Project Plan

TEAM: CHARLIE

Presented by: Michael Le, Debashis Jena, Austin Johnson, Prince Antwi Aboagye, Didimus Kimbi, Damion Sevilla

SWEN 670 – SOFTWARE ENGINEERING PROJECT
JUNE 11, 2021
REVISION 4.0

Project name: Mnemosyne, Disability Mobile Application

Date: July 1, 2021

Project Leader: Michael Le

Phase: Project Planning

For approval: Michael Le

Michael le Date: 07/01/2021

For approval: Dr. Mir Mohammed Assadullah

_____ Date: 07/02/2021

Revision History

Version Number	Date	Description	Approved By
1.0	06/11/2021	Initial Project Plan Release	Michael Le
2.0	07/01/2021	Revision Project Plan Release – After Design Phase	Michael Le
3.0	07/23/2021	Revision Project Plan Release – after Execution Phase	Michael Le
4.0	08/06/2021	Final Version	Michael Le

Table of Contents

General information about the project

1. Situation sketch and problem definition of the project

A. Background

a. Dr. Mir Assadullah created the Short-Term Memory Assistant mobile application for the UMGC Software Engineering Project course Capstone Project in the summer 2021 term. The Charlie team will be working with DevSecOps Team to complete the overall Capstone Project. Each team will have similar general requirements, and the business analyst will develop a high-level requirement to envision the team for a unique design/approach mobile application solution. After the Capstone Project, the customers (disabled with Short-Term memory) will use the application.
b. The Capstone Project's customers include disabled people from dementia to Alzheimer's. The public service professionals, such as doctors, medical analysts, etc., would benefit from the service using this memory-impaired mobile application.
c. The Charlie Team will be working on the intelligent mobile memory-impaired application remotely. UMGC Faculty staff choose DevSecOps (DSO) and Project Manager (PM) for each team. The PM is voluntarily choosing their team, which I select team Charlie in this case. PM evaluated each member's skills and established an outline such as tasking, scheduling, meeting, organizing, and leading the team to succeed at the end of the semester.

B. Statement of Need

a. Disabled people need a Short-Term Memory Assistant mobile application that will help them with short-term memory disabilities.

b. Presently, there is only a voice recorder that records the other person's voice. With the short-term memory assistant app, users will have the ability to read the text and hear their conversation repeatedly to remind them of the past event.

C. Vision Statement

a. This short-term memory assistant mobile application might be the first Artificial Intelligent Application to make the best personal speech recognition application for the memory impaired. It will offer public service professionals an application to listen to a conversation between an interviewer and interviewee (short-term memory disability). After that, the mobile app will capture the conversation as a recording, translate it into text, and stored the text data on the user's local device. A completed functional application will showcase at the end of the term semester. The Short-Term memory assistant Mobile app will fulfill all the high-level requirements. Finally, all the documentation such as Software Requirement Specification (SRS), technical design, software test plan, programmer guide, deployment operation guide will generate for the customers to view and print as necessary.

b. Team Charlie is a group of graduate students who perform the effort to complete Short-Term memory assistant mobile application and attend the Software Engineering Project course to complete the Master of Science in Information Technology with Software Engineering Specialization, Capstone Project.

2. Project assignment

For this project assignment, the project objective and goal are outlined in detail in Table 1. The project will utilize the SMART tool for Specific, Measurable, Achievable, Realistic, and Time. The table has separated the meaning of SMART that might make it easy to illustrate to the reader

to understand its matrix and system. Each letter indicated the word SMART wrote to the left columns, and each row to the right of the corresponding column express the meaning of the first letter. For example, Specific will aim to accomplish the project showcasing relevant questions such as common words who, what, why, where, and how. Measurable will aim to track project goals and range the outcome of the short-term memory assistant mobile application. Achievable will strive to satisfy the objective of how the project successful. This term can explain how the team member achieves the objective by utilizing skills and knowledge to reach the end goal. Realistic meaning that the goals and objectives are possible and that objectives outcome might reduce the overall project's overall plan. Time is the most viable to address the solution within a deadline/milestone (Project-Managment.com, 2019). Overall, the SMART tool is similar to the project management perspective in terms of sets a guideline to achieve the following objective: scope, budget, and time.

Table 1- SMART Goals and Objectives

Specific	Currently, the Unified Model Language (UML) illustrated minimal requirements of the Mnemosyne mobile application. It stated the app should do the following things: Speech to text converter, listen to the voice of user only, recognize distinct phrases, save the text in the device and retain for a week, search in the texts by keywords, format texts in the app (zoom, bold, increase font size), and training videos for the users. The short-term memory assistant application will deploy to the Android platform on the Google Play Store. The IOS devices are limited due to their constraints.

Measurable	Team Charlie will implement Hybrid methodologies to develop the mobile application. The method that will be using is Agile methodology combine with the Scrum framework. At the end of every Sprint/weekly meeting, a prototype/progress will satisfy the high-level requirement.
Achievable	After evaluating the PM team lead and team member volunteer for their position, the team has concluded each task and role for the team. Team Charlie consists of three Software Developers (one team lead developer), one Tester, one business analyst, and a Project Manager. By understanding each role; each team members are working simultaneously to incorporate solution efficiently. For example, a Software developer creates a functional feature requirement. The tester in the team should test that feature and ensure it is working correctly before moving on to the next set of requirements. Likewise, PM outline tasks, schedule, and planning to lead the team stay on schedule. Business analysts try to work with everyone, including members and customers, to make the project progress smooth.
Realistic	Dr. Mir Assadullah has initiated the project, and PM came up with a specific requirement. The project will move forward, and each team will come up with different unique design and solution that aim to suffice the need. The short-memory assistant mobile application will offer to public service professionals, disabled people (dementia to Alzheimer) upon

	completing the Software Engineering Project courses at the end of the semester.
Time	The Charlie team PM has set a milestone for each phase. It breaks down into three phases: planning phase, design & engineering phases, and execution phase. In the meantime, the team using hybrid methodologies, a prototype, and progress will be showcased to internal team members each week in the meeting. Its purpose is to make sure it good before delivering the final product to the customer. The mobile application will launch after the eleven weeks of the semester. Upon completing the mobile application during the execution phase, the system will launch Android free to all users through the Google Play Store.

The Charlie Team committed to deliver a mobile application quality product within budget and meet the deadline. We will not waste time developing more functional features if the approval general requirement is out of scope.

2.1 Project Scope

The Project Plan's scope for the Charlie Team is the Planning Phase, Design & Engineering Phase, and Execution Phase. The planning phase is where the vision statement, business need statement, object & goal, gathered requirement, a Project Management and Business Analyst (BA) Statement of Work (SOW) created, and communication plan within the team. The design & engineering phase is where software developers and tester write code to make such prototypes

that fulfill the requirement. Finally, the execution phase is to finalize a completion product, deploy an app, and deliver to the customer. The PM created a project schedule that outlines each stage's breakdown level using the MS office. For specific things need to be done, customer can refer to Appendix A for visualization.

