PROJECT PLAN

Team Amazing

Presented by: Ayodeji Famudehin, Chauntika Broggin, Christian Ahmed, Mitchell Olshansky, Mo Drammeh, Nicholas Ballo, Shawn Kelly..

Revision 1.2

Project name: Memory Enhancer Application
Date: July 24, 2021
Project Leader: Shawn Kelly
Phase: Project Development

Revision History

Version Number	Date	Description	Sign Off
1.0	June 1, 2021	Initial Release	Shawn Kelly
1.1	July 2, 2021	Updated for Milestone 2. Added Software Test Plan section. Added details based on professor feedback. Updated budget and Labor based on recent sprint planning	Shawn Kelly
1.2	July 24, 2021	Updated for Milestone 2. Move table headers to top of table to comply with APA format	Shawn Kelly

Table of Contents

1. General Information

1.1. Background

About 14 million people in the United States have a disability that impairs their short-term memory. Short-term memory (STM) indicates different systems of memory involved in the retention of pieces of data for a relatively short time (usually up to 30 seconds). The causes of the disability range from aging to dementia to Alzheimer’s. According to the U.S. Centers for Disease Control and Prevention, noticing a decline in your mental abilities ("cognitive decline") is one of the earliest signs of impending Alzheimer's disease or dementia. There is a need to provide a solution to adults that assists them with the short-term memory function. To meet this need, United Global Masters Coders proposes the development of a mobile application that recognizes the speech of the user as they engage in a conversation and help them live a fuller life despite their disability.

1.2. Statement of Need

The purpose of this project is to develop the Memory Enhancer App for people with short-term memory disabilities. There are several ways to develop this app and this effort is to determine which way would be the most optimal. The Memory Enhancer App meets the business need by providing a solution for a critical and underserved market segment to have a solution with them that meets their need for memory assistance.

1.3. Vision Statement

The vision for United Global Masters Coders is to make the best personal speech recognition application for the memory impaired.

2. Project Assignment

2.1. Project Scope

For those with short-term memory loss, it can be frustrating and difficult retaining details from everyday life. One such difficulty is remembering conversation details. Team Amazing will help alleviate some of this difficulty by developing a Memory Enhancer App mobile application. The cross-platform application will assist users with remembering important information from their daily conversations, such as dinner plan details or appointment dates and times. The application will record these details for the user to review when needed.

The application will listen for keywords/phrases within the user's conversation that will trigger the conversion of those details to text for recording to the app. The user will also have the ability to press a button to start conversion. The app will convert only the user's speech to text to ensure privacy rights are not violated. Also, the user will have the ability to edit and/or delete information from the app.

The Memory Enhancer App will be developed using Flutter, an open-source UI toolkit, and the Dart programming language. The app will be built for the Android operating system. Google Cloud Platform's (GCP) Speech-to-Text API and Text-to-Speech API will be used to transcribe user's speech. Also, GCP voice recognition will be used to learn the user's voice.

Within scope of project:

- A functional 508-compliant GUI shall be designed and built.
- GCP's voice recognition and speech-to-text APIs will be the used in building the application.
- Mobile application shall be compatible with the Android and iOS operating systems.
- The application shall be tested using a software test plan.

Out of scope of project:

- A database system will not be built for the application.
- The project will not include user profiles for the user.

2.1.1. Project Objectives

The objectives listed below adhere to the "SMART" criteria (specific, measurable, attainable, relevant, and time-bound).

- Develop the Memory Enhancer App, by August 10, 2021, that will capture important details from conversations of users with short-term memory disabilities.
- Develop a 508-compliant application by August 10, 2021.
- To turn in each milestone deliverable on the scheduled time and date during the project.
- Design and develop a functioning graphical user interface (GUI) by June 29, 2021.
- To identify and record transcribed speech of user by implementing speech-to-text API and voice recognition by end of project.
- Ensure that the app is available to Android and iOS mobile users by end of project.

- Provide a fully tested and functioning app utilizing software test plans by the end of the project.

2.1.2. Project Requirements

The Memory Enhancer App's functional and non-functional requirement are presented below.

Functional Requirements

1. The application shall be trained to recognize the user's voice.
2. The application shall be Section 508 compliant such that an elderly person may use it without eyeglasses.
3. The application shall recognize the user's unique voice and phrases then will automatically begin recording a memo.
4. The application should ignore everything except what the user speaks.
5. The application shall feature a turn-on button to start capturing user's speech.
6. The application shall record transcribed text.
7. The application shall replay transcribed text to user when the user requests with unique phrases.
8. The application shall pair notes with subject of conversation.
9. The application shall not save any recorded voice audio.
10. The application shall allow user to edit and update text.
11. The application shall have a flexible and functioning GUI that is user friendly.
12. The application shall provide embedded training video guides to display the apps functionalities.
13. The application shall encrypt the user's notes.
14. The application shall work on both iOS and Android platforms.
15. The application shall allow the user to edit phrases to begin and end recording a memo.
16. The application shall allow the user recall memos using phrases.
17. The application shall save notes to local storage.
18. The application shall have a search function in notes.
19. The application shall keep notes for 1 week.
20. The application shall recognize phrases that end the recording memo.

Non-functional Requirements

1. The application shall integrate with APIs to support speech-to-text, text-to-speech, and wake-up words.

2. The application shall be built using the Dart programming language.
3. The application's user interface (UI) will be built utilizing Flutter, an open-source UI toolkit.
4. The application shall assume permission to begin recording.

2.2. Project Assumptions

The project assumptions are broken down into categories and listed below:

Team

- Team members may assist those in other roles but will stay in their assigned roles throughout the life of the project.

Scope

- The scope may change after the acceptance and approval of the Project Plan by the Customer, Dr. Mir Assadullah. Variations will be handled according to the Change Management Plan
- All changes shall be logged in a change log and approved by the Customer and the Project Manager.
- All milestones' deliverables will be approved by Dr. Mir Assadullah.

Methodology

- The project will follow the Scrumban methodology through the project's life cycle.
- Weekly sprints shall be conducted for the time-sensitive (12 weeks) project.
- Daily meetings (status updates, posts, or chats) shall be given to keep track of team members' tasks and their progress.

Budget

- No budget will be allocated for Hardware or Software needs.
- Labor costs will amount to an estimated budget of $55,660.

