate potential risks and mitigate negative impacts, as well as to capitalize on positive impacts. Having a understanding of risk probability allows the team to focus on the most likely risks and avoid wasted effort.

3.1 Risk matrix

Risk matrix allows identifying and evaluating the threats and risks of a specific situation in a visual way by looking at their possibilities and consequences. Usually, a risk matrix, 5x5 risk matrix or risk assessment matrix, is made up of 5 cells along the X axis and 5 cells along the Y axis. It does not have to be 5x5, yet this is the most common type. The 5x5 grids of risk matrices show the likelihood of risks occurring along the Y axis and the severity of their consequences along the X axis. Each axis follows a scale of very low to very high. The risks that could be faced are placed within the risk matrix depending on where they fall on this scale which assists in defining levels of risk (Glossop, 2023).

Likelihood x Consequence = Level of Risk

Likelihood

Table 4. Risk matrix diagram (Tucci, 2023)

		Consequences				
		Insignificant	Minor	Moderate	Significant	Severe
	Almost Certain	Medium # 5	High # 10	Very High # 15	Very High # 20	Very High # 25
	Likely	Medium # 4	Medium # 8	High # 12	Very High # 16	Very High # 20
	Possible	Low # 3	Medium # 6	Medium # 9	High # 12	Very High # 15

	Unlikely	Low # 2	Low # 4	Medium # 6	Medium # 8	High # 10
	Rare	Low # 1	Low # 2	Low # 3	Medium # 4	Medium # 5

3.2 Risk register

Table 5. Risk factors and mitigation strategies

Risk ID	Risk Description	Risk Chance	Mitigation Plan	Risk Owner
R1	Information privacy concerns (e.g., HIPAA).	Low # 2	This project is a proof of concept and therefore will not contain any personally identifiable information (PII).	Developers
R2	ChatGPT's analysis of the transcriptions	Medium # 6	Although ChatGPT has been shown to be effective at answering user inquiries, testing will need to be done to ensure that the answers given are satisfactory and what method of questioning is needed.	Developer External Risk. Note: Service Level Agreement
R3	Access to ChatGPT API in production scenario	Low # 3	This project will ensure that level access to the ChatGPT API is sufficient for expected users. The free tier of access to the API should be sufficient for the expected number of requests.	Developers

R4	Avoiding API rate limit requests.	Low # 2	If at any time it is noted that too many the application is approaching the rate limit for the API, the project will immediately plan for costs using the PRO tier for ChatGPT to increase the rate limit allowance.	Developer Lead BA
R5	Speech diarization service availability, reliability, and scalability	High # 12	Access to a free service for speech diarization may not be available or meet capacity needs. Diarization processing takes time, and may hinder the user experience if recordings cannot be queried directly after generation. While STT service may work well, diarization adds context by identifying speakers and adds value to recordings, especially if information can be linked together by a speaker across multiple recordings.	Developer Lead BA Team Lead
R6	A person leaves the team	Very High # 20	Other team members will need to take over the responsibilities that person had held and any incomplete work that they did. Additionally, any future responsibilities would need to be reassigned.	PM, Team Lead
R7	Mistaking or deficiency in requirements	Low # 2	Designate multiple checkpoints throughout the project life to check out the work with stakeholders.	PM
R8	Lacking or improper tools to deliver the product	Low # 3	Identify the required tools and equipment at the project inception and plan appropriately with stakeholders.	PM, Team lead

R9	Complexity of the system	Medium # 5	Identify system intricacy at the project early stages and discuss it with the team. Prioritize each complexity.	Developers
-----------	--------------------------	---------------	------------------------------------------------------------------------------------------------------------------------	------------

Legend

Low	Watch item, may not necessarily result in a schedule delay
Medium	May cause minor delay to schedule (1-2 days)
High	May cause moderate delay to schedule (3-5 days)
Very High	Schedule and/or progress are at risk (5+ days)

Risks that fall into the green areas of the map require no action nor monitoring. Yellow and orange risks require action. However, risks that fall into red portions of the map need urgent action (Tucci, 2023).

4. Communications

Effective communication is essential to ensure that all stakeholders are in alignment on the purpose, direction, and cadence of a project.

4.1 Objective

The objective of this communication plan is to ensure efficient and effective information flow amongst all stakeholders involved in the software development project. This will enhance coordination, foster mutual understanding, aid in timely decision-making, and streamline the resolution of issues. The plan is designed to maintain transparency, drive engagement, and support the successful execution of the project within the stipulated timeline and quality standards.

4.2 Communication Methods

The following methods will be utilized for project communication:

- **Emails:** Regular updates and documentation
- **Meetings:** Weekly team meetings, monthly stakeholder meetings
- **Project Management Software:** Real-time project status updates and document sharing

4.3 Stakeholder Analysis

Table 6. Project stakeholders

Stakeholder	Information Needed	Frequency	Method
Project Manager	Progress reports	Daily	Email, Software
Team Members	Task Assignments	As needed	Email, Meetings
Stakeholders	Status Reports	Monthly	Meetings
Executive Team	High-level summaries	Quarterly	Meetings

4.4 Communication Schedule

- **Daily Updates:** Sent by Project Manager via email and Project Management Software.
- **Weekly Team Meetings:** Held every Monday at 10am to discuss the tasks for the week.
- **Monthly Stakeholder Meetings:** Held on the first Wednesday of each month to discuss progress and next steps.
- **Quarterly Executive Meetings:** Summaries sent for the executive team to keep them updated.

4.5 Crisis Communication Plan

In the event of a project crisis, an emergency meeting will be called by the project manager. All relevant stakeholders will be informed through an email and a meeting will be conducted to discuss the issue and possible solutions.

4.6 Feedback Mechanism

Feedback will be sought after each major milestone. This will be conducted through a combination of one-on-one discussions and anonymous feedback surveys.

5. Change Management

Change management is the process of ensuring project management is aware of and approves all changes to be made to the system. Changes may have an impact on project scope and schedule and therefore should be dealt with deliberately and with sufficient tracking. Change management reduces risk and decreases confusion, while also ensuring that changes to the requirements do not put the project at unnecessary risk. Change control is to be enforced from the start of the project and is ultimately the goal of project management (PMI, 2017). Changes may be requested by any stakeholder in the project and may be initiated at any time.

5.1 Change Control Board (CCB)

The CCB is responsible for gatekeeping changes proposed to the STeMS backend services system.

Table 7. Change Control Board Membership

CCB Member	Role	Responsibilities
Mir Assadullah	Customer	<ul style="list-style-type: none">• Final approval or rejection of change• Determines change priority
David Babers	Project Manager	<ul style="list-style-type: none">• Determines if internal change requests are viable for review by the CCB
Jonathan Nagy	Team B Leader	<ul style="list-style-type: none">• Determines if internal changes requests from Team B are viable for review by CCB
Team Members	BA, Developer, QA, Writer	<ul style="list-style-type: none">• Provide analysis of change requests to produce an estimated level of effort and potential impacts to other requirements

Changes may be initiated verbally, but they must be written before submission to the Team B Leader, Project Manager, or the Change Control Board for consideration.

5.2 Internal Change Requests

Internal change requests are those that are generated by Team B members. Requests for change are first submitted to the Team B Leader and Project Manager in written form. The change request is either rejected for submission or forwarded to the CCB after the change request form has been created. Upon approval, an internal change request can be treated as an external change request and processed as such.

5.3 External Change Requests

External change requests are initiated by the customer and are sent to the PM and Team Leader to review. A subject matter expert (SME) on the team is selected from among the team members and they are responsible for filling out the change request form. Once the form is created, is presented to all the CCB members, the change is discussed, and the Customer has final approval or rejection authority. Upon approval, a priority is assigned by the Customer to help determine whether to introduce the change immediately or as a backlog item for future implementation.

Table 8. Change Priority

Priority	Description
High	The change is needed as soon as possible because of potential project impact.
Medium	The change will solve irritating problems or repair missing functionality. This change can be moved to the backlog.
Low	The change will lead to improvements, changes in workflow, or configuration. This change can be moved to the backlog.

5.4 Change Request Form

A template for the form can be found in Teams under Team B's Files in the Change Control Board folder or at this [link](#).