This project's scope mainly concentrates on developing artificial intelligence (AI) short-term memory assistant mobile applications to help disabled people. The users can be doctors, nurses, and other disabled people. Besides, the mobile application platform will also base on Android Studio, which uses Flutter on Google Graphical User Interface (GUI) toolkits. The software developer will use Dart as a programing language as it is compatible with to Flutter user interface (UI). The final products of the app will deploy to IOS android, google play.

Below is a high-level requirement that has been decomposed:

The application shall provide the following means to activate recording by tapping on the app (or toggle in) and immediate voice recognition. The application shall provide a trigger to pause, resume and end the voice recording.

The app shall track the user's (person with a disability) communication with other parties. The application should tune in to listen to only the voice of the user. The application should ignore everything except what the user speaks, and for that reason, the application should bypass asking everyone permission to record.

The application shall provide the means for the user to train the app on its voice, and by so making the application should recognize distinct phrases and sentences that he or she uses while speaking to him or herself or with others. The application shall learn the phrases that the user

wants to use when talking to someone while trying to save crucial spoken text and the phrases that the user wants to use speaking to him or herself trying to retrieve the noted information.

After recording the user's speech, the application shall provide the ability to save the speech from converting texts to local device storage. The application shall not save any voice recording but will retain speech to text recognition notes for one week in duration.

The application shall provide the ability to search through the saved speech to text notes via text field and voice command. The application shall retrieve all results related to the search command.

The application shall provide a user interface that incorporates the following device features: Bold text, Display zoom, the ability to increase the text size to ensure a flexible environment for the user to customize.

The application shall provide training videos within the app to guide the user regarding its various features and functionalities.

Work within the scoping plan from the PM is following:

- The methodologies will be hybrid that is Agile and Scrum.
- The Flutter and Dart will be download for free from the opensource / internet.
- The outline detail to track progress on the team and assign team member tasks will show in the MS project office and Github platform.
- The code will share on Github by branches so the team member can easily access the code.
- The code will merge into main branches, and DevSecOps (DSO) will compile for submission.

- Testing will perform simultaneously during development to minimal achievable requirements.

2.2 Project Assumptions

The project assumptions described as following:

- The scope requirement will not adjust unless the Project Management, Stakeholder, and Dr. Mir Assadullah, approve a change. The scope is open-ended since the solution might not suffice the requirement due to limited technological progress.
- Team Charlie will aim to suffice the original outline requirement, which depends on software developers and researcher approach to find the best solution.
- The mobile application will be available for Android users through the Google Play Store for free.
- PM will be assigned team members task to perform based on their role.
- The project's health will monitor by the PM and BA weekly to make sure it on schedule.
- Milestone deliverables date will not change.
- Team Charlie applies hybrid methodologies using Agile and Scrum to create a plan for the project's short timeline. The team will have six weeks sprint for the development and documentation on their founding.
- PM creating logged status on Github to tracking progress and issues for each member task.
- There will be a potential data breach on the mobile application since it is open-source, and it is not guaranteed user protection to use the app.

- There will be no warranty for this app since the application might not maintain or continue development by UMGC software students. The University of Maryland is not responsible for anything when the app establishes.

2.3 Product Approval/Acceptance Criteria

The outcome of this document is to outline the project Milestone in set deliverable dates. There will be fours Milestone that due in this project. The following people: Dr. Mir Assadullah will grade each deliverable Milestone. Milestone 1, which included the Project Plan and Software Requirements Specifications, will be changed if further development does not suffice the professor's requirement or approval/acceptance feedback.

Change Management Plan contain in this document and leverage more detail in the future if receive feedback from the program manager or stakeholder. One of the changes in the process during development might be the architecture of the code. If the change is not made, the lead software developer on the team should be aware and mitigate the risk. Lead software developers should be in charge of the technical issues and find the best way forward of the deliverable. Another level of change is involving in impact functional requirements. The PM will affect these issues and discuss across other PM team and lead PM development to find the solution. If the solution is not resolvable, the external stakeholders' team (e.g., Professor Mir) will decide to reconcile the impact of this change. There will be an additional update change in the MS offices and GitHub management task process to track the progress of an approval/acceptance of the previous changes.

2.4 Project Exclusion

This section lists what is out of scope for the project.

- There will be no artificial intelligent model to develop in this document.

- Team Charlie might not spend any money to develop this app.
- The submission to deploy the app into the Google store is unknown due to partnering with an outside third party.

2.5 Acronyms and Abbreviations

Acronyms and Abbreviations	Definitions
AI	Artificial Intelligence
PM	Project Manager
BA	Business Analyst
DSO	DevSecOps
LEO	UMGC online platform class
SMART	A SMART tool stands for Specific, Measurable, Achievable, Relevant/Realistic, and Time-bound. It is used to define project goals and objectives.
UMGC	University Maryland Global Campus
GUI / UI	Graphical User Interface / User Interface
MS office	Microsoft Office Software
QA	Quality Assurance
WBS	Words Break Down Structure
UAT	User Acceptance Testing
MSTP	Mnemosyne Software Test Plan

JVM	Java Virtual Machine
API	Application Program Interface
SDLC	Software development life cycle
ASAP	As Soon As Possible

2.6 Referenced Documents

Table 2 below shows documents use as a reference for this Project Plan. The Title column shows the document's title, and the Reference column corresponds to the appropriate title where the paper is the reference. For example, the reference lists website locations and the name of the author of the document.

Table 2 - Referenced Documents

Title	Reference
Kick-Off Meeting, SWEN 670, Software Engineering Project, Course Homepage	https://learn.umgc.edu/d2l/home/545048
How to Write SMART Project Management Goals.	Project-Managment.com. (2019, November 14). *How to Write SMART Project Management Goals*. Retrieved from Project Management: https://project-management.com/what-are-smart-objectives-and-how-to-write-them/

ODESSA mobile technology project Plan ODESSA mobile technology project	Whitson, D. (2003, July 10). ODESSA mobile technology project Plan ODESSA mobile technology project. Retrieved June 06, 2021, from https://www.academia.edu/33300250/ODESSA_MOBILE_TECHNOLOGY_PROJECT_Project_Plan_Odessa_Mobile_Technology_Project.
Project Management Handbook	Project Management Handbook. Through the LEO Course Introduction Page. https://learn.umgc.edu/d2l/le/content/545048/Home

3.0 Risk Analysis

This initial risk assessment attempts to identify, characterize, prioritize, and document a mitigation approach relative to those risks that can be identified before starting the project. The risk assessment will be continuously monitored and updated throughout the life of the project. During weekly sprint meetings, the project team will dedicate time to identify new risks and discuss mitigation strategies. Project leadership will assess the impact of the risk, the project's ability to accept the risk, and the feasibility of mitigating the risk before mitigation approaches can be approved. Weekly assessment will be documented in the status report to be opened for amendment by the project manager. The Project Manager will convey amendments and recommended contingencies to the project team, or more frequently, as conditions may warrant (Whitson, 2003). For accurate and effective risk analysis, the risk matrix, analysis, and risk register will be developed to assist the projects in understanding the problem they are facing and approach it in a meaningful way.