Technology

- The project will be written in the Dart programming language.
- The project will use Flutter, an open-source UI kit, to build the app.

End User

- The user will have access to a mobile device.
- The application will be available for Android and iOS mobile device users.
- The user has the proper equipment (device microphone, Bluetooth microphone, or wired audio device) for audio input and output.
- The user's device is able to connect to the Internet.
- The user is able to operate the mobile device correctly.

2.3. Product Approval/Acceptance Criteria

The project consists of four Milestones with scheduled deadlines and expected deliverables. At the end of each milestone, the Customer shall review and grade each deliverable. If the deliverable needs editing, then those edits will be included in the next Milestone deliverable. At the end of the project, the final, functioning product will be approved by the Customer. Also, all changes will be logged into a Change Log.

2.4. Project Costs

The estimated costs are shown below in Table 1. The costs are calculated on cost of labor alone, approximately 3 hours a day, 5 days a week, for 12 weeks (hourly rate shown in bulleted list*). The software used in the project is available at no cost, and the project does not require any hardware acquisition or installation.

- **Project Manager:** $62.00hr
- **Business Analyst:** $44.00hr
- **Lead Developer:** $53.00hr
- **Developer (3):** $46.00hr
- **Tester:** $42.00hr

***Hourly rates found on ZipRecruiter.com**

Table 1: Project Costs

Category	Total Estimated Costs
Labor	$66,444
Software	$0
Hardware	$0

2.5. Acronyms and Abbreviations

Table 2 lists the acronyms and abbreviations used in this this project plan to describe certain information, features, and components of the Memory Enhancer App.

Table 2: Acronyms and Abbreviations

Acronyms and Abbreviations	Meaning
API	Application Programming Interface
FT	Functional Testing
GCP	Google Cloud Platform
GUI	Graphical User Interface
IDE	Integreated Development Environment
NLP	Natural Language Processor
RT	Rgression Testing
SRS	Software Requirements Specification
STM	Short-term memory
UAT	User Acceptance Testing
UI	User Interface
WBS	Work Breakdown Structure

2.6. References

Below is the list of references used to complete this project plan.

Anonymous. (2021, April 8). Definitive Guide to Implement Scrumban Methodology. Kissflow. https://kissflow.com/project/agile/guide-to-scrumban-methodology/

Bartlett, J. (2020, October 29). What is a Test Plan in Software Testing? TestLodge Blog. https://blog.testlodge.com/what-is-a-test-plan-in-software-testing/.

Diaz, J. (2013). Voice recognition system. http://web.mit.edu/6.111/www/s2007/PROJECTS/14/Project14.pdf

Kibuuka, R. (2013). Project schedule management. https://www.researchgate.net/publication/273759807_Project_Schedule_Management

PMI (2017). *A Guide to the Project Management Body of Knowledge: PMBOK® Guide* (6th edition). Project Management Institute.

Santos, J. M. D. (2021, May 18). Understanding Responsibility Assignment Matrix (RACI Matrix). Project. https://project-management.com/understanding-responsibility-assignment-matrix-raci-matrix/.

Srivastava, K. (2015). Effective risk management technique in development of IVR software. *International Journal of Computer Science, 4*(SPL-2), 87-90. http://www.ijcst.com/vol4/spl2/c0043.pdf

3. Risk Analysis

3.1. Risk Management Framework

The Memory Enhancer Application allows the user to interact with the application in an automated fashion, through voice and touch-tone mobile phone or device. However, the Memory Enhancer Application needs extra security to make it a reliable service for the user.

The basic risk analysis and management framework for the project involved the following steps.

- Identify, characterize, and assess the threats.
- Assess the vulnerabilities of critical assets to those threats.
- Determine the associated risk: The likelihood and consequence of attacker and threats.
- Identify mitigating measure to reduce risks.
- Prioritize risk reduction measure based on strategy that respects the interest and requirements of all constituents.

3.2. Risk Analysis and Vulnerabilities Assessment

In order to complete the risk and vulnerability assessment, it is very important to identify critical asset, risk vulnerability and threats related to the Memory Enhancer Application. Risk is the probability of losing a valuable asset of the system, while vulnerability is the system's weakness which can allow an attacker to reduce the system information assurance.

The identified threats and vulnerabilities for the project are highlighted and briefly discussed below.

1. **Improper system configuration:** This vulnerability will appear when there is mis match between actual system configuration required and the configuration being used.

2. **Improper results due to different platform:** The application should function exactly same on any mobile devices irrespective of platform used. Whether deployed on Android, Mac OS it should give the same result.
3. **Loss of pre-existing functionality:** In case any new functionality is added, there should not be loss of pre-existing functionalities.
4. **Use of proper voice recognition Engine:** Proper voice recognition engine should be use. Faulty voice recognition engine can cause system crash.
5. **Version mismatch between voice recognition Engine and voice packs:** There should be proper compatibility between voice recognition engine and voice peaks.
6. **Underlaying operating system or firmware DoS:** Most of the underlying Operating System for the system will run on popular operating system or firmware that regularly becomes vulnerable to new threats, example: Viruses.
7. **Network Service Dos:** This defines the threat that an attacker targets network component or service that the Memory Enhancer Application service depends on. Example, attacker can flood routers, switches, proxies, etc. to make them unable to function properly.
8. **App crash:** App crash will deny users of any service unless there is some backup storage in place.

Based on the identified vulnerabilities the following risk assessment has been done.

Table 3: Vulnerabilities, Threats and Risk Summary

No	Vulnerability	Threat	Risk Summary
1	Improper system configuration	Denial of Service	Loss of Availability
2	Platform dependency	Variation in output	Improper result
3	Loss of pre-existing functionality due to enhancement	Functionality Manipulation	Loss of Availability & Improper result
4	Improper Voice Recognition engine	System crash	Loss of Availability
5	Mismatch between Voice Recognition Engine & Voice packs Versions	System cannot recognize a particular Voice packs.	Improper or no result

6	Underlaying Operating system or firmware	System crash	Improper or no result
7	Network services DoS	System unable to function properly	Improper or no result
8	Generation of personal information in log files.	Information theft	Confidential loss
9	Loss of power	Deny endpoint services	No result

Table 4: Some other Vulnerability base on existing similar Technology

No	Vulnerability	Threat	Risk Summary
1	Improper use of device System	Malicious use by Unauthorized person	Users' data loss
2	Poorly written Application	Unauthorized access of important code	Data loss
3	Spy ware	Unauthorized access	Data loss
4	Vendor back doors	Unauthorized access	Data loss
5	Stolen credentials	Unauthorized access	Compromised confidentiality.
6	Software bugs	Unauthorized access	Data loss
7	Speech to text system	Server access. Poor user authentication	Hacker might enter the system.