Figure 2. Change Request Form

Project Name	STeMS Backend Services
Team	Team B
Requested by	
Change Description	
Change Reason	
Impact of Change	
Proposed Action	
Status (Submitted, Approved, or Rejected):	
Approval Date:	
Approved By:	

5.5 Requirements Traceability Matrix (RTM)

To ensure coverage of all requirements, a requirements traceability matrix (RTM) will be maintained. This will track each requirement forward to the test case it is tested in, and reverse traceability from test case to all tested requirements. The RTM will be added to the test plan.

6. Organization of the Project

Project organization is critical to the success of the project. This section will develop the structure of the project, the project software development life cycle (SDLC) model, and the timeline of project events.

6.1 Project Model

This project will follow the Waterfall Methodology with aspects of the Spiral Method for gate checks and releases. The Waterfall Method is defined as, “also known as the Waterfall model — is a sequential development process that flows like a waterfall through all phases of a project (analysis, design, development, and testing, for example), with each phase completely wrapping up before the next phase begins” (Adobe, 2022). Within the Waterfall method there are five stages: requirements gathering, design, implementation, verification and testing, and maintenance. Each phase is sequential, although for this project the Waterfall method will be adapted to include more iterative gate checks during the development process.

Milestones’ chronology:

The chronology of Milestones throughout STeMS life cycle is defined as follows:

Milestone One - Project Initiation (06/03/2023)

- Project Plan
- Software Requirements Specification (SRS)
- Presentation to stakeholders

Milestone Two (06/17/2023)

- Updated project initiation
- Technical Design Document
- Add Software Test Plan to the Project Plan
- Presentation to stakeholders

Milestone Three (07/22/2023)

- Updated documents
- Programmer Guide
- Deployment and Operations Guide (Runbook)
- Add Software Test Plan to the Project Plan
- Presentation to stakeholders.

Milestone Four (08/05/2023)

- Finished software
- User Guide
- Test Report
- Programmer Guide.
- Presentation to stakeholders.

Why Waterfall is a better fit for STeMS?

Which project management methodology is right for any project depends on the industry, goals, team size, and budget. STeMS is intended to serve those impacted by STML. Its goals and outcomes are established from the beginning. Moreover, it requires completing deliverables to progress to the next phase. By studying the milestones' chronology, it is evident that they follow the linear approach which allows to plan and set a structure for the project early on. That makes it easier to execute and manage progress in the project. Usually, the Waterfall paradigm divides a project into 5 specific phases; requirement phase, design phase, implementation phase, testing phase, and maintenance phase. Generally, the budget for projects using the Waterfall paradigm is fixed. Because the project is defined from start to finish.

Regardless of its overwhelming amount of upfront documentation. Hence, it involves larger teams to take control of the project, 19 persons are split into two groups (one for frontend and another for backend). These individuals involved abide by a strict hierarchy, with the lead role going to the project manager. It is clear to notice that the Waterfall methodology closely fits STeMS project. Every stage is critical to project success, yet requirements and design are vital and require more attention.

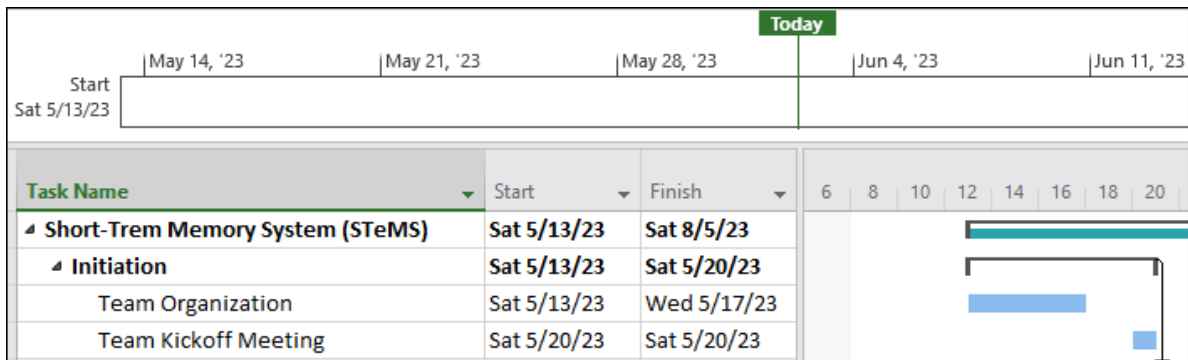
6.1.1 Initiation Phase

Planned starting date: 05/13/2023

Planned ending date: 05/20/2023

The initiation phase has consisted of several activities to prepare for work on this project. The team was assigned and convened by the customer and project manager(s). Team members have had three planning meetings to begin assignments, coordinate the creation of this document and the SRS, and create an initial presentation to show to the client and project stakeholders. Team members discussed and were assigned access to required resources, tools, and applications. Past experiences, skills, and engineering aptitudes were documented and provisional assignments to project task areas were determined. The end of this phase will be concluded by the version 1.0 of this document being released and the initial client presentation being completed.

Figure 3. Initiation phase timeline



Budget:

The budget will be calculated using the hourly rates for each employee with the assumption of a 20-hour work week with 20% extra for any unforeseen circumstances. In the case that more time is worked on the week then overtime will be worth time and a half. Therefore, the budget for the first phase would be \$10,488 given the salary of the team members of \$437/hr for the 20-hour work week and multiplying that by 1.2 for the 20% overhead

6.1.2 Definition Phase

Planned starting date: 05/20/2023

Planned ending date: 06/03/2023

Description of the result of the definition phase:

A list of requirements concerning the project result will be compiled in the definition phase.

Most important milestones:

Table 9. Milestone 1 and associated responsibilities

Milestone	Description	Responsibility	Gate/Quality Check
Project Plan	The version 1.0 of the Project Plan document.	Team B, Project Plan Team.	Submission of Project Plan Version 1.0 for Milestone 1.
SRS	The version 1.0 of the SRS document.	Team B, SRS document team.	Submission of SRS Version 1.0 for Milestone 1.
Client Presentation	Present current status of project.	Team B, PM.	Submission and presentation to client.

- List of functional requirements
- Research on legal requirements
- Requirements from interviews with end users
- Requirements from end-user tests
- Report of technical requirements
- Client approval of list of requirements

Activities in the definition phase:

Table 10. Definition phase activities

Activity	Responsible Parties	Description	Approval Required By	Approval Date
Create the Project Plan	Team B	The Project Plan is the document that serves as the organizing record of the project phases, outcomes, and deliverables.	PM, Client.	6/2/2023
Create the SRS	Team B	The document that describes the intended functionality, performance, and use of the project software.	PM, Client.	6/2/2023
Client Presentation	Team B	The client presentation is a combination of power point slides and audio to provide an overview of project activities during this phase.	PM.	6/3/2023

Budget:

This project will use open-source tools, frameworks, and technologies. As such, there is no immediate cost associated with the development and deployment of the mobile application. The team has decided to use the free tier of the ChatGPT API. Under this assumption and using the salary of the team, the budget for this phase would be \$20,976 Using the team's salary of \$437/hr multiplied by 40 hours after having 20 hours of work each week and then the whole result would be multiplied by 1.2 for a 20% overhead in case of any unforeseen circumstances.

Internal information:

Information for the Definition Phase will be recorded in three primary documents. This document, the SRS, and client presentation will be the main documents produced in this phase.

They will be made available to the client, project managers, and all team members. Each document will be stored in the Microsoft Teams Team B file directory. Access to the Teams space is managed by the client and project team.

External information:

Approval for this phase is accomplished by the approval from the client for this document and the successful conclusion of the client presentation.

6.1.3 Design Phase

Planned starting date: 06/04/2023

Planned ending date: 06/17/2023

Description of the result of the design phase:

The result of the design phase will be a full mockup of the functionality of the application so that the customer and potential users will see how the application will act according to various stimuli.

Most important milestones:

Table 11. Milestone 2 and associated responsibilities

Milestone	Description	Responsibility	Gate/Quality Check
Project Plan	Updated Project Plan document version 2.0 with tracked changes.	Team B Project Plan Team.	Submission of Project Plan Version 2.0 for Milestone 2
SRS	The version 2.0 of the Software Requirements Specification document with tracked changes.	Team B SRS document team.	Submission of SRS Version 2.0 for Milestone 2.
TDD	Present version 1.0 of Technical Design Document	Team B	Initial Submission of Version 1.0 for Milestone 2
Client Presentation	Present status of project.	PM Team B	Submission and presentation to client.