3.1 Risk Matrix

The purpose of the risk matrix is to increase risk visibility and assist the team in making good decisions. Below is the risk matrix to determine the level of risk by considering the category of probability or likelihood against the category of consequence severity. The matrix forms a box where the seriousness of risk lines the left side, and the likelihood of the risk lines the bottom of the box.

	Likelihood				
Severity	Unlikely	Rare	Possible	Likely	Certain
5 Fatal	5	10	15	20	25
4 Major	4	8	12	16	20
3 Moderate	3	6	9	12	15
2 Minor	2	4	6	8	10
1 Insignificant	1	2	3	4	5

3.2 Analysis

The purpose of analysis to help the project team understand the problems facing the project and enable them to approach the issue in a meaningful way. The Analysis table below is created based on the risk matrix. Severity is assigned to risk along with the probability of the risk occurring. Each risk has an owner responsible for applying the mitigation strategy specified depending on the likelihood and severity of the risk.

Risk #	Risk	Trigger Event/Severity	Responsible	Likelihood/ Probability	Mitigation
R1	The physical location of the team prevents effective management	4: Team is dispersed among several sites	Project Manager	Likely	Use of Microsoft teams and other comprehensive communication plans.
R2	Project Team Availability	3: A distributed team makes availability questionable	Project Manager	Possible	Continuous review of project momentum by all levels. Connect with members at regular intervals to identify and prevent any impacts caused by unavailability. Provide coaching
R3	Absence of Commitment Level/Attitude of Management	1: Understands value & supports project	Project Manager	Unlikely	Frequently seek feedback to ensure continued support
R4	Project Scope Creep	1: Scope generally defined, subject to revision	Project Sponsor, PM & BA	Unlikely	Consider mentorship with senior-level members.

R5	Timeline Estimates Unrealistic	3: Timeline assumes no derailment	Project Manager	Possible	Timeline reviewed weekly by the Project Manager to prevent undetected timeline departures.
R6	General Testing Problems	5: Inadequate software testing	Project Manager	Certainty	Ensure the testing team performs quality checks and test software functions before the product goes live.

3.3 Risk Register

The purpose of the risk register is to help the project team stay on top of potential issues that can hinder intended outcomes. The projected track any obstacle to the success by, the risk register is created below. The risk trigger ranking used in the analysis in section 3.2 assigned to the potential risks will determine the severity of the issue. The risk ID used in section 3.2 will determine the actions and steps to mitigate the risk.

Ranking	Potential Risk	Risk ID
1	Increased workload creates conflict with team goals	R5
3	Scheduled activities not completed on time	R1 & R2
3	Search command not responding	R6

4	Audio recording limitation with Flutter	R6

4. Organizational Overview

Team Charlie will be following the Agile methodology while implementing the Scrum framework. This project will include sprints, weekly meetings, and daily check-ins over Microsoft Teams. Additional meetings will be held before and after each Milestone, both within the team and with stakeholders.

The project will consist of three major phases: the Initiation Phase, Design & Engineering Phase, and the Execution Phase. The Initiation Phase covers Milestone 1, which includes the initial Project Plan document (this document) and the SRS document. During the Initiation Phase, the team will create a plan, prepare for adherence to Agile principles, outline requirements, and a rough sequence of Sprints that the team expects to follow. The Design & Engineering will be actual development stage. The team will focus on development and document the progress within high-level stress in six weeks. The Execution Phase consists of building on top of several development sprints culminating with the finished application. Each Sprint will have planning, work, a review, and a retrospective.

4.1 Communication

The majority of communication between team members, stakeholders, instructors, and advisors will be done through email and Microsoft Teams. Teams allow regular text-based communication, file sharing, scheduled meetings, and a shared calendar for the team to stay organized.

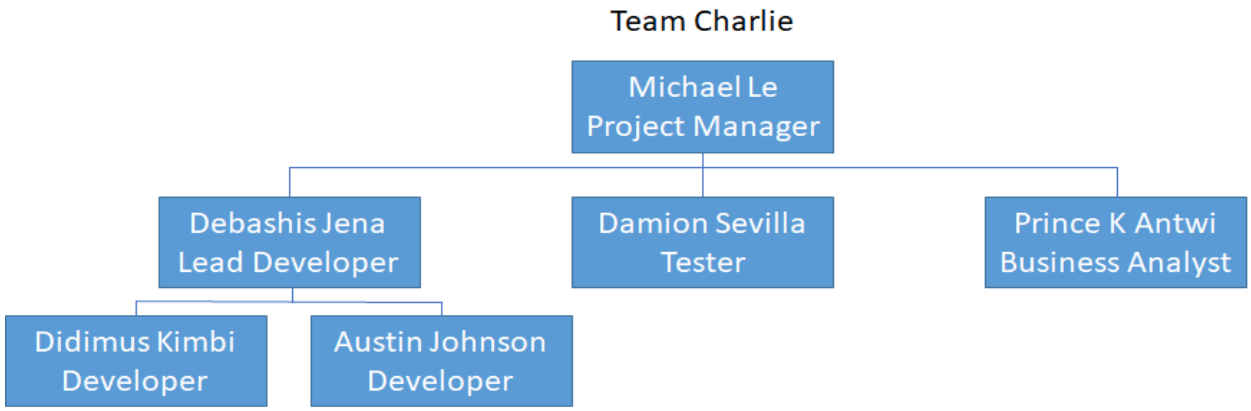
Team meetings will happen weekly on Thursdays. All members will discuss current progress and what everyone should be working on for that week. In addition to weekly meetings, team members will stay in constant communication through text chats to ensure that everyone is on the same page.

Stakeholders and advisors can contact team members through this same group chat. Additionally, if necessary, voice and video meetings can be held, for example, at the end of each Milestone. The team's manager will often meet with other managers and the overall project manager to discuss issues that apply to all groups.

PM will establish and breaking down the task for each team member to perform. GitHub project management and MS office will be two tools to create a project plan. PM will utilize GitHub to have the outline detail for tracking issues that show the current progress. Also, the lead developer and a team member will use Github for code sharing within the branch. It will merge to the main branch that DSO can take over to utilize the whole application.

4.2 Team Members

Team Charlie has six members, including a project manager, a test engineer, a business analyst, and three developers. These roles may cross over from time to time as the project needs to develop. Outside of Team Charlie are two other development teams, a DevSecOps team and an overall project manager.