3.3. Risk Matrix and Mitigation

In the risk assessment process, assets and threats are identified. Impact or probability of each threat along with the vulnerability of assets from a particular threat are assigned a value from low to high**.** When the system is successfully deployed, some measures need to be taken in order to approach the reliability that the system offers. Therefore, this section also identifies and highlights actions available to significantly reduce risks involved with the Memory Enhancer Application system. While some actions are general and do not counter any specific threat directly, other actions are strictly aimed at countering specific threats.

Based on already identified Vulnerabilities, the risk summaries, impact ratings are listed along with suggested mitigations in the table below.

1	Vulnerability	Risk Summary	Impact Rating	Mitigation Technique
2	Improper system configuration	Loss of Availability on integrity	High	Proper check & validation point at the time of development.
3	Platform dependency	Loss of Confidentiality	High	System to be provided to authorized person only.
4	Loss of pre-existing functionality due to enhancement	Loss of Users important information	Moderate	Backup system is required.
5	Improper Voice Recognition engine	Damage important data	High	Proper user access control is required
6	Mismatch between Voice Recognition Engine & Voice pack versions	Unauthorized access	Moderate	Proper user access control is required
7	Generation of personal information in log files	Compromise confidentiality	Moderate	User ID and password should be confidential
8	Network services DoS	Improper or no result	High	Regular software update and put anti-virus system in place
9	Underlying Operating system or firmware DoS	Improper or no result	High	Regular software update and put anti-virus system in place

10	Speak to text system	Poor credential can allow hacker to access the system	High	Proper user access control is required
11	Loss of pre-existing functionality in case of enhancement	Financial loss and loss of important user data	High	Implement applicable Software policy

4. Organization of the project

Team Amazing will organize project tasks using the Scrumban methodology. The project tasks are kept tracked using GitHub's project section in the team's development branch. This area allows the team to breakdown the milestone deliverables into individual tasks. All members will have access to the progress of the project, in addition to individual items at-a-glance. All of this will be accessible without needing to get into an intimate knowledge of the task and the individuals assigned to the tasks for completion.

4.1. Communication

The project team will maintain daily communication over Microsoft Teams through the chat features. This daily communication ensures that the team is on track with milestone goals. Weekly Teams calls allow the team to discuss progress of the action items and discuss what is required to complete the project.

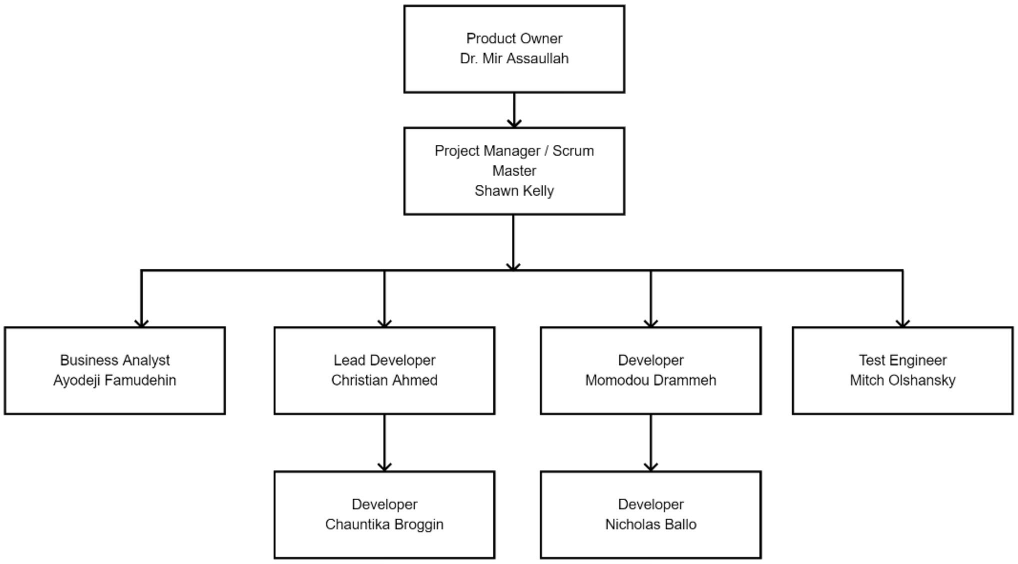
The project milestones can be tracked through Office 365 where the documents are kept on team member's student OneDrive. This allows collaboration between team members to run smoothly since all members can view and edit the documents at the same time.

4.2. Explanation of the methodology

The Scrumban methodology uses a combination of Scrum and Kanban. The Scrum methodology splits deliverables into tasks that are granular enough to complete in each sprint (Anonymous, 2021). A sprint is a period of one to four weeks allocated for a single task. Kanban is a visual format that represents the organization of tasks on a project. Tasks can be in four states of completion: backlog, to do, in progress, and complete. The combination of Scrum and Kanban is known as Scrumban. Team Amazing’s sprints will each last two weeks in order to meet the project milestones on time.

4.3. Project Staffing

Team Amazing is staffed properly for this project with a team that includes a Project Manager, Business Analyst, Lead Developer, Tester, and four Developers. The Project Manager will also function as the Scrum Leader to allow all the developers to focus on producing the best product.