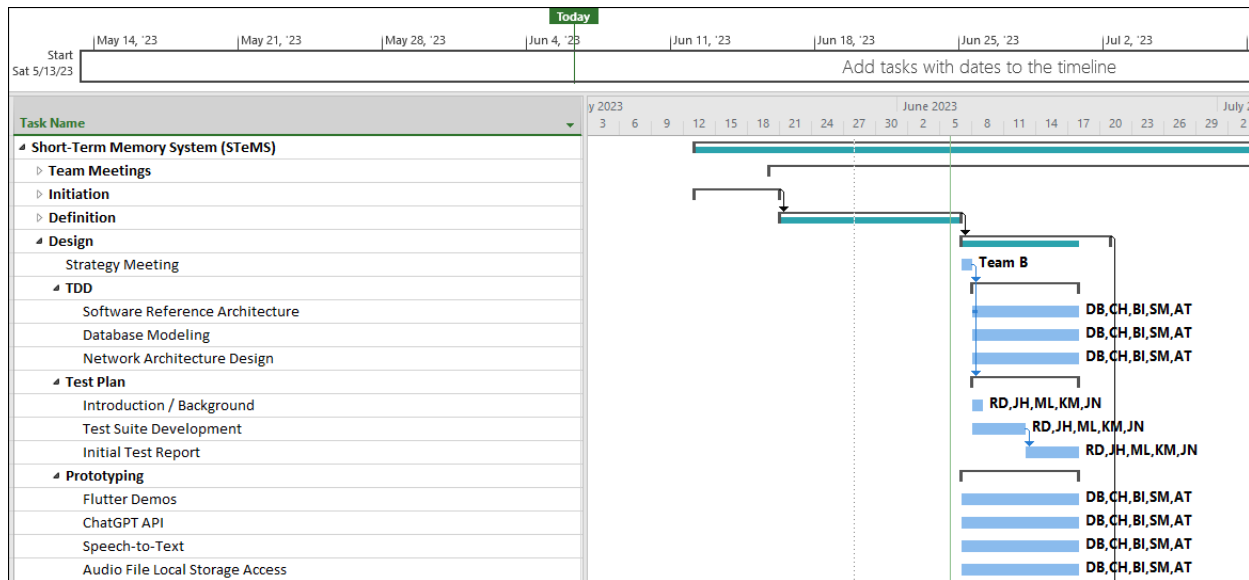
Activities in the design phase:

Table 12. Design phase activities

Activity	Responsible Parties	Description	Approval Required By	Approval Date
Update the Project Plan	Team B	Updating the project plan with new information and edits.	PM, Client	6/17/2023
Update the SRS	Team B	Updating the SRS with new information about the backend functionality, new information, and edits.	PM, Client	6/17/2023
Create TDD	Team B	Creating initial version of the Technical Design Document.	PM, Client	6/17/2023
Create Test Plan	Team B	Creating the initial version of the Test Plan document,	PM, Test Team	6/17/2023
Client Presentation 2	Team B	The client presentation is a combination of power point slides and audio to provide an overview of project activities during this phase.	PM	6/17/2023

Timeline:

Figure 4. Design phase timeline



Cost estimates:

The costs associated with this phase would be the pay for the team member as the previous phase established that the team would only be using free software. Using the team rate of \$437/hr and accounting for two 20 hour weeks on top of which would be a 20% overhead for emergencies meaning that the budget would be \$20,976.

Internal information:

The information of this phase will be used to format the actual application in the development phase and will be given to developers and testers to see how the application is intended to be used. This will be useful in making sure that application behaves exactly as the requirements need it to and to ensure that it will not output errors from correct inputs.

External information:

This phase is successfully completed when the updated documents have been approved, the new TDD has been accepted, and the presentation has been shown to the client.

6.1.4 Development Phase

Planned starting date: 06/18/2023

Planned ending date: 07/22/2023

Description of the result of the Development Phase:

The team will have finished the initial coding of the application and would have a deployment and operations run book and Programmer's Guide available for perusal.

Most Important Milestones

Table 13. Milestone 3 and associated responsibilities

Milestone	Description	Responsibility	Gate/Quality Check
Software	The backend code for the application	Team B Developers	Working application that is available to use on GitHub
Deployment and Operations Run Book	Create deployment and operations run book which shows how the application should be run.	Team B	Submit Run Book to Client/PM
PG	Create PG of backend code	Team B	Submit PG to Client/PM
Client Presentation	Presentation reporting the current status of the application to the client	Team B	Submit Report to Client/PM

Activities in the implementation phase:

Table 14. Activities in the Deployment phase

Activity	Responsible Parties	Description	Approval Required By	Approval Date
Start coding application	Team B	Code the initial application according the previous documents	PM, Client	07/22/23
Create Deployment and Operations Run Book	Team B	Creating initial version of the Deployment and Operations Run Book	PM, Client	07/22/23
Create PG	Team B	Create initial version of the Programmer's Guide	PM, Client	07/22/23

Client Presentation 3	Team B	The client presentation is a combination of power point slides and audio to provide an overview of project activities during this phase.	PM, Client	07/22/23
------------------------------	--------	------------------------------------------------------------------------------------------------------------------------------------------	------------	----------

Cost estimates:

The costs associated with this phase would be the hourly rate of the team of \$437/hr multiplied by five 20 hour work weeks plus the 20% overhead for emergencies, meaning that the budget would be \$52,440. This is under the understanding that only free software is being used.

Internal information:

The information from this phase would be recorded within the Deployment and Operations Run Book, the Programmer's Guide, and the results from using the initially coded application

External information:

The phase would've been successfully completed once the peer reviews are done, the deployment and operations run book is submitted, and the presentation that is shown to the client and has achieved the client's approval.

6.1.5 Implementation Phase

Planned starting date:07/23/2023

Planned ending date: 08/05/2023

Description of the result of the implementation phase:

The finished project will be delivered in the implementation phase. The client will receive the finished software with the test report as well as a programmer's guide, and user's guide. The client will also receive updated versions of any previously submitted documents and a presentation displaying all the results from this project.

Most important milestones:

Table 15. Milestone 4 and associated responsibilities

Milestone	Description	Responsibility	Gate/Quality Check
Completed Software	Completed Application of the Project.	Team B Developers	Working application that is available to use on GitHub
User Guide	Guide on how users should be able to use the application.	Team B SRS document team.	Submit Guide to Client/PM
Test Report	Report from the QA team regarding the tests conducted on the application.	Team B	Submit Report to Client/PM
Updated Previous Documents	Previous documents that have been submitted have been updated.	Team B	Submit latest revision of documents to Client/PM
Client Presentation	Final status of project.	Team B	The last presentation for the project that was shown to the client

Activities in the implementation phase:

Table 16. Activities in the implementation phase

Activity	Responsible Parties	Description	Approval Required By	Approval Date
Update Older Documents	Team B	Updating the older documents with tracked changes.	PM, Client.	08/05/23
Create User's Guide	Team B	Create User's Guide of the application	PM, Client.	08/05/23
Conduct Testing	Team B, QA	QA tests the application to ensure it meets all requirements.	PM, Client.	07/29/23

Create Test Report	Team B, QA	Create Test report based on Test Results.	PM, Client.	08/05/23
Submit Completed Code	Team B, Developers	Complete the coding of the application and ensure it is working.	PM, Client.	08/05/23
Client Presentation Final	Team B	The client presentation is a combination of power point slides and audio to provide an overview of project activities during this phase.	PM.	08/05/23

Cost estimates:

The costs associated with this phase would be the work done by the team so it would have the cost be the rate of \$437/hr multiplied by two 20 hour weeks which is also multiplied by a 20% overhead leading to a total of \$20,976.

Internal information:

The information from this phase would be recorded within the Deployment and Operations Run Book, User's Guide, Programmer's Guide, and Test Report. The only resource needed to complete these documents would be the working application. The entire team and client will have access to these documents.