4.3 Roles and Responsibilities

Role	Member(s)	Responsibilities
Project Manager	Michael Le	The Project Manager monitors the project's overall progress while planning meetings, keeping tabs on everyone's work, ensuring that the project is working towards fulfilling requirements, and acting as a middleman between the team and the overall project manager.
Lead Developer	Debashis Jena	The Lead Developer works closely with other Developers to implement the project and acts as a point of contact between the development team and other development teams.

Developer	Austin Johnson Didimus Kimbi	The Developers code, design, develop and debug the actual application.
Tester	Damion Sevilla	The tester familiarizes themselves with the workings of the project, and its requirements test the application for adherence to those requirements and finds any potential bugs.
Business Analyst	Prince K Antwi	The Business Analyst is responsible for determining high-level requirements for the project based on communications with stakeholders. They are also responsible for ensuring that these requirements continue to satisfy the business needs of stakeholders as the project develops.

4.4 Responsibility Assigned Matrix

Responsible

Do the work. Completes the objective or makes the decision.

Accountable

The owner of the work. They must approve of assigned tasks.

Consulted

Gives input on work before it is done.

Informed

Kept up to date on the work but not formally consulted.

	Project Manager	Lead Developer	Developers	Tester	Business Analyst
Maintain Scrum Practices	R/A/C	C	I	I	I
Provide high-level goals for the project	A/C	I	I	I	R
Manage requirements	A	R	R	C	I
Implement requirements	I	R/A/C	R	C	I
Define acceptance criteria	I	C	C	R	C

Write acceptance tests	I	C	C	R	C
Ensure quality of the application	R/A/C	R	R	R	R
Manage Release	R/A/C	R/A	R	C	C

4.5 Project Tools

Tool	Function
LEO	While most of the work happens outside of LEO, the LEO ecosystem is still essential for submitting milestones, personal work descriptions, and peer reviews
Google Play Store	The mobile application store for android devices
App Store	Mobile application store for iOS devices
Flutter	A toolkit for developing applications for both iOS and Android with the same codebase

Dart	The programming language used with Flutter to create the application
GitHub	A GitHub repository will use to organize the code for the project

4.6 Project Document Storage

Most documents will store within Microsoft Teams, and the code will keep in a Github repository. This concept should ensure that all team members have constant access to necessary materials and deliverables regardless of time or location.

4.7 Deliverables

Milestone 1 will consist of the Project Plan and Software Requirements Specification documents.

Milestone 2 will consist of a Technical Design Document and a Software Test Plan.

Milestone 3 will consist of a Programmer Guide and a Deployment and Operations Guide.

Milestone 4 will consist of a User Guide, a Test Report, and the final version of the actual application.

4.8 Initiation Phase

The Initiation Phase began with a Kick-Off meeting between the entire class, the instructor, and advisors. During this meeting, the project's overall goals were established, and an explanation of critical roles and teams was given. Shortly after, the class members formed their groups and were assigned roles based on previous experience or chose parts for themselves.

Once Team Charlie was established, the team had a meeting to decides how to proceed with Milestone 1, finalized roles, and planned weekly sessions to continue throughout the project.

Milestone 1 is wrapping up with a Project Plan document, and a Software Requirements Specification document was created. Further documentation will be made throughout future Milestones within the Design&Engineering Phase, Execution Phase, but creating these two initial documents will allow the team to begin the next phase, which is Design & Engineering phase and Execution Phase.

4.9 Design & Engineering and Execution Phase

The Execution Phase will last for the remainder of the project until the application is finished and satisfactory. This phase will consist of some overall sprint planning followed by several development sprints.

Before sprints begin, the team will meet to discuss high-level project goals and use those goals to create a Sprint Backlog. The backlog will use throughout the project to choose the subject of focus for each Sprint and to ensure that the work done during each Sprint continues to satisfy requirements. During this initial planning, the Technical Design document and Software Test Plan documents were created.

Each Sprint will focus on implementing and testing one feature of the Sprint Backlog. The Sprint will begin with an initial meeting to discuss the goals of this specific Sprint and assign responsibilities. Throughout the Sprint, weekly sessions will be held through Microsoft Teams, while daily check-ins will conduct with Microsoft Teams text chat. The meetings will discuss progress, discuss what everyone is currently working on, and discuss the following week/day goals. Testing will conduct near the end of a sprint where acceptance tests are created and

applied to ensure the feature implement correctly and accurately satisfies requirements. At the end of a sprint, the team will meet to discuss how the Sprint went, whether or not any changes need to make to the backlog or future sprints, and the next Sprint planned.

During these development sprints, further documentation creates to meet Milestones 2 through 4. As more features are developed and finalized, information about those features can add to the documentation.

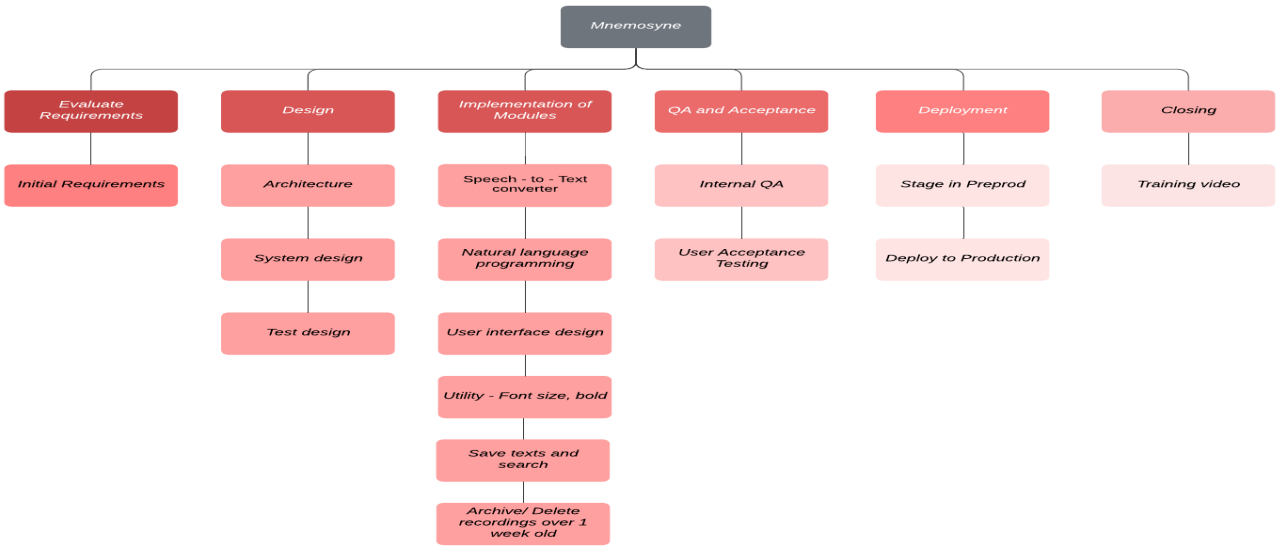
The project will conclude with some final testing and a meeting to ensure everything is satisfactory before the application finalize.