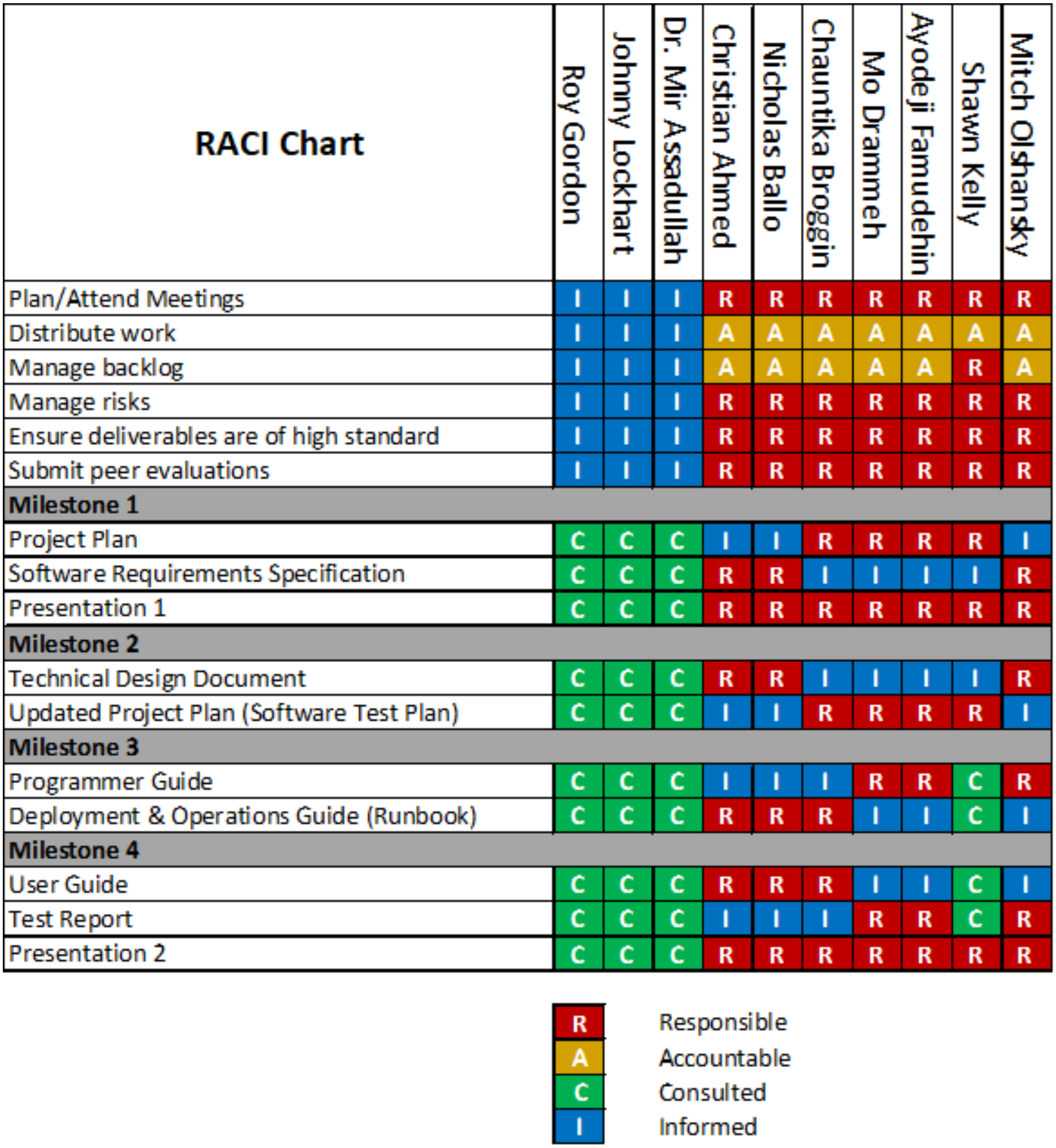
4.4. Roles and Responsibilities

Each member will be responsible for sections of a project based on their role but are not limited to those responsibilities. Members may be asked to take on additional responsibilities if required in order to complete the project successfully.

Role	Assignee	Responsibility
Project Owner	Dr. Mir Assadullah	Sets the project’s end product and the parameters to consider the project successful. Is available for clarification in project’s features.
Project Manager	Shawn Kelly	Organize the team and ensure the team is informed with the goals and logistics of the project.

Business Analyst	Ayodeji Famudehin	Ensures the project is on time, budget, and meets the requirements.
Lead Developer	Christian Ahmed	Dividing up the project tasks into sprints. Hold code reviews with the fellow developers and ensure the product meets the requirements
Developer	Chauntika Broggin Mo Drammeh Nicholas Ballo	Designing, coding, developing, and debugging the application.
Tester	Mitch Olshansky	Create the test plans and automated tests for the program. This includes stress testing and limit testing.

4.5. Responsibility Assigned Matrix (RACI)



RACI Chart	Roy Gordon	Johnny Lockhart	Dr. Mir Assadullah	Christian Ahmed	Nicholas Ballo	Chauntika Broggin	Mo Drammeh	Ayodeji Famudehin	Shawn Kelly	Mitch Olshansky
Plan/Attend Meetings	I	I	I	R	R	R	R	R	R	R
Distribute work	I	I	I	A	A	A	A	A	A	A
Manage backlog	I	I	I	A	A	A	A	A	R	A
Manage risks	I	I	I	R	R	R	R	R	R	R
Ensure deliverables are of high standard	I	I	I	R	R	R	R	R	R	R
Submit peer evaluations	I	I	I	R	R	R	R	R	R	R
Milestone 1										
Project Plan	C	C	C	I	I	R	R	R	R	I
Software Requirements Specification	C	C	C	R	R	I	I	I	I	R
Presentation 1	C	C	C	R	R	R	R	R	R	R
Milestone 2										
Technical Design Document	C	C	C	R	R	I	I	I	I	R
Updated Project Plan (Software Test Plan)	C	C	C	I	I	R	R	R	R	I
Milestone 3										
Programmer Guide	C	C	C	I	I	I	R	R	C	R
Deployment & Operations Guide (Runbook)	C	C	C	R	R	R	I	I	C	I
Milestone 4										
User Guide	C	C	C	R	R	R	I	I	C	I
Test Report	C	C	C	I	I	I	R	R	C	R
Presentation 2	C	C	C	R	R	R	R	R	R	R

R	Responsible
A	Accountable
C	Consulted
I	Informed

Figure 1: RACI Chart

4.6. Project Tools

The tools used to complete this project is shown in the table below.

Table 8: Project Tools

Tool	Description
Android Studio	An integrated development environment (IDE) used to develop Android applications.
Dart	It is a client-optimized language that allows for the flexible and fast production of multi-platform applications.
Flutter	An open-source UI toolkit developed by Google.
GitHub	Offers code repositories that will help the team organize code and communicate during project development.
Microsoft Teams	A business communication platform by Microsoft that will allow the team to communicate, share documents, and more during the project.
Microsoft Word	A word-processing software by Microsoft that allows document preparation. All document preparation will be done in this software.