External information:

The phase would've been successfully completed once the peer reviews are done, the documents and code are submitted, and the presentation that is shown to the client and has achieved the client's approval.

6.2 Project Summary

Throughout the 5 phases of development, with phase closing activities will be completed for each phase to ensure the successful completion of all deliverables required and appropriate progress toward the end goal of project completion.

Total Cost Estimates:

The project has a constant run rate of \$437 per hour for all staff, with all staff members projected to be fully utilized for 20 hours a week and with 20% overhead added in to cover equipment, supplies, and overtime. The total costs are projected to be \$125,856.

Table 17. Project Budget

Phase	Team Rate/Hour	Weeks	Hours/Week	Base Cost	Cost Plus Overhead
Initiation	\$437	1	20	\$8,740	\$10,488
Definition	\$437	2	20	\$17,480	\$20,976
Design	\$437	2	20	\$17,480	\$20,976
Development	\$437	5	20	\$43,700	\$52,440
Implementation	\$437	2	20	\$17,480	\$20,976
Total		12		\$104,880	\$125,856

7. Software Test Plan

This section contains the software test plan for the STeMS backend services project, including what will be tested, how it will be tested, and a list of test cases paired with requirements under test.

7.1 Purpose

The software test plan, a living document, describes the goals and objectives for one the most important phase of product development. It is considered as a blueprint of how the testing activities will be conducted in the project. The project test plan is designed to assign the testing approach used for the STeMS backend services. The main objective is to pinpoint potential issues that prevent the application from satisfying the defined project requirements. The development team will make every effort to mitigate any issue that is found to avoid any user distraction and ensure that the application is functioning faultlessly as it should.

7.2 Scope

The test plan scope incorporates three testing approaches; Unit testing, Integration or Regression Testing, and User Acceptance Testing (UAT) – acceptance criteria that will be used to evaluate the system.

Unit testing will be used by the team throughout the development process to test the functionality of separate components. The bulk of the testing to be completed will be comprised of unit testing to verify that components are communicating properly, from the ChatGPT API, the local device, to the web browser. In dealing with various services, discrete testing must be conducted on the connectivity, latency, and security of each line of communication to ensure the application is fully functional.

Since the final product is being developed by two teams, a front-end team and a backend team, the STeMS will be composed of two applications. Therefore, STeMS application should undergo integration testing to ensure that both development efforts are operating in harmony.

User Acceptance Testing will be utilized to evaluate whether the application meets the requirements of the client and end-users – impacted by STML. If bugs are found in the system during the testing process, the issue must be prioritized by severity and added as a defect in the project repository.

7.3 Considered Requirements

This section describes the items to be tested and not to be tested, making clear what concerns are in scope of testing and those that are not.

7.3.1 To Be Tested

The software test will cover all features and functionalities comprised in the SRS document, including the API, to verify and validate that the STeMS mobile backend services work as intended and accomplishes the tasks it was designed for. Those features and functionalities consist of:

1. STeMS backend services shall maintain audio recordings made by the user and saved to local storage.
2. STeMS backend services shall maintain audio recordings metadata added by the backend system and in part maintained by the user and saved to local storage.
3. STeMS backend services shall generate audio transcripts with speaker identification.
4. STeMS backend services shall generate a food order from a recorded audio interaction between a waiter and one or more customers.
5. STeMS backend service shall generate reminders based on an audio recording.
6. STeMS backend service shall display operating system managed reminder notifications at the time a reminder is pertinent.
7. STeMS backend service shall provide an API to allow a browser extension to fill in a web form using an audio recording as context.
8. STeMS backend service shall make metadata and transcripts from multiple recordings searchable through a provided API.

7.3.2 Not to be tested

Any feature or requirement that is not included in the SRS document will be out of scope and shall not be tested for STeMS application. It includes:

1. STeMS ConvoBuddy application user interface functionality.
2. STeMS browser extension user interface functionality.
3. STeMS backend services shall integrate with the adopted STT API.
4. STeMS backend services shall run on the Android operating system for mobile devices.
5. STeMS backend services shall be built using Dart programming languages and Flutter, an open-source UI toolkit.

7.4 Approach

All software flaws will be reported on the GitHub board. They will be logged with their type, actual results, expected results, level of criticality, and traced to a specific requirement.

Unit/Widget Testing – Unit tests will provide the foundation for STeMS testing, verify the validity of classes and algorithms employed in the STeMS backend services. Widget testing is at a minimum as widgets are primarily a UI concern and the front-end team is responsible for developing that part of the application.

Integration/Regression Testing – As individual components of the application are built, a robust testing cycle to help integrate the application with local device and minimize re-work and defects must be ensured. With each new software build, the team will make sure that new code will not influence current functionalities. Both integration and regression testing can be achieved with assistance from test automation suite via Flutter integration tests. Flutter integration tests provide a similar environment to developing unit tests, reducing the learning curve, and increasing speed and proficiency of developers writing integration tests. Automated testing will considerably reduce effort later by reducing the criticality of manual regression testing each component. A test that would normally take minutes to execute, will be completed in seconds.

User Acceptance Testing / Focus Group – All tests identified as UAT will verify acceptance and exit criteria of requirements. The user acceptance tests are meant to fully verify the objective of the requirement. Once UAT is complete for all functional requirements (must-haves), signing off on completion of the application will be close. Real individuals who the STeMS application is aimed at will be testing on the platform of their choosing operating system (iOS/Android) and feedback for “future” versions will be received.

7.5 Pass/Fail Criteria

- Tests considered successful once no Major or Critical bugs are observed over repeated consecutive testing sessions.
- Audio and transcription quality will be considered when performing tests. Low quality audio will be rerecorded to improve before failing a test.
- Any test where a Major or Critical bug exists or is uncovered is considered a failure.

7.6 Defect Severity Levels

There are four levels of severity that will be used to track any found defects:

1. Low – Meant for non-functional, non-essential issues that do not take away from functional or performance requirements (minor text/display bugs).
2. Moderate – Includes both functional and non-functional bugs that can be worked around.
3. Major – Functional requirement bugs that may delay further development and need to be addressed immediately.
4. Critical – An issue that prevents any further development of one or more feature required for project success.

7.7 Risks and Contingencies

This section outlines potential and known risk to successful execution of the test plan and contingencies that will be used to mitigate the negative impact of those risks.

Risk	Contingencies
The ConvoBuddy application and browser extension are written by team A and may be in an incomplete state while testing of backend services begins.	Testing of backend services should not be dependent on front-end interfaces but instead utilize a testing utility written directly against STeMS backend service interfaces and methods.
Code may be in an unwritten or incomplete state while developing test utility.	The test utility should be written against mock objects using the same interfaces as the intended final product.
Validating the result of third-party services matches with test utility output requires seeing the raw return values from the service.	All third-party service calls should be written to a log file or database and should be reviewable by timestamp or unique service call id available from the test case utility.

7.8 Test Cases

This section contains all the test cases for the STeMS backend services, subdivided by functional area. It consists of twelve categories; Recording API, Browser Extension API, ChatGPT Query API: Food Orders, ChatGPT Query API: STML Reminders, ChatGPT Query API: Global Search, User Profile, Reminder Notifications, Non-Functional Requirements, and Future Requirements.

7.8.1 Recordings API

Test Case 1A: Metadata Update

Description	The ConvoBuddy application can update recording metadata.
Test Type	Functional
Requirement	A recording's metadata like title and author can be modified.
Req. ID	1
Prerequisite	A recording must be successfully created to edit its metadata. Access to local storage must be enabled.
Steps	1. The ConvoBuddy application utilizes the Flutter package for access to audio file metadata and calls the package's set method to define new metadata values.
Expected Output	The audio file's metadata is overwritten with the defined values in the set method call.
Assumption	The path to the audio file used in the method is accurate. Necessary packages are imported.

Test Case 2A: Metadata Retrieval

Description	The ConvoBuddy application can get recording metadata.
Test Type	Functional
Requirement	A recording's metadata is retrieved from the audio file.
Req. ID	2
Prerequisite	A recording must exist for metadata to be retrieved. Access to local storage must be enabled.