5 Cost and Schedule Overview

The final section of the project plan provides an overview of the costs and the timeline for the entire project.

5.1 Work Breakdown Structure

Below Work breakdown structure (WBS) diagram of the project has been created considering each use case to be implemented.



5.2 Cost Overview

The cost for this project is estimated based on the bottom-up approach. The above WBS diagram enables the estimates to be calculated for each of the detailed level modules/components. The table below shows a timeline and the cost of the whole project. Each of the tasks in the WBS diagram will span through the entire timeline, as mentioned in the cost overview table below. For example, the implementation phase will span through milestones 2, 3, and 4. Each module's "Implementation" phase is estimated separately and summed to calculate the total by each Milestone. The price is estimated by the effort for analysis, development, QA, and project management. The below table illustrated the team member role on the left. The Milestone is breaking down into fours, and each Milestone indicated the dollar amount with the number of hours each member in the team is working.

	Milestone 1 Total Cost		Milestone 2 Total Cost		Milestone 3 Total Cost		Milestone 4 Total Cost
Week 1	$1,719	Week 4	$2,865	Week 7	$1,910	Week 10	$2,228
Week 2	$1,719	Week 5	$2,228	Week 8	$2,865	Week 11	$1,910
Week 3	$1,719	Week 6	$1,910	Week 9	$1,910	Week 12	$1,910
Total	$5,157		$7,003		$6,685		$6,048

5.3 Overview of Costs

The below estimation is calculated with a man-hour unit.

Phase	Expected starting date	Expected date of completion	Total cost estimate
Milestone 1	5/19/2021	6/11/2021	$5,157.00
Milestone 2	6/12/2021	07/03/2021	$7,003.00
Milestone 3	07/04/2021	7/24/2021	$6,685.00
Milestone 4	7/25/2021	8/5/2021	$6,048.00
Initiation	5/19/2021	6/11/2021	
Design & Engineering and Execution	6/12/2021	8/5/2021	
		Total:	$24,893.00

5.4 Schedule Overview

The project schedule has been set based on the project sponsor's milestones and discussed with the stakeholders. As mentioned in the WBS, the plan starts with the gathering requirements and then goes through analysis, development, QA, UAT, and deployment. Each of the milestones has deliverables and is broken down by the phases of the project execution.

6 Software Test Plan

6.1 Test Plan Identifier

Mnemosyne Software Test Plan (MSTP)

6.2 Introduction

The mobile team will design and develop a short-term memory assistant mobile application using the Flutter and Android studio platform. The application will include the following use cases:

- Record Speech
- Saved and Retain Text
- Voice Recognition Training
- Record Speech of User Only
- Visual Options
- Searching Text
- Training Videos
- Send notification to users

The software test will cover all functionalities, including the User Interface, to verify and validate that our mobile application works well and accomplishes the tasks given.

6.2.1 Background

The Mnemosyne Software Test Plan (MSTP) was created to test the UMGC Mnemosyne mobile application to ensure that the software system meets the software requirements specification and the user's and stakeholder's expectations. The MSTP serves as the master test plan for the mobile application. These tests will perform throughout the project's life cycle. Also, the test

results will report to the Project Manager, the developers, and the stakeholders. Negative test results and bugs will fix to improve the quality and performance of the software system.

6.2.2 Objective of the Test Plan

Software testing is a method to check whether the software product matches the standard requirements and specifications requested and ensure the system's functionality. It involves the execution of software components using manual or automated tools to evaluate the various properties of the system. The purpose of software testing is to identify errors, gaps, or missing requirements in the project. This software test plan aims to communicate a detailed plan for running development tests to the development team and stakeholders.

6.2.3 Objective of MSTP

The MSTP objective for the mobile application will ensure that the stakeholder requirements are met, and the final product will deliver to public service professionals with functional features. The MSTP will follow the Software Requirements Specification when approaching the software testing. Specific attention will place on the Use Cases section of the SRS for the MSTP.

6.3 Reference Documents

- Scope Statement
- Test Plan Document Milestone 2_Team Charlie Summer 2021

6.4 Test Items

Before launching the Mnemosyne mobile application, seven Use Cases from the SRS will test by the tester and development team to validate and verify the mobile application.

Table 2- test item based on Use Case: Speech to Text Conversion, the user's speech converted to text.

Table 2- Speech to Text Conversion

Items to be Tested	**Version Number**	**Test Action**	**Expected Results**
Speech to Text Conversion	1.0	1. The actor speaks into the microphone	1. The words spoken are converted into text.

Table 3- test item is base on Use Case: Saved Text Files, converted speech is saved to a file

Table 3- Saved Text Files

Items to be Tested	**Version Number**	**Test Action**	**Expected Results**
Saved Text Files	1.0	1. The actor speaks into the microphone.	1. The words spoken are converted to text. 2. The converted text is then saved to a file to view later.

Table 4- test item is base on Use Case: Recognize Distinct Phrases, the user's unique speech patterns are recognized

Table 4- Recognize Distinct Phrases

Items to be Tested	**Version Number**	**Test Action**	**Expected Results**
Recognize Distinct Phrases	1.0	1. The actor speaks unique lines repeatedly.	1. The speech-to-text converter will accurately recognize and convert the words without spelling errors.

Table 5- test item is base on Use Case: The mic will recognize user Voice Recognition, only the user's voice

Table 5- User Voice Recognition

Items to be Tested	**Version Number**	**Test Action**	**Expected Results**

User Voice Recognition	1.0	1. Several actors talk while the microphone is active.	1. The primary user's voice is the only voice that is converted into text.

Table 6- test item is base on Use Case: Settings/Options button, the menu for editing app settings

Table 6- Settings/Options button

Items to be Tested	**Version Number**	**Test Action**	**Expected Results**
Settings/Options Button	1.0	1. The actor will click on the settings button.	1. The settings/Options menu will display for the user to customize.

Table 7- test item is base on Use Case: search bar, the user can search for relevant text files with a keyword

Table 7- Search Bar

Items to be Tested	Version Number	Test Action	Expected Results
Search Bar	1.0	1. The actor will enter a word into the search bar and start the search.	1. The search bar will pull text files with text related to the keyword searched.

Table 8- test item is base on Use Case: Training videos, media for teaching the user how to navigate and use the app

Table 8- Training Videos

Items to be Tested	Version Number	Test Action	Expected Results
Training Videos	1.0	1. The actor will click on the training videos.	1. The training videos will play for the user.

6.5 Features to be Tested

Features that will test for the Mnemosyne mobile application listed in the bullet points below:

- Record Speech

- Saved and Retain Text
- Voice Recognition Training
- Record Speech of User Only
- Visual Options
- Searching Text
- Training Videos
- Send notification to users

6.6 Features not to be Tested

Features not included in the requirements will be out of scope and not tested for the Mnemosyne mobile application. Any requirements not listed in Section 6.5 will not be tested.

6.7 Approach

An approach to testing will be broken down into five steps. Those steps are listed below.