4.7. Project Storage

Team Amazing's documents/deliverables will be stored in Microsoft Teams. The documents will be collaborated on by team members using SharePoint within the Microsoft Teams environment. A GitHub repository has been created to remotely store the team's application code during project development.

4.8. Deliverables

The project is divided into four milestones. Milestone One is the initiation phase where the Project Plan and the Software Requirements Specification (SRS) are developed and delivered. Milestone Two is when the Technical Design Document is created and delivered, as well as adding the Software Test Plan to the Project Plan. Milestone Three's deliverables are the Programmer Guide, the Deployment and Operations Guide (Runbook), and, if needed, an updated Project Plan. Finally, Milestone Four is when the finished application is turned in, as well as an updated Project Plan, a User Guide, and a Test Report.

Table 9: Deliverables

Deliverable	Description	Due Date
Milestone 1		June 11, 2021
Project Plan	Initial plan for the project, including project scope, requirements, risks, objectives, etc.	

Software Requirements Specification (SRS)	Breakdown of the requirements for the application	
Milestone 2		July 2, 2021
Technical Design Document	The design document that describes the minute technical details of the application.	
Updated Project Plan	Updated Project Plan with the Software Test Plan added and any changes that need to be made.	
Milestone 3		July 23, 2021
Programmer Guide	A guide for programmers of the Assistance application that has coding standards and conventions.	
Deployment & Operations Guide (Runbook)	A guide on how to deploy and operate the application.	
Updated Project Plan (if needed)	Project Plan with any necessary changes.	
Milestone 4		August 10, 2021
Updated Project Plan	Project Plan with any necessary changes.	
User Guide	A guide to show users how to use the application.	
Test Report	A report that shows the results from the software test plan.	
Finished and Functioning App	Finished Memory Emhancer application	

4.9. Project Schedule/WBS

The time management plan for this project provides an outline of the project schedule. The schedule is required to understand the resources requirements for the completion of this project and identifies who will perform the specific task for the completion of the project via a Gantt chart.

Breakdown of tasks and deliverable for this project, timeline, and dates of completion for specific tasks are provided by the Gantt chart in Appendix A.

4.10. Change Management Plan

As we progress through the project, modifications or additions will inevitably come forth and require coordination among those involved in the project, including the Development and DevSecOps teams and the Project Managers to be able to adapt to the change. These changes may include variance in scope, added features, patching security vulnerabilities, or several other potential changes to requirements and the project. All changes and updates must go through this process.

Once the modification has been identified, the development teams will work to determine the scope of the change and several factors such as the level of effort, importance, risks, among others. After the initial investigation is complete, the teams will take their findings to each of the Project Managers for them to discuss alongside the DevSecOps team to determine the next steps and further compile the requirements for the change. The resulting information will then be presented to the stakeholder to determine its worth and prioritize the change. If the change is agreed to be taken on, it will be recorded, and all impacted documents will be updated accordingly.

The sequential approach is designed to ensure that all teams are made aware of the potential change and are on the same page for the entirety of the project. Communication for any impending changes as well as other Project tasks will primarily occur over Microsoft Teams with some communication occurring via email.

5. Software Test Plan

5.1. Purpose

This test plan describes the steps taken to support the deployment of the Memory Enhancement Application for Android and iOS phones. The Memory Enhancement Application will be introduced to assist individuals with impaired short-term memory and age-related cognitive impairment. This document describes the steps taken to ensure that the application does the following:

- Interoperates properly with all required internal and external components.
- Operates in accordance with the Project Objectives outlined in earlier sections of the Project Plan.
- Abides by all requirements defined in the Software Requirements Specification document.
- Integrates successfully into both Android and iOS environments.

Through this plan, Team Amazing ensures that all inspection and testing is performed in compliance with contract requirements and that all test data is complete, correct, traceable, repeatable, and acceptable.

This plan is developed in close cooperation within Team Amazing. It reduces risk by incremental system integration and regular testing to ensure the success of the project. Commercial testing ensures quality and standards are built into the Memory Enhancement Application.

5.2. Scope

The test program is developed and implemented following structured phases consisting of:

- **Functional Testing** to identify any issues or bugs found during the project life cycle based on functional requirements in tandem with application development.
- **Regression Testing** to confirm previous build fixes are intact after new build is deployed.
- **User Acceptance Testing** to test application from end user perspective, including 508 compliance.

Functional Testing (FT) – Team Amazing shall conduct Functional Testing (FT) to ensure that the requirements from the SRS (Software Requirements Specification) document are met. Team Amazing will conduct Functional Testing remotely from our homes and on-site lab. Testing will be considered complete and successful based on Team Amazing's demonstration of the product's ability to fulfill the Specification requirements to the satisfaction of the customer. All necessary quality assurance policies will be put into effect during the FT phase.

Regression Testing (RT) – Team Amazing shall conduct Regression Testing (RT) remotely and at on-site lab to ensure that all Specification technical requirements are met. All opened bugs will be addressed in iterative Agile-like setting utilizing Kanban methodology on GitHub. Testing and development will occur in tandem, with builds being pushed and deployed in order to ensure any previously resolved issues stay resolved, and that new functionality is implemented correctly for usability. Testing will be considered complete and successful based on Team Amazing's demonstration of the product's ability to fulfill the Specification requirements to the satisfaction of the customer. All necessary quality assurance policies will be put into effect during the RT phase.

Focus Group – The Focus Group will consist of volunteer testers outside of Team Amazing. Their mission will be to "break" the application and discover any bugs that may occur. We will ask the Focus Group to test on real hardware across different versions of iOS and Android for maximum test coverage.

User Acceptance Testing (UAT) – This will be the final stage before application deployment; Team Amazing conduct User Acceptance Testing (UAT) from end-user perspective to ensure users with impaired cognitions, sight-challenged vision, etc. will be able to comfortable use the application on any iPhone and Android phone. In addition to utilizing emulators, we will also be installing the application on our personal mobile devices for additional UAT support. The test procedures may be based on the FT procedures, subject to final definition by the customer. All necessary quality assurance policies will be put into effect during the UAT phase.