Steps	1. The ConvoBuddy application utilizes the Flutter package for access to audio file metadata and calls the package's get method to retrieve the audio file's metadata.
Expected Output	The audio file's metadata is retrieved and returned to the ConvoBuddy application.
Assumption	The path to the audio used in the method is accurate. Necessary packages are imported.

Test Case 5A: Get List of Recordings

Description	The ConvoBuddy application can get a list of all recording titles.
Test Type	Functional
Requirement	A list of recording names is returned to the ConvoBuddy application.
Req. ID	5
Prerequisite	One or more recordings must exist in local storage. Access to local storage must be enabled.
Steps	1. The ConvoBuddy application utilizes the Flutter package for access to recording metadata and calls the package's get method to retrieve the recording's metadata for each recording in local storage.
Expected Output	A list of all recording file names is returned from local storage.
Assumption	Necessary packages are imported.

Test Case 6A: Delete Recording

Description	The ConvoBuddy application can request the deletion of a recording and it will be removed.
Test Type	Functional
Requirement	The requested recording file is deleted from the local storage.
Req. ID	6
Prerequisite	One or more recordings must exist in local storage. Access to local storage must be enabled.

Steps	1. The ConvoBuddy application utilizes the Flutter package for access to the recording file and calls the package's delete method to delete the recording file with the file path in local storage.
Expected Output	The recording file and recording metadata is deleted from local storage.
Assumption	Necessary packages are imported.

Test Case 7A: Save Recording

Description	The ConvoBuddy application can store a recording when it is stopped.
Test Type	Functional
Requirement	A recording file is created into local storage.
Req. ID	7
Prerequisite	Microphone access must be enabled. Local storage must have remaining capacity to store the recording file. Access to local storage must be enabled.
Steps	1. The ConvoBuddy application utilizes the Flutter package to check if permission is granted for recording functionality. 2. Then, the start record method is called with the desired destination path as a parameter for the recording to be saved. 3. The ConvoBuddy application calls the stop method to stop the recording after a defined period of time.
Expected Output	A recording file is created at the defined path.
Assumption	Necessary packages are imported.

Test Case 9A: Request Transcription

Description	The ConvoBuddy application can send a request to convert a recording to text and receive text back.
Test Type	Functional
Requirement	Request is sent to an API to get a response back as a text transcription of the recording.
Req. ID	9

Prerequisite	OpenAI API key needs to be set up. A recording file must be available in local storage.
Steps	<ol style="list-style-type: none"> 1. The ConvoBuddy application initializes the package with the API key. 2. An OpenAI audio model is instantiated. 3. The model object calls the create transcription method by providing the following parameters: <ol style="list-style-type: none"> a. file path b. model type c. response format (JSON)
Expected Output	The audio transcription is returned as a JSON object. Transcript must include speaker identification (numerical or otherwise) for each continuous set of words from one speaker in the original audio recording.
Assumption	Necessary packages and dependencies are installed. API key is defined.

Test Case SR1: Recording File Location

Description	All recordings are to be stored in the device's local storage.
Test Type	Functional
Requirement	The file path of a recording file shows that it is located in local storage.
Req. ID	SR1
Prerequisite	None
Steps	<ol style="list-style-type: none"> 1. By code inspection verify that all methods for saving recordings is routed to local storage.
Expected Output	No instance of use of remote storage are found.
Assumption	None

Test Case SR2: Record in Background

Description	Recording will continue in the background even while the application is not open.
-------------	-----------------------------------------------------------------------------------

Test Type	Functional
Requirement	Record action persists as background service until stopped from ConvoBuddy application.
Req. ID	SR2
Prerequisite	Microphone access must be enabled. Access to local storage must be enabled. Background services must be enabled.
Steps	<ol style="list-style-type: none"> 1. The ConvoBuddy application calls the foreground service. 2. The record action is called from the on-start property of the foreground service. 3. The pause action is called from the on stopped property of the foreground service.
Expected Output	The record action runs in the background without the app actively open.
Assumption	Necessary packages are imported.

Test Case SR3A: Single Recording Session

Description	Only one recording can be recorded at a time.
Test Type	Functional
Requirement	The record action produces one recording file per record process.
Req. ID	SR3
Prerequisite	Microphone access must be enabled. Access to local storage must be enabled. Sufficient storage must be available.
Steps	<ol style="list-style-type: none"> 1. A recording request is received. 2. The ConvoBuddy application launches the record method. 3. A pause request is received. 4. A single recording is saved into local storage.
Expected Output	Single record session is executed upon request.

Assumption	Necessary packages are imported.
------------	----------------------------------

Test Case SR3B: Single Recording Session

Description	An error will be generated when more than one recording is attempted at a time.
Test Type	Functional
Requirement	The record action produces one recording file per record process.
Req. ID	SR3
Prerequisite	Microphone access must be enabled. Access to local storage must be enabled. Sufficient storage must be available.
Steps	<ol style="list-style-type: none"> 1. The ConvoBuddy application is launched the record method. 2. The system returns indicating that the recording has started 3. Some audio is produced for recording. 3. The ConvoBuddy application launches a second recording. 4. Additional audio is produced for recording. 5. The ConvoBuddy application is launched to stop the recording.
Expected Output	The second request to start recording results in an error stating that only one recording can be active at a time. The recording produced should contain all audio produced for recording from the initial recording start.
Assumption	Necessary packages are imported.

7.8.2 Browser Extension API

Test Case 11A: Create App Instance Code

Description	Verify an app instance code can be requested from and returned by the system.
Test Type	Functional
Requirement	The ConvoBuddy application can request and receive an app instance code.
Req. ID	11

Prerequisite	None
Steps	1. The ConvoBuddy application calls the system method for generating an app instance code.
Expected Output	The system returns a response containing a generated app instance code. The app instance code is stored in local storage on the device.
Assumption	None

Test Case 12B: Store App Instance Code

Description	Verify newly generated app instance codes are stored on local storage.
Test Type	Functional
Requirement	When an app instance code is generated for the ConvoBuddy app it is stored on local storage.
Req. ID	12
Prerequisite	None
Steps	1. The ConvoBuddy application calls the system method for getting the app instance code.
Expected Output	The generated app instance code is stored on local storage.
Assumption	None

Test Case 13A: Get App Instance Code

Description	Verify an existing app instance code can be requested from and returned by the system.
Test Type	Functional
Requirement	When the ConvoBuddy application requests the app instance code, the code stored on local storage is returned.
Req. ID	13
Prerequisite	An app instance code has been previously generated by the system.
Steps	1. The ConvoBuddy application calls the system method for getting the app instance code.
Expected Output	The system returns the previously generated app instance code.

Assumption	None
------------	------

Test Case 14A: App Instance Code Required in Extract Field Values Request

Description	They system shall verify that the app instance code is included in requests to extract form values.
Test Type	Functional
Requirement	When the browser extension requests to extract form values, the app instance code and form fields to be populated must be sent.
Req. ID	14
Prerequisite	An app instance code has been previously generated by the system.
Steps	1. The Browser Extension calls the system method to extract form values without an app instance code.
Expected Output	The system returns an error response stating that app instance code is missing.
Assumption	None

Test Case 14B: Form Fields Required in Extract Field Values Request

Description	The system shall verify that form fields to extract are included in requests to extract form values.
Test Type	Functional
Requirement	When the browser extension requests to extract form values, the app instance code and form fields to be populated must be sent.
Req. ID	14
Prerequisite	An app instance code has been previously generated by the system.
Steps	1. The Browser Extension calls the system method to extract form values with an app instance code and without form fields to populate.
Expected Output	The system returns an error response stating that no form fields were specified.
Assumption	None

Test Case 15A: Verify Transcript and Form Fields to be Filled

Description	Verify that currently selected recording transcript and list of form fields to be populated is used to generate a ChatGPT prompt.
Test Type	Functional
Requirement	When the Browser Extension API receives a request for form value extraction with an app instance code and form fields to be populated, the transcript of the currently selected recording in the interface is used to build a ChatGPT prompt.
Req. ID	15
Prerequisite	An app instance code has been previously generated by the system.
Steps	1. The Browser Extension calls the system method to extract form values with an app instance code and form fields to populate.
Expected Output	A ChatGPT prompt is generated and logged to the local system.
Assumption	A logging framework is utilized and all requests to ChatGPT are logged.