- Develop test
- Prepare the test
- Run the test
- Review the test
- Report the test.

6.7.1 Develop Tests

The test tools below will be used to develop tests.

- Firebase Test Lab v28.1.0
- Flutter test function v2.2.0
- Visual Studio v1.56.2

Flutter test can be used to:

- Perform Unit tests to validate operations.
- Perform Widget tests for the home and favorite pages.
- Perform UI and performance tests for the entire app using Flutter Driver.

Android Studio was designed to make testing simple. We can set up a JUnit test that runs on the local JVM or an instrumented test that runs on a device with little effort. Test capabilities can be expanded upon by integrating test frameworks to test Android API calls in local unit tests.

During System Integration Testing, the mobile team will perform the following test types:

- Acceptance Testing: Performed in System Integration Phase
- Unit Testing: Performed during the System Development Phase
- Functional Testing: Performed during Use Case Development Cycle
- Regression Testing: Performed throughout the SDLC

Test Development includes the following activities:

- Reviewing and analyzing the Software Requirements and the Project Plan

- Development of scenarios and required testing techniques.
- Define acceptance criteria to meet the accepted standard
- Construction of test case which defined by the use case input/output
- Writing and creation of test scripts
- Running tests and documenting the results of each test
- Review of all testing documentation

The team will conduct the following studies:

- Test plan review
- Test case review
- Test progress and milestone review
- Post-test review

6.7.2 Prepare to Test

A preparation to test the Mnemosyne mobile application will have the following steps:

- Test environment preparation: hardware, testing tools, and staffing will prepare to initiate testing.
- Testing documentation preparation: test documentation will create for setting test guidelines and reports.

6.7.3 Run Tests

We can use the following processes to run the test:

- Test scripts and test cases will generate in the appropriate assigned development section of the mobile application.
- Tracking tests are done through the testing tools listed in Section 6.7.1

6.7.4 Review Test Results

Evaluation will be done for each component throughout the testing process, and all results will be documented. Deviations will be addressed, and possible ways to correct these deviations will be discussed for future releases. Test results will cross-examine with the system requirements. The developer for the assigned development task will work with the tester to review the results.

Defects will have ratings determined by severity and priority, as shown in the table below.

Defect Priority Levels	Defect Severity Levels
P1 Critical	S1 Critical
P2 High	S2 High
P3 Medium	S3 Medium
P4 Low	S4 Low

6.7.5 Test Results and Defects

Severity defines the level of impact a defect can create on the application or system.

The defects are classified as follows:

Critical (S1)

A defect that prevents testing of the product/feature is critical. For any reason, if the application crashes or cannot continue the testing/debugging process, the defect could be classified under essential severity.

Major (S2)

Any prominent feature implemented that does not meet requirements and behaves differently than expected can be classified under Major severity. Any defect that could lead to data issues or wrong application behaviors can classify under Major Severity.

Minor/Moderate (S3)

Any feature that does not meet the requirements or uses cases and does not significantly affect the application can classify under Minor severity.

A moderate defect occurs when the product or application does not meet specific criteria; however, the functionality is not impacted.

Low (S4)

Any cosmetic defects like spelling errors, alignment mistakes, font casing, etc. that be classified as low severity.

A low severity bug occurs when there is almost no impact on functionality but is still a valid defect that should be corrected.

The priority of the defects can be classified as follows:

Critical (P1)

This concept means must be fixed immediately, ASAP. It generally occurs when a complete functionality is blocked, and testing cannot continue until its fixed. Any defect that needs immediate attention and impacts the testing process will be classified as Critical.

High (P2)

Once critical defects have been fixed, a deficiency of this priority is next in line to be fixed for any test activity to match the expected result. Usually, when the feature is not usable due to an error in the program, new code needs to be written or an issue with the IDE. These issues must resolve before the release is made but after all critical errors have been fixed.

Medium (P3)

A defect of this priority must be in contention to be fixed as it could also deal with functionality issues that are not as per expectation. Some cosmetic errors such as displaying the correct error message during failure can be a high-priority defect. These defects should resolve after all severe issues have been resolved.

Low (P4)

A defect with low priority indicates an issue; however, it does not need to fix to meet the end criteria. It does, however, need to resolve before the GA is done. Typically, some typos or cosmetic errors, like mentioned before, can be categorized here. Sometimes low priority defects are opened to discuss enhancements in the existing design or request to implement a minor feature to enhance the user's experience. These defects can be resolved at any time in the future and do not require immediate attention.

6.8 Item Pass/Fail Criteria

The tester will create test cases for each requirement listed in Section 6.5 Feature to be Tested. Test cases will include user inputs and the appropriate output. Besides, the test will pass based on the test case outcome and the predefined result. The tester can determine if a test has failed by the test output will be reviewed, and if it does not match the predefined outcome, it will document as failed.

6.9 Test Plan Matrix

A matrix will be put in place for the mobile application (Mnemosyne app) to record tests, document test coverage, what tests were done, where the defect might be, and assure requirements were met.

Table 9 shows the Test Plan Matrix, with each column labeled. The first column will list Test Cases Planned, the second column Test Cases Executed, the third column Test Cases Passed, and the last column Test Cases Failed. What has been entered into Table 16 is an example of what kind of data will input.

Table 9 – Test Plan Matrix

Test Cases Planned	**Test Cases Executed**	**Test Cases Passed**	**Test Cases Failed**
Key phrase registration	1. The actor taps and holds on the speech button on the main screen.	50%	50%

6.10 Test Deliverables

The mobile team will deliver to the tester the outcome of a test:

- Testing Acceptance Plan, which describes the overall activities of the mobile application tests.
- The results of each test case
- Test log
- Incident report
- Incident report log update
- Test Summary Report

6.11 Testing Tasks

Testing tasks will be adapted in the following manner:

1. Software Test Plan
2. Communication with the Lead Developer of the assigned requirements for the test deliverables following corresponding documentation
3. Confirm that proper testing tools are helpful and developers are trained to know how to use them
4. Run tests

5. Tests are document

6.12 Environmental Needs

The following tools will need to process the testing that will be done for the mobile application:

- Flutter test
- Android Studio
- Firebase Test Lab

6.13 Responsibilities

The Lead Tester will be responsible for evaluating the requirements of the mobile application attributes and capabilities, then organizing and managing the testing process to verify that visibility, traceability, and control of tests met to deliver a working mobile application.

For the testing to be successful, the Lead Tester and Lead Developer will collaborate with other team channels to ensure met requirements. Table 10 shows the assigned test tasks and which team member will be responsible for the specified task.

Table 10 - Test Responsibilities

Task	Assigned To
Software Test Plan	Damion Sevilla
Requirements Documentation	Prince k Antwi Aboagye
Test Creation	Damion Sevilla, Austin Johnson, Debashis Jena, Didimus Kimbi

Run Test	Damion Sevilla, Austin Johnson, Debashis Jena, Didimus Kimbi
Summary of Test	Damion Sevilla

6.14 Staffing and Training Needs

Testing will perform by the tester, developers, and lead developer of the mobile team. The Project Manager and Lead developer will work together to ensure the requirements are met in conjunction with the test results.