Automated Testing – via Selenium IDE (happy paths, non-functional scenarios, etc.). Automated testing suites will initially take some more time to set up but will dramatically improve speed of regression test case execution. Manual testing which normally takes minutes, will be complete in seconds.

5.3. Test Items

5.3.1. To Be Tested

1. The application shall be trained to recognize the user's voice.
2. The application shall be Section 508 compliant such that an elderly person may use it without eyeglasses.
3. The application shall recognize the user's unique voice and phrases then will automatically begin recording a memo.
4. The application should ignore everything except what the user speaks.
5. The application shall feature a turn-on button to start capturing user's speech.
6. The application shall record transcribed text.
7. The application shall replay transcribed text to user when the user requests with unique phrases.
8. The application shall pair notes with subject of conversation.
9. The application shall not save any recorded voice audio.
10. The application shall allow user to edit and update text.
11. The application shall have a flexible and functioning GUI that is user friendly.
12. The application shall provide embedded training video guides to display the apps functionalities.

5.3.2. To Not Be Tested

1. The application shall integrate with Google's Speech-to-Text API.
2. The application shall run on the Android operating systems for mobile devices.
3. The application shall be built using the Dart programming language.

4. The application’s user interface (UI) will be built utilizing Flutter, an open-source UI toolkit.

5.4. Approach

All bugs/defects will be reported on the GitHub Scrumban 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 testing for this project, representing the largest number of tests as well as the most complete code coverage. Unit tests will be nearly, if not entirely, manual due to the necessity of a human voice when using the application’s features.

Integration/Regression Testing – As individual components of the application are built, we must ensure a robust testing cycle to help integrate the application with local device and minimize re-work and defects. With each new software build, the team will ensure that new code, will not impact currently existing functionality. Both integration and regression testing can be achieved with assistance from test automation suite via Selenium IDE. Although more prep work is required, automated testing will significantly reduce effort later. 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), we will be close to signing off on completion of the application. Real individuals who the Memory Enhancement Application is aimed at will be testing on the platform of their choosing (iOS or Android) and we will receive feedback for “future” versions.

5.5. Pass/Fail Criteria

- Tests relating to display and navigation within the application shall be considered a success once no noticeable bugs occur after 10 repetitions of the test.
- Tests relating to voice recording shall be considered a success if more than 90% of speech is correctly transcribed.

5.6. Defect Severity Levels

1. Low – Meant for non-functional, non-essential issues that do not take away from functional or performance requirements (i.e., minor text/GUI bugs).

2. Moderate – Could be for both functional and non-functional but have a slightly greater impact on the mission of the application/development process.
3. Critical – Functional requirement bugs that need to be addressed immediately.
4. Blocker – Major issue that prevents any further work and development which completely impairs the entire project or developer from completing their task.

5.7. Risks & Contingencies

1. Messages that are not properly transcribed may mislead the user when played back in the future.
2. The application may begin recording at unwanted times which will lead to increased battery usage on the device.

5.8. Tests Cases

Test Case 1: Train Voice Recognition

Description:	**This test verifies that a user is able to train voice recognition within the application.**
Test Type:	User Acceptance
Requirement:	The application shall be trained to recognize the user's voice.
Req. #:	MEA001
Prerequisites:	
Steps:	1. User turns app on. 2. User speaks trigger phrases/words. 3. User repeats this process until application is trained. 4. User logs off application
Expected Output:	Application recognizes certain words to trigger events based on functional scenarios.
Assumption:	Application has been created.

Test Case 2: 508 Compliancy

Description:	This test verifies 508 compliancy in terms of font size, color contrast, etc. to support potential userbase.
Test Type:	508
Requirement:	The application shall be section 508 compliant such that an elderly person may use it without eyeglasses.
Req. #:	MEA002
Prerequisites:	None
Steps:	1. User turns app on. 2. User presses zoom in button. 3. Application font and GUI increases in size. 4. User presses zoom out button. 5. Application font and GUI decreases in size.
Expected Output:	GUI meets 508 standards through zoom in/out function and usability.
Assumption:	GUI is functional for a sight-challenged user.

Test Case 3: Automatic Voice Detection

Description:	This test verifies that the software will automatically begin transcribing the user’s voice once it is recognized that they are speaking.
Test Type:	Functional & User Acceptance
Requirement:	The application shall recognize the user’s unique voice and phrases then will automatically begin recording a memo.
Req. #:	MEA003
Prerequisites:	The app has been trained to recognize the user’s voice.
Steps:	1. The user turns app on. 2. The user begins speaking and mentions a phrase that the app has been trained to recognize.
Expected Output:	The application displays the user’s words as text on screen.
Assumption:	None

Test Case 4: Irrelevant Speech Avoidance

Description:	**This test verifies that irrelevant speech and noises will not be detected by the application.**
Test Type:	Functional & User Acceptance
Requirement:	The application should ignore everything except what the user speaks.
Req. #:	MEA004
Prerequisites:	The app has been trained to recognize the user's voice.
Steps:	1. The user turns app on. 2. The user either presses the record button or says a phrase that causes the app to begin recording. 3. Someone other than the user that the app has been trained for begins speaking.
Expected Output:	The app does not record or display the text of the non-user that is speaking.
Assumption:	None.

Test Case 5: Record Button Functionality

Description:	**This test verifies that speech is transcribed after the user has pressed the capture/record button.**
Test Type:	Functional & User Acceptance
Requirement:	The application shall feature a turn-on button to start capturing user's speech.
Req. #:	MEA005
Prerequisites:	The app has been trained to recognize the user's voice.
Steps:	1. The user opens the app. 2. The user presses the capture/record button. 3. The user begins speaking.
Expected Output:	The app transcribes and displays the user's speech as text.
Assumption:	None.

Test Case 6: Text Display

Description:	This test verifies that text is displayed on screen that matches the user’s speech.
Test Type:	Functional & User Acceptance
Requirement:	The application shall record transcribed text.
Req. #:	MEA006
Prerequisites:	The app has been trained to recognize the user’s voice.
Steps:	1. The user opens the app. 2. The user presses the capture/record button or says a phrase that causes the app to begin recording. 3. The user begins speaking.
Expected Output:	The app transcribes and displays the user’s speech as text.
Assumption:	None.