7.8.3 ChatGPT Query API: Food Orders

Test Case 17A: Validate Audio Transcript and Generate Prompt

Description	When the ConvoBuddy application sends an audio transcript and the user profile to the Food Orders API endpoint, the transcript is validated, the text is sent to OpenAI/ChatGPT via API
Test Type	Functional
Requirement	When the ConvoBuddy application submits a job to send audio transcript for processing on the Food Orders API endpoint, the system validates the transcript and if valid the system generates the food order prompt containing the transcript and user profile data and sends the prompt to ChatGPT.
Req. ID	17
Prerequisite	Transcription must be validated prior to creation of query to be sent to OpenAI.
Steps	1. The ConvoBuddy application is used to submit a Food Orders API request with a valid transcript. 2. The tester waits until the request completes.

Expected Output	The response from ChatGPT is logged. Formatted data to be consumed by the front-end is output by the ConvoBuddy application. Data should contain a list of food orders separated by speaker identification number or name as available from ChatGPT and audio transcript.
Assumption	Audio transcription is valid. A logging mechanism is in place to output the request to the ChatGPT API.

Test Case 20A: Validate Audio Transcript

Description	ConvoBuddy app determines the audio transcription is invalid and will generate an appropriate error message.
Test Type	Functional
Requirement	When the ConvoBuddy application submits a job to send audio transcript for processing on the Food Order endpoint, the system validates the transcript and if not valid the endpoint returns an error response.
Req. ID	20
Prerequisite	None.
Steps	1. The ConvoBuddy application initiates a food order request with an invalid transcript.
Expected Output	An error message stating that the transcript is invalid is created and logged. A human readable error is returned to the testing utility.
Assumption	None.

Test Case 21A: Validate OpenAI Service is Available

Description	ConvoBuddy app has query ready for OpenAI, but the OpenAI API service is unavailable, which causes an error message.
Test Type	Functional
Requirement	When the ConvoBuddy application submits a job to send audio transcript for processing on the Food Order endpoint, the system validates the transcript and it is valid, and OpenAI API service is unavailable, the endpoint returns an error response.
Req. ID	21
Prerequisite	Req. ID 17

Steps	<ol style="list-style-type: none"> 1. A service interruption in OpenAI API service is simulated by disabling the internet connection. 2. The testing utility is called to perform a Food Order request with a valid transcript.
Expected Output	An error message stating OpenAI API service is unavailable is created and logged. The human readable error is returned to the testing utility.
Assumption	Audio transcription has been validated by ConvoBuddy app.

7.8.4 ChatGPT Query API: STML Reminders

Test Case 24A: Validate Audio Transcript and Generate Prompt

Description	ConvoBuddy app validates audio transcription and generates a STML Reminders prompt with those details along with user profile information to OpenAI for processing.
Test Type	Functional
Requirement	When the ConvoBuddy application submits a job to send audio transcript for processing on the STML reminders endpoint, the system validates the transcript and if valid the system generates a reminders prompt containing the transcript and user profile data and sends the prompt to OpenAI.
Req. ID	24
Prerequisite	Transcription must be validated prior to creation of query to be sent to OpenAI.
Steps	<ol style="list-style-type: none"> 1. The ConvoBuddy app calls the STML reminders endpoint of the backend services with a valid transcript. 2. The tester waits until The ConvoBuddy app has completed the request.
Expected Output	Formatted data is generated and displayed on the front-end. Data should contain a list of reminders with times.
Assumption	Audio transcription is valid.

Test Case 27A: Validate Audio Transcript

Description	ConvoBuddy app determines the audio transcription is invalid and will generate an appropriate error message.
Test Type	Functional

Requirement	When the ConvoBuddy application submits a job to send audio transcript for processing on the STML reminders endpoint, the system validates the transcript and if not valid the endpoint returns an error response.
Req. ID	27
Prerequisite	None.
Steps	1. The testing utility is launched to send a STML request to the backend services with an invalid transcript.
Expected Output	An error message stating that the audio transcript is invalid is created and logged. A human readable error message is returned to ConvoBuddy for display.
Assumption	ConvoBuddy app has confirmed that the audio transcription is invalid.

Test Case 28A: Validate OpenAI Service is Available

Description	ConvoBuddy app has query ready for OpenAI, but the OpenAI API service is unavailable, which causes an error message.
Test Type	Functional
Requirement	When the ConvoBuddy application submits a job to send audio transcript for processing on the STML reminders endpoint, the system validates the transcript and it is valid, and OpenAI API service is unavailable, the endpoint returns an error response.
Req. ID	28
Prerequisite	Req. ID 24
Steps	<ol style="list-style-type: none"> 1. OpenAI Api service interruption is simulated by disconnecting the internet or updating the host file to redirect OpenAI servers to localhost. 2. The testing utility is launched to send a STML request to the backend services with a valid transcript.
Expected Output	An error message is created and logged stating the OpenAI service is unavailable. A human readable error message is returned to the testing utility for display.
Assumption	None.

7.8.5 User Profile

Test Case 38A: Get User Profile

Description	When the ConvoBuddy application sends a request for user profile, the system returns the text of the user profile.
Test Type	Functional
Requirement	Use profile data is returned back to the ConvoBuddy application.
Req. ID	38
Prerequisite	User profile data must exist.
Steps	<ol style="list-style-type: none">1. The ConvoBuddy application is used to submit a request to get the user profile object.2. Correct parameters corresponding to a user profile are used in the method.3. User profile attribute data is returned from the method.
Expected Output	User profile data is returned as an object for parsing.
Assumption	None

Test Case 39A: Get User Profile Error

Description	When the ConvoBuddy application sends a request for user profile and the user profile is unavailable, an error response is returned to the application.
Test Type	Functional
Requirement	An error response is returned back to the ConvoBuddy application.
Req. ID	39
Prerequisite	None
Steps	<ol style="list-style-type: none">1. The ConvoBuddy application is used to submit a request to get the user profile object.2. Invalid parameters are passed into the method.3. Request returns a response object with error details.
Expected Output	A response object with error details is returned upon invalid request.
Assumption	None

Test Case 40A: Set User Profile Data

Description	When the ConvoBuddy application sends a request to set user profile, the system validates the input and if valid overwrites the text of user profile.
Test Type	Functional
Requirement	User profile data is overwritten with new data.
Req. ID	40
Prerequisite	User profile data must exist.
Steps	<ol style="list-style-type: none">1. The ConvoBuddy application is used to call the method to set a user profile object.2. System validates the method parameter for correct parameter types on the set method.3. The set method is invoked, and the specified user profile's data is overwritten with request data.
Expected Output	A successful response is returned to validate successful execution.
Assumption	None

Test Case 41A: Set User Profile Data Input Error

Description	When the ConvoBuddy application sends a request to set user profile, the system validates the input and if validation fails, an error response is returned.
Test Type	Functional
Requirement	Invalid request for setting user profile data is provided.
Req. ID	41
Prerequisite	User profile data must exist.
Steps	<ol style="list-style-type: none">1. The ConvoBuddy application is used to call the set method on a user profile object.2. Incorrect parameter data type is passed into the set method call.3. An error response is returned with error details in response body.
Expected Output	An error response is returned to indicate a failed request back to the ConvoBuddy application.
Assumption	None

7.8.6 Reminder Notifications

Test Case 43A: Display Reminder Notification

Description	Reminder date and time has passed without a reminder being shown causing a separate reminder to be sent to the device's operating system.
Test Type	Functional
Requirement	If the system detects that a reminder time has passed and the reminder has not been shown yet, the reminder is sent to the operating system as a notification.
Req. ID	43
Prerequisite	None.
Steps	<ol style="list-style-type: none">1. A STML reminder is generated by submitting a STML reminder request with a transcript containing a event scheduled for the next few minutes or seconds.2. The tester waits for the reminder time to ellapse and checks for a notification.
Expected Output	Notification of a passed reminder is sent to the device's operating system.
Assumption	Local device has a saved reminder with valid date and time.