The following bullet points will address that the tests run successfully:

- The mobile team will meet regularly during sprints to ensure testing tools are used and that the tools work.
- Through MS Teams' weekly meeting screen sharing, the team will be trained on the testing tool. A video of the session will post on the MS Team mobile channel to reference it when needed.
- The mobile team will provide the Software Test Plan to review.
- Lead Tester, Lead Developer will work together to communicate tests to developers.

6.15 Schedule

Sprints will be held weekly to include Product Increment (Test Cases and Test Results) held throughout the Sprint regarding the schedule. Sprint 1 will start June 11, 2021, and the last Sprint

6 will end July 22, 2021. Regarding the Appendix A of the Project Plan, the testing and Sprint schedule can be seen in more detail.

Table 11 shows the Staffing and Training Need Table, with the first column showing the Task Name, second column Duration, third column Start, fourth column Finish, and fifth column Resource Name.

Table 11 - Staffing and Training Needs Table

Task Name	**Duration**	**Start**	**Finish**	**Resource Name**
Software Test Plan	3 weeks	F 6/11/21	TH 7/01/21	Damion Sevilla
Performing Test Cases	3 weeks	F 7/02/2021	TH 7/22/2021	Damion Sevilla, Austin Johnson, Debashis Jena, Didimus Kimbi
Test Report	10 days	S 7/24/2021	F 8/06/2021	Damion Sevilla, Prince

6.16 Risks and Contingencies

Each risk and its cause should be identified, the likelihood of it happening should be addressed. A risk management plan should be created. The risk management process needs a schedule to determine how often and when risk activities should occur throughout the project. Qualitative risk analysis qualifies the risks that have been identified in the project. Not all risks are worth

responding to, but some demand attention. Qualitative analysis is a subjective approach to organizing and prioritizing risks. Identified risks can rate according to probability and potential impact. See table 12 for an example:

Table 12 – Risks and Contingencies

Risks	**Probability**	**Impact**	**Risk Score**
The mobile app project cannot complete on time.	Low	High	Moderate
Use cases development completed late	Moderate	Moderate	High
Testing delay due to Use case late deliverables.	Low	Low	Moderate
Project integration with Dialog Flow encounters a technical problem	Moderate	High	High

The high score risks require more attention to be mitigated and have more priority than the others.

6.17 Test Revision History

A table was created to track test changes. The test changes log will be available in MS Teams mobile channel file folder labeled “Test Revision History Log.” Table 13 shows an example of the log.

Table 13 - Test Revision History Log

Test Name	Date	Description	Approved By
Key phrase registration	8/05/2021	Test user key phrase registration functionality	Damion Sevilla

Appendices:

Appendix A – Detail Timeline Word Breakdown Structure

Task Name	Duration	Start	Finish	Predecesso	Resource Names
First day of class meeting	1 hr	Wed 5/19/21	Wed 5/19/21		Michael
PM kick Off Meeting & Roy Gordon	1 hr	Sun 5/23/21	Sun 5/23/21		Michael
Jira Meeting with Malcom (Program manager)	1 hr	Mon 5/24/21	Mon 5/24/21		Michael
PM Second Meeting	1 hr	Tue 5/25/21	Tue 5/25/21		Michael
PM Meeting With DSO	1 hr	Fri 5/28/21	Fri 5/28/21		Michael
PM, BA Meeting Requirement	1 hr	Tue 6/1/21	Tue 6/1/21		Michael,Prince
PM, Liason Tech Lead, DSO Meeting	1 hr	Tue 6/1/21	Tue 6/1/21		Michael,Debashis
PM Third Meeting	1 hr	Fri 6/4/21	Fri 6/4/21		Michael
PM Fourth Meeting	1 day	Wed 6/9/21	Wed 6/9/21		Michael
Milestone 1	**3.6 wks**	**Wed 5/19/21**	**Fri 6/11/21**		
Project plan Document	8 days	Thu 5/27/21	Sat 6/5/21		All Team member
Software Requirment Specification	8 days	Thu 5/27/21	Sat 6/5/21		All Team member
Pencil down date	1 hr	Sat 6/5/21	Sat 6/5/21		All Team member
Review/Edit Milestone 1	2 days	Sun 6/6/21	Mon 6/7/21		Michael,Damion
End Milestone 1	0 days	Fri 6/11/21	Fri 6/11/21		Michael,Damion

Task Name	Duration	Start	Finish	Predecesso	Resource Names
Milestone 2	**3.2 wks**	**Sat 6/12/21**	**Fri 7/2/21**		
Technical Design Document	14 days	Sat 6/12/21	Wed 6/30/21		All Team member
Add Software Test Plan to Project Plan	14 days	Sat 6/12/21	Wed 6/30/21		All Team member
Research development	14 days	Sat 6/12/21	Wed 6/30/21		
Pencil down date	1 hr	Wed 6/30/21	Wed 6/30/21		All Team member
Updates from Milestone 1	2 days	Mon 6/28/21	Tue 6/29/21		Michael,Damion
End Milestone 2	0 days	Fri 7/2/21	Fri 7/2/21		Michael,Damion
Milestone 3	**3.2 wks**	**Sat 7/3/21**	**Fri 7/23/21**		
Programmer Guide	14 days	Sat 7/3/21	Wed 7/21/21		All Team member
Deployment and Operations Guide (Runbook)	14 days	Sat 7/3/21	Wed 7/21/21		All Team member
Research Development	14 days	Sat 7/3/21	Wed 7/21/21		All Team member
Pencil down date	1 hr	Wed 7/21/21	Wed 7/21/21		All Team member
Update from Milestone 2	2 days	Wed 7/21/21	Thu 7/22/21		Michael,Damion
End Milestone 3	0 days	Fri 7/23/21	Fri 7/23/21		Michael,Damion
Milestone 4	**2.2 wks**	**Sat 7/24/21**	**Fri 8/6/21**		
User Guide	9 days	Sat 7/24/21	Wed 8/4/21		All Team member
Test Report	9 days	Sat 7/24/21	Wed 8/4/21		All Team member
Final Research Development	9 days	Sat 7/24/21	Wed 8/4/21		All Team member
Pencil down date	1 hr	Wed 8/4/21	Wed 8/4/21		All Team member
Update from Milestone 3	2 days	Wed 8/4/21	Thu 8/5/21		Michael,Damion
End Milestone 4	0 days	Fri 8/6/21	Fri 8/6/21		Michael,Damion