Test Case 7: Message Replay

Description:	This test verifies that the application is capable of replaying messages when upon the user’s request.
Test Type:	Functional & User Acceptance
Requirement:	The application shall replay transcribed text to user when the user requests with unique phrases.
Req. #:	MEA007
Prerequisites:	The user has previously recorded a message that is saved in the application.
Steps:	1. The user opens the app. 2. The user selects a message that has been saved. 3. The user says a phrase that the app has been trained to recognize for message replay.
Expected Output:	The application replays the selected message using text-to-speech software.
Assumption:	The user’s device is unmuted.

Test Case 8: Subject Recognition

Description:	This test verifies that the application recognizes certain subjects when recording speech.
Test Type:	Functional & User Acceptance
Requirement:	The application shall pair notes with subject of conversation.
Req. #:	MEA008
Prerequisites:	The app has been trained to recognize the user's voice.
Steps:	1. The user opens the app. 2. The user presses the capture/record button or says a phrase that causes the app to begin recording. 3. The records multiple messages regarding the same subject.
Expected Output:	The application groups the messages with the same subject together
Assumption:	The application has been trained to recognize the subject

Test Case 9: Automatic Data Deletion

Description:	This test verifies that no audio is stored on the user's device.
Test Type:	Functional
Requirement:	The application shall not save any recorded voice audio.
Req. #:	MEA009
Prerequisites:	The app has been trained to recognize the user's voice.
Steps:	1. The user opens the app. 2. The user presses the capture/record button or says a phrase that causes the app to begin recording. 3. The user records a message. 4. The user closes the app. 5. The user view's the devices saved data.
Expected Output:	The user does not find any saved audio from the application.
Assumption:	The user's device contains space to theoretically save audio files.

Test Case 10: Text editing

Description:	**This test verifies that the application allows the user to modify text that has been transcribed from speech.**
Test Type:	Functional & User Acceptance
Requirement:	The application shall allow user to edit and update text.
Req. #:	MEA010
Prerequisites:	The app has been trained to recognize the user's voice.
Steps:	1. The user opens the app. 2. The user presses the capture/record button or says a phrase that causes the app to begin recording. 3. The user records a message. 4. The user taps on the displayed message 5. An on-screen keyboard is displayed allowing the user to make modifications to the message. 6. The user makes modifications and presses "Save"
Expected Output:	The modified message is saved and displayed in the application
Assumption:	None

Test Case 11: Navigation

Description:	**This test verifies that the application is built intuitively so that users can learn to use it quickly.**
Test Type:	User Acceptance
Requirement:	The application shall have a flexible and functioning GUI that is user friendly.
Req. #:	MEA011
Prerequisites:	None.
Steps:	1. The user opens the app. 2. The user presses on the various menu screens (help, settings, etc.)
Expected Output:	The application displays each menu as the user selects them.
Assumption:	None.

Test Case 12: Training Videos

Description:	**This test verifies that the user can access training videos that explain how to operate the software.**
Test Type:	Functional & User Acceptance
Requirement:	The application shall provide embedded training video guides to display the apps functionalities.
Req. #:	MEA012
Prerequisites:	None
Steps:	1. The user opens the app 2. The taps the help menu icon 3. The user taps on a training video
Expected Output:	The application begins playing a video that explains how to operate the application.
Assumption:	None

Test Case 13: Google Speech-to-Text API Integration

Description:	**This test verifies that the software integrates with Google's Speech-to-Text software for speech transcription.**
Test Type:	Functional & User Acceptance
Requirement:	The application shall integrate with selected APIs.
Req. #:	MEA013
Prerequisites:	None.
Steps:	1. The application uses Google's Speech-to-Text software when transcribing the user's speech
Expected Output:	Text is displayed on screen after being transcribed with Speech-to-Text.
Assumption:	None.

Test Case 14: Android Capability

Description:	**This test verifies that the application can run on the Android operating system.**
Test Type:	Functional
Requirement:	The application shall run on the Android operating systems for mobile devices.
Req. #:	MEA014
Prerequisites:	The user is using an Android device.
Steps:	1. The user opens the app 2. The user ensures that the app's various features are functional
Expected Output:	The app works correctly on the Android device.
Assumption:	None.

Test Case 15: Built Using Dart Language

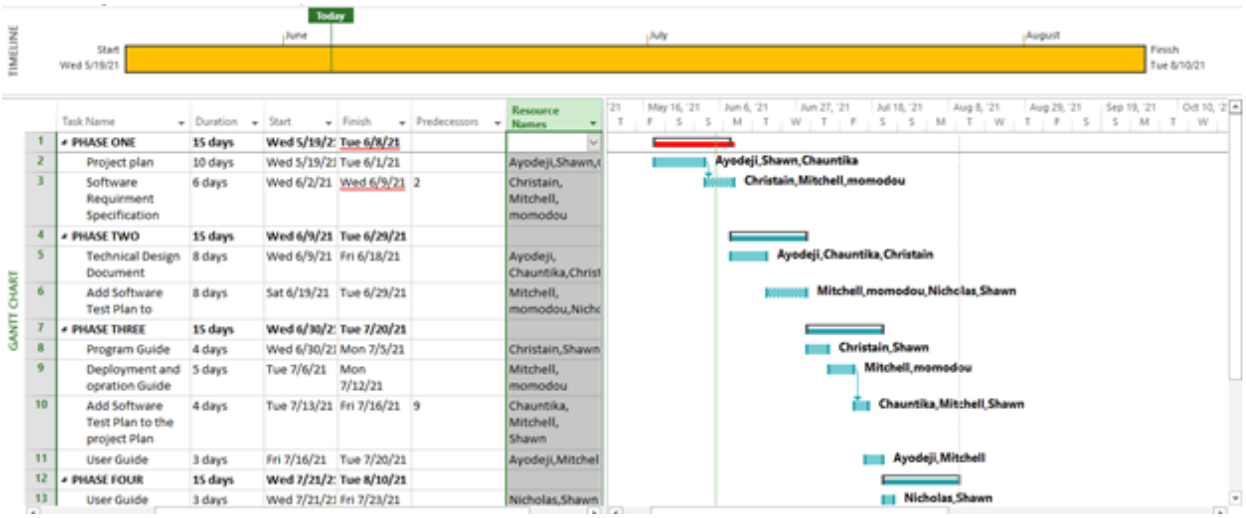
Description:	**This test verifies that the application has been coded in the Dart language.**
Test Type:	Functional
Requirement:	The application shall be built using the Dart programming language.
Req. #:	MEA015
Prerequisites:	None
Steps:	1. The application meets all requirements and is developed using the Dart language.
Expected Output:	None
Assumption:	None