Test Case 44A: Mark Reminder as Displayed

Description	Once ConvoBuddy app sends notification to the local operating system to be shown to the user, the metadata for that reminder instance will be updated to reflect that it has been shown.
Test Type	Functional
Requirement	If the system sends a notification to the operating system for display, the system will update the reminder metadata to reflect that it has been shown.
Req. ID	44
Prerequisite	Req. ID 43
Steps	<ol style="list-style-type: none">1. A STML reminder is generated by submitting a STML reminder request with a transcript containing a event scheduled for the next few minutes or seconds.2. The tester waits for the reminder time to ellapse and checks for a notification.

Expected Output	Reminder metadata reflects the status of the instance as successfully shown to the user.
Assumption	None.

7.8.7 Non-Functional Requirements

Test Case NF2A: Encrypt Communication Transmissions

Description	Data transmission security to include communication protocols, such as HTTPS.
Test Type	Nonfunctional
Requirement	The API must use secure communication protocols (e.g., HTTPS) to protect data transmission.
Req. ID	NF2
Prerequisite	None.
Steps	1. Verify by code and configuration inspection that all service calls are routed through HTTPS.
Expected Output	Communications performed through STeMS are encrypted.
Assumption	The STeMS internal network can be fully trusted.

Test Case NF3A: Use API Secrets

Description	Data transmission security level for internet communications.
Test Type	Nonfunctional
Requirement	Services exposed to the internet should implement API secret or user authentication to prevent external actor use or denial of service attacks.
Req. ID	NF3
Prerequisite	Internal communications protocols have been implemented to any outgoing transmission.
Steps	1. Verify by inspection of code and configuration that all external data transmission is routed through encrypted means.
Expected Output	Secure transmission of encrypted data across the internet.
Assumption	Req. ID NF2 has been successfully implemented.

Test Case NF4A: No API Keys in Source

Description	Separation of authentication details and the program's source code.
Test Type	Nonfunctional
Requirement	API keys and other secrets must be sanitized from source code.
Req. ID	NF4
Prerequisite	None.
Steps	<ol style="list-style-type: none">1. Verify by inspection of code that no API keys are hard coded.2. Verify by inspection that all API keys are loaded from a local secrets file or similar non-source code integrated mechanism.
Expected Output	Source code repository will not contain the authentication database for all user profiles.
Assumption	None.

Test Case NF5A: Service Availability

Description	ConvoBuddy app will perform at a monthly level of 99% uptime.
Test Type	Nonfunctional
Requirement	The API servers should have a minimum uptime of 99% per month.
Req. ID	NF5
Prerequisite	None.
Steps	<ol style="list-style-type: none">1. Verify the hosting services used comply with minimum uptime.
Expected Output	Minimum of 99% uptime of the application.
Assumption	None.

Test Case NF8A: Scalability

Description	Verify that API can handle increased user load.
Test Type	Nonfunctional

Requirement	The API should be designed to handle increased user load by scaling horizontally or vertically.
Req. ID	NF8
Prerequisite	Load balancers should be planned for the architecture of the system.
Steps	<ol style="list-style-type: none"> 1. ConvoBuddy app maintains a record of the load expected from the app. 2. If user load crosses threshold of 85% capacity, a notification will be sent to the developers to plan for scaling the application's capacity.
Expected Output	Load balancing across all user profiles within the ConvoBuddy app.
Assumption	None.

7.8.8 Future Requirements

Test Case F1A: Cloud Storage

Description	Third-party storage compatibility.
Test Type	Functional
Requirement	Recordings may be backed up or offloaded to the cloud if the user configures a third-party cloud storage service.
Req. ID	F1
Prerequisite	None.
Steps	<ol style="list-style-type: none"> 1. User selects audio recordings to be backed up to a third-party cloud storage system. 2. ConvoBuddy app communicates with the third-party storage service and sends the data to be securely stored. 3. ConvoBuddy app removes that data from the local device.
Expected Output	Audio recordings saved to a third-party cloud storage system and more available storage on the local device.
Assumption	User has a third-party cloud storage account with available storage.

8. Overview

Providing an overview of the costs and the timeline of the project.

Table 18. Costs and timeline overview

Description	Start Date	Completion Date	Cost Estimate
Initiation	05/13/2023	05/20/2023	7 Days
Definition	05/20/2023	06/03/2023	14 Days
Design	06/03/2023	06/17/2023	14 Days
Development	06/17/2023	07/22/2023	35 Days
Implementation	07/22/2023	08/05/2023	14 Days
Total			84 Days

Table 19. Work Breakdown Structure

WBS Level	Task Name	% Complete	Start	Finish	Resource
1	Short-Term Memory System (STeMS)	42%	Sat 5/13/23	Tue 8/8/23	Team B
1.1	Team Meetings	24%	Sat 5/20/23	Tue 8/8/23	Team B
1.1.1	Meeting 1	100%	Sat 5/20/23	Sat 5/20/23	Team B
1.1.2	Meeting 2	100%	Sun 5/21/23	Sun 5/21/23	Team B
1.1.3	Meeting 3	100%	Wed 5/24/23	Wed 5/24/23	Team B
1.1.4	Meeting 4	100%	Sun 5/28/23	Sun 5/28/23	Team B
1.1.5	Meeting 5	100%	Wed 5/31/23	Wed 5/31/23	Team B
1.1.6	Meeting 6	100%	Mon 6/5/23	Mon 6/5/23	Team B
1.1.7	Meeting 7	0%	Wed 6/7/23	Wed 6/7/23	Team B
1.1.8	Meeting 8	0%	Sun 6/11/23	Sun 6/11/23	Team B
1.1.9	Meeting 9	0%	Wed 6/14/23	Wed 6/14/23	Team B
1.1.10	Meeting 10	0%	Sun 6/18/23	Sun 6/18/23	Team B
1.1.11	Meeting 11	0%	Wed 6/21/23	Wed 6/21/23	Team B
1.1.12	Meeting 12	0%	Sun 6/25/23	Sun 6/25/23	Team B
1.1.13	Meeting 13	0%	Wed 6/28/23	Wed 6/28/23	Team B
1.1.14	Meeting 14	0%	Sun 7/2/23	Sun 7/2/23	Team B
1.1.15	Meeting 15	0%	Wed 7/5/23	Wed 7/5/23	Team B
1.1.16	Meeting 16	0%	Sun 7/9/23	Sun 7/9/23	Team B
1.1.17	Meeting 17	0%	Wed 7/12/23	Wed 7/12/23	Team B
1.1.18	Meeting 18	0%	Sun 7/16/23	Sun 7/16/23	Team B
1.1.19	Meeting 19	0%	Wed 7/19/23	Wed 7/19/23	Team B
1.1.20	Meeting 20	0%	Sun 7/23/23	Sun 7/23/23	Team B
1.1.21	Meeting 21	0%	Wed 7/26/23	Wed 7/26/23	Team B
1.1.22	Meeting 22	0%	Sun 7/30/23	Sun 7/30/23	Team B
1.1.23	Meeting 23	0%	Wed 8/2/23	Wed 8/2/23	Team B
1.1.24	Meeting 24	0%	Sun 8/6/23	Sun 8/6/23	Team B
1.1.25	Meeting 25	0%	Tue 8/8/23	Tue 8/8/23	Team B
1.2	Initiation	100%	Sat 5/13/23	Sat 5/20/23	Team B
1.2.1	Team Organization	100%	Sat 5/13/23	Wed 5/17/23	Team B
1.2.2	Team Kickoff Meeting	100%	Sat 5/20/23	Sat 5/20/23	Team B