Task Name	Duration	Start	Finish	Predecesso	Resource Names
Planning Phase (Milestone 1)	**3.6 wks**	**Wed 5/19/21**	**Fri 6/11/21**		
SRS and Project Plan Documentation	**18 days**	**Wed 5/19/21**	**Fri 6/11/21**		
Initial Meeting with team members	1 hr	Thu 5/27/21	Thu 5/27/21		All Team member
Roles assigned	1 day	Fri 5/28/21	Fri 5/28/21		Michael
Project/Requirements Gathered	6 days	Sun 5/23/21	Fri 5/28/21		PM Team
Scope of Work	6 days	Sun 5/23/21	Fri 5/28/21		PM Team
Communication Plan	1 day	Thu 5/27/21	Thu 5/27/21		PM Team
Project Plan	8 days	Thu 5/27/21	Sat 6/5/21		All Team member
Software Requirment Specification (SRS)	8 days	Thu 5/27/21	Sat 6/5/21		All Team member
Create Product Backlc	1 day	Sun 6/6/21	Sun 6/6/21		Michael
SRS and Project Plan Due	1 day	Fri 6/11/21	Fri 6/11/21		Michael

Task Name	Duration	Start	Finish	Predecesso	Resource Names
Design & Engineering Phase (Milestone 2 & 3)	**6 wks?**	**Fri 6/11/21**	**Thu 7/22/21**		
Sprint 1	**5 days**	**Fri 6/11/21**	**Thu 6/17/21**		
Sprint Planning	1 day	Sat 6/12/21	Sat 6/12/21		Michael
Sprint Monitoring	4 days	Sun 6/13/21	Wed 6/16/21		Prince
Scrum Meeting Weekly	1 day	Thu 6/17/21	Thu 6/17/21		All Team member
Create Test Plan	4 days	Sun 6/13/21	Wed 6/16/21		Damion
Sprint Review	2 days	Tue 6/15/21	Wed 6/16/21		Damion,Prince
Sprint End/Demo	0 days	Thu 6/17/21	Thu 6/17/21		Michael
Sprint 2	**5 days**	**Fri 6/18/21**	**Thu 6/24/21**		
Sprint Planning	1 day	Sat 6/19/21	Sat 6/19/21		Michael
Sprint Monitoring	4 days	Sun 6/20/21	Wed 6/23/21		Prince
Scrum Meeting Weekly	1 day	Thu 6/24/21	Thu 6/24/21		All Team member
Create Test Plan	4 days	Sun 6/20/21	Wed 6/23/21		Damion
Sprint Review	2 days	Tue 6/22/21	Wed 6/23/21		Damion,Prince
Sprint End/Demo	0 days	Thu 6/24/21	Thu 6/24/21		Michael

Task Name	Duration	Start	Finish	Predecesso	Resource Names
Sprint 3	**5 days?**	**Fri 6/25/21**	**Thu 7/1/21**		
Sprint Planning	1 day	Sat 6/26/21	Sat 6/26/21		Michael
Sprint Monitoring	4 days	Sun 6/27/21	Wed 6/30/21		Prince
Scrum Meeting Weekly	1 day	Thu 7/1/21	Thu 7/1/21		All Team member
Create Test Plan	4 days	Sun 6/27/21	Wed 6/30/21		Damion
Sprint Review	2 days	Tue 6/29/21	Wed 6/30/21		Damion,Prince
Internal documentation review	2 days	Wed 6/30/21	Thu 7/1/21		Michael
Sprint End/Demo	0 days	Thu 7/1/21	Thu 7/1/21		Michael
Technical Design Document & Add Software Test Plan to the Project Plan	0 days	Thu 7/1/21	Thu 7/1/21		Michael
Sprint 4	**5 days?**	**Fri 7/2/21**	**Thu 7/8/21**		
Sprint Planning	1 day	Sat 7/3/21	Sat 7/3/21		Michael
Sprint Monitoring	4 days	Sun 7/4/21	Wed 7/7/21		Prince
Scrum Meeting Weekly	1 day	Thu 7/8/21	Thu 7/8/21		All Team member
Product Increment (Test Cases and Test Results)	4 days	Sun 7/4/21	Wed 7/7/21		Damion
Performance Test Case	4 days	Sun 7/4/21	Wed 7/7/21		Damion,Austin,Debashis,Didimus
Sprint Review	2 days	Tue 7/6/21	Wed 7/7/21		Damion,Prince
Sprint End/Demo	0 days	Thu 7/8/21	Thu 7/8/21		Michael

Task Name	Duration	Start	Finish	Predecesso	Resource Names
Sprint 5	**5 days**	**Fri 7/9/21**	**Thu 7/15/21**		
Sprint Planning	1 day	Sat 7/10/21	Sat 7/10/21		Michael
Sprint Monitoring	4 days	Sun 7/11/21	Wed 7/14/21		Prince
Scrum Meeting Weekly	1 day	Thu 7/15/21	Thu 7/15/21		All Team member
Product Increment (Test Cases and Test Results)	4 days	Sun 7/11/21	Wed 7/14/21		Damion
Performance Test Case	4 days	Sun 7/11/21	Wed 7/14/21		Damion,Austin,Debashis,Didimus
Sprint Review	2 days	Tue 7/13/21	Wed 7/14/21		Damion,Prince
Sprint End/Demo	0 days	Thu 7/15/21	Thu 7/15/21		Michael
Sprint 6	**5 days**	**Fri 7/16/21**	**Thu 7/22/21**		
Sprint Planning	1 day	Sat 7/17/21	Sat 7/17/21		Michael
Sprint Monitoring	4 days	Sun 7/18/21	Wed 7/21/21		Prince
Scrum Meeting Weekly	1 day	Thu 7/22/21	Thu 7/22/21		All Team member
Product Increment (Test Cases and Test Results)	4 days	Sun 7/18/21	Wed 7/21/21		Damion
Performing Test Cases	4 days	Sun 7/18/21	Wed 7/21/21		Damion,Austin,Debashis,Didimus
Sprint Review	2 days	Tue 7/20/21	Wed 7/21/21		Damion,Prince
Internal Documentation Review	2 days	Wed 7/21/21	Thu 7/22/21		Michael
Sprint End/Demo	0 days	Thu 7/22/21	Thu 7/22/21		Michael
Programmer Guide & Deployment and Operations Guide (Runbook)	0 days	Thu 7/22/21	Thu 7/22/21		Michael

Task Name	Duration	Start	Finish	Predecesso	Resource Names
Execution Phase (Milestone 4)	**2.2 wks**	**Fri 7/23/21**	**Fri 8/6/21**		
Launch Android or IOS app in the stores	10 days	Fri 7/23/21	Thu 8/5/21		All Team member
Share it to the DSO team	10 days	Fri 7/23/21	Thu 8/5/21		All Team member
Finalize Test Report/Result	10 days	Sun 7/25/21	Thu 8/5/21		Damion,Prince
User Guide and Test Report	0 days	Fri 8/6/21	Fri 8/6/21		Michael
Finish Products	0 days	Thu 8/5/21	Thu 8/5/21		All Team member

Appendix B – Project Timeline