Test Case 16: Built Using Flutter Framework

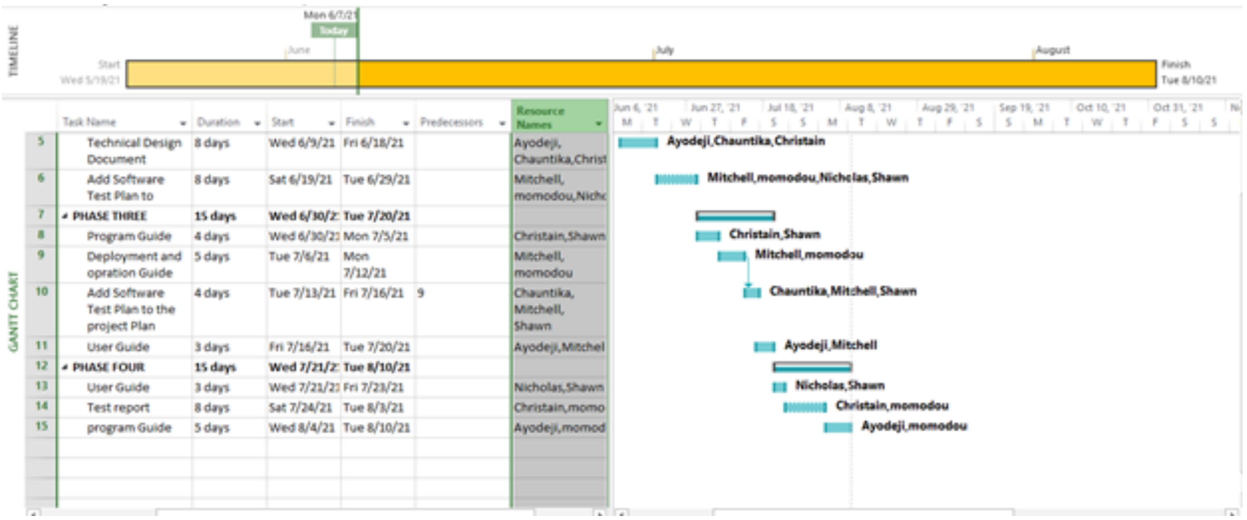
Description:	**This test verifies that the application utilizing the Flutter framework.**
Test Type:	Functional
Requirement:	The application’s user interface (UI) will be built utilizing Flutter, an open-source UI toolkit.
Req. #:	MEA016
Prerequisites:	None
Steps:	1. The application meets all requirements and is developed using the Flutter framework.
Expected Output:	None
Assumption:	None

6. Appendices

6.1. Appendix A: Project Schedule/WBS



	Task Name	Duration	Start	Finish	Predecessors	Resource Names
1	**PHASE ONE**	**15 days**	**Wed 5/19/2:**	**Tue 6/8/21**		
2	Project plan	10 days	Wed 5/19/21	Tue 6/1/21		Ayodeji,Shawn,(
3	Software Requirment Specification	6 days	Wed 6/2/21	Wed 6/9/21	2	Christain, Mitchell, momodou
4	**PHASE TWO**	**15 days**	**Wed 6/9/21**	**Tue 6/29/21**		
5	Technical Design Document	8 days	Wed 6/9/21	Fri 6/18/21		Ayodeji, Chauntika,Christ
6	Add Software Test Plan to	8 days	Sat 6/19/21	Tue 6/29/21		Mitchell, momodou,Nicho
7	**PHASE THREE**	**15 days**	**Wed 6/30/2:**	**Tue 7/20/21**		
8	Program Guide	4 days	Wed 6/30/21	Mon 7/5/21		Christain,Shawn
9	Deployment and opration Guide	5 days	Tue 7/6/21	Mon 7/12/21		Mitchell, momodou
10	Add Software Test Plan to the project Plan	4 days	Tue 7/13/21	Fri 7/16/21	9	Chauntika, Mitchell, Shawn
11	User Guide	3 days	Fri 7/16/21	Tue 7/20/21		Ayodeji,Mitchel
12	**PHASE FOUR**	**15 days**	**Wed 7/21/2:**	**Tue 8/10/21**		
13	User Guide	3 days	Wed 7/21/21	Fri 7/23/21		Nicholas,Shawn



	Task Name	Duration	Start	Finish	Predecessors	Resource Names
5	Technical Design Document	8 days	Wed 6/9/21	Fri 6/18/21		Ayodeji, Chauntika,Christ
6	Add Software Test Plan to	8 days	Sat 6/19/21	Tue 6/29/21		Mitchell, momodou,Nicho
7	**PHASE THREE**	**15 days**	**Wed 6/30/2:**	**Tue 7/20/21**		
8	Program Guide	4 days	Wed 6/30/2:	Mon 7/5/21		Christain,Shawn
9	Deployment and opration Guide	5 days	Tue 7/6/21	Mon 7/12/21		Mitchell, momodou
10	Add Software Test Plan to the project Plan	4 days	Tue 7/13/21	Fri 7/16/21	9	Chauntika, Mitchell, Shawn
11	User Guide	3 days	Fri 7/16/21	Tue 7/20/21		Ayodeji,Mitchel
12	**PHASE FOUR**	**15 days**	**Wed 7/21/2:**	**Tue 8/10/21**		
13	User Guide	3 days	Wed 7/21/2:	Fri 7/23/21		Nicholas,Shawn
14	Test report	8 days	Sat 7/24/21	Tue 8/3/21		Christain,momo
15	program Guide	5 days	Wed 8/4/21	Tue 8/10/21		Ayodeji,momod

6.2. Appendix B: Test Execution Layout

Test Name:	
Test Start Date & Time:	
Test End Date & Time:	
Test Conductor:	
Software Version:	
Test Environment:	
Witnesses:	
Issues Opened:	
Action Items Opened:	
Status:	**Pass ____** **Partial Pass ____** **Fail ____**
Test Conductor Signature / Date:	