1.3	Definition	95%	Sun 5/21/23	Tue 6/6/23	Team B
1.3.1	Use Case Discussion	100%	Sun 5/21/23	Sun 5/21/23	Team B
1.3.2	Resource Assignment	100%	Sun 5/21/23	Sun 5/21/23	Team B
1.3.3	SRS	95%	Mon 5/22/23	Tue 6/6/23	SRS Group
1.3.3.1	Use Case Reports	95%	Mon 5/22/23	Tue 6/6/23	JN,SM,JH,BI
1.3.3.2	Use Case Diagram	95%	Mon 5/22/23	Tue 6/6/23	JN,SM,JH,BI
1.3.3.3	Screen Mock-Ups	95%	Mon 5/22/23	Tue 6/6/23	JN,SM,JH,BI
1.3.4	Requirements	95%	Mon 5/22/23	Tue 6/6/23	JN,SM,JH,BI
1.3.4.1	Functional Requirements	95%	Mon 5/22/23	Tue 6/6/23	JN,SM,JH,BI
1.3.4.2	Non-Functional Requirements	95%	Mon 5/22/23	Tue 6/6/23	JN,SM,JH,BI
1.3.5	Project Plan	95%	Mon 5/22/23	Tue 6/6/23	Project Plan Group
1.3.5.1	Scope Definition	95%	Mon 5/22/23	Tue 6/6/23	DB,RD,CH,ML, KM,AT
1.3.5.2	Risk Analysis	95%	Mon 5/22/23	Tue 6/6/23	DB,RD,CH,ML, KM,AT
1.3.5.3	Communications Plan	95%	Mon 5/22/23	Tue 6/6/23	DB,RD,CH,ML, KM,AT
1.3.5.4	Change Management Plan	95%	Mon 5/22/23	Tue 6/6/23	DB,RD,CH,ML, KM,AT
1.3.5.5	Change Control Board (CCB)	95%	Mon 5/22/23	Tue 6/6/23	JN
1.3.5.5.1	Change Request Process	95%	Mon 5/22/23	Tue 6/6/23	JN
1.3.5.5.2	Requirements Traceability Matrix (RTM)	95%	Mon 5/22/23	Tue 6/6/23	JN
1.3.5.6	Project Management Methodology	95%	Mon 5/22/23	Tue 6/6/23	DB,RD,CH,ML, KM,AT
1.4	Design	1%	Wed 6/7/23	Tue 6/20/23	Team B
1.4.1	Strategy Meeting	0%	Wed 6/7/23	Wed 6/7/23	Team B
1.4.2	TDD	2%	Thu 6/8/23	Sat 6/17/23	DB,CH,BI,SM,A T
1.4.2.1	Software Reference Architecture	5%	Thu 6/8/23	Sat 6/17/23	DB,CH,BI,SM,A T
1.4.2.2	Database Modeling	0%	Thu 6/8/23	Sat 6/17/23	DB,CH,BI,SM,A T
1.4.2.3	Network Architecture Design	0%	Thu 6/8/23	Sat 6/17/23	DB,CH,BI,SM,A T
1.4.3	Test Plan	0%	Thu 6/8/23	Sat 6/17/23	RD,JH,ML,KM,J N
1.4.3.1	Introduction / Background	0%	Thu 6/8/23	Thu 6/8/23	RD,JH,ML,KM,J N
1.4.3.2	Test Suite Development	0%	Thu 6/8/23	Mon 6/12/23	RD,JH,ML,KM,J N
1.4.3.3	Initial Test Report	0%	Tue 6/13/23	Sat 6/17/23	RD,JH,ML,KM,J N
1.4.4	Prototyping	0%	Wed 6/7/23	Sat 6/17/23	DB,CH,BI,SM,A T
1.4.4.1	Flutter Demos	0%	Wed 6/7/23	Sat 6/17/23	DB,CH,BI,SM,A T
1.4.4.2	ChatGPT API	0%	Wed 6/7/23	Sat 6/17/23	DB,CH,BI,SM,A T
1.4.4.3	Speech-to-Text	0%	Wed 6/7/23	Sat 6/17/23	DB,CH,BI,SM,A T

1.4.4.4	Audio File Local Storage Access	0%	Wed 6/7/23	Sat 6/17/23	DB,CH,BI,SM,A T
1.5	Development	0%	Wed 6/21/23	Tue 7/25/23	Team B
1.5.1	Strategy Meeting	0%	Wed 6/21/23	Wed 6/21/23	Team B
1.5.2	Product Backlog	0%	Thu 6/22/23	Tue 7/25/23	Team B
1.5.2.1	Finalize Backlog	0%	Thu 6/22/23	Sun 6/25/23	Team B
1.5.2.1.1	CRUD - Metadata	0%	Thu 6/22/23	Sun 6/25/23	Team B
1.5.2.1.2	Search through audio recordings	0%	Thu 6/22/23	Thu 6/22/23	Team B
1.5.2.1.3	CRUD - All Recordings	0%	Thu 6/22/23	Thu 6/22/23	Team B
1.5.2.1.4	Speech-to-Text Audio File Conversion	0%	Thu 6/22/23	Thu 6/22/23	Team B
1.5.2.1.5	CRUD - Short Code for Browser Extension	0%	Thu 6/22/23	Thu 6/22/23	Team B
1.5.2.1.6	Collect Field Names to Query ChatGPT to Retrieve Data to be Auto-filled	0%	Thu 6/22/23	Thu 6/22/23	Team B
1.5.2.1.7	Query to ChatGPT to Create Food Order from Audio Recording	0%	Thu 6/22/23	Thu 6/22/23	Team B
1.5.2.1.8	Query to ChatGPT to Find Reminders	0%	Thu 6/22/23	Thu 6/22/23	Team B
1.5.2.1.9	Create Reminder on Local Device from Audio Recordings	0%	Thu 6/22/23	Thu 6/22/23	Team B
1.5.2.1.10	Arbitrary Query to ChatGPT from Audio Recordings	0%	Thu 6/22/23	Thu 6/22/23	Team B
1.5.2.1.11	CRUD - User Profiles	0%	Thu 6/22/23	Thu 6/22/23	Team B
1.5.2.1.12	Display User Profile Info on UI	0%	Thu 6/22/23	Thu 6/22/23	Team B
1.5.3	Update Backlog	0%	Mon 6/26/23	Tue 7/25/23	Team B
1.5.4	Programmer Guide	0%	Thu 6/22/23	Tue 7/25/23	Team B
1.5.5	Runbook	0%	Thu 6/22/23	Tue 7/25/23	Team B
1.5.6	Final Product	0%	Mon 7/3/23	Tue 7/25/23	Team B
1.5.6.1	Testing	0%	Mon 7/3/23	Mon 7/17/23	A /B
1.5.6.1.1	Unit Testing	0%	Mon 7/3/23	Sun 7/9/23	Team B
1.5.6.1.2	Integration Testing	0%	Mon 7/10/23	Mon 7/17/23	Team A
1.5.7	Rework	0%	Tue 7/18/23	Tue 7/25/23	Team B
1.6	Implementation	0%	Wed 7/26/23	Tue 8/8/23	Team B
1.6.1	Strategy Meeting	0%	Wed 7/26/23	Wed 7/26/23	Team B
1.6.2	Release	0%	Thu 7/27/23	Thu 7/27/23	Team B
1.6.3	Post-Release Testing	0%	Fri 7/28/23	Mon 7/31/23	Team B
1.6.4	Finalize User Guide	0%	Thu 7/27/23	Mon 8/7/23	Team B
1.6.5	Test Reports	0%	Tue 8/1/23	Mon 8/7/23	Team B

9. Works Cited

- Adobe. (March 18, 2022). Waterfall methodology: Project management | Adobe Workfront. Waterfall Methodology: A Complete Guide. Retrieved May 27, 2023, from <https://business.adobe.com/blog/basics/waterfall>
- Cascella, M., Khalili, Y. (July 21, 2022). Short Term Memory Impairment. Retrieved May 23, 2023, from <https://www.ncbi.nlm.nih.gov/books/NBK545136/>
- Glossop, A. (Jul 23, 2021). Risk matrix: what is it and should you use one? Retrieved May 27, 2023, from <https://www.ideagen.com/thought-leadership/blog/risk-matrix-what-is-it-and-should-you-use-one>
- Healthline Media (October 6, 2021). What to know about short-term memory and short-term memory loss. Retrieved May 23, 2023, from <https://www.medicalnewstoday.com/articles/short-term-memory>
- Sampson, R., Gartell, B., Johnpeter, S., De Jesus, S., Srinivasan, A., & Detoito, A. (July 29, 2022). Project Plan: USPS Informed Deliver App – Visually Impaired. Retrieved June 10, 2023, from <https://umgc-cappms.azurewebsites.net/download/d809d25a-9b76-427c-9920-71560dac115b---Team-A-ProjectPlan-Revised.docx>
- Tucci, L. (January 2023). What is risk management and why is it important? Retrieved May 25, 2023, from <https://www.techtarget.com/searchsecurity/definition/What-is-risk-management-and-why-is-it-important>