

Project Plan: Harkify  
SWEN 670  
University of Maryland Global Campus  
Version 1.0.0

For approval: Dr. Mir Assadullah  
Date: 06/07/2021

**Revision History**

<b>Date</b>	<b>Version</b>	<b>Description</b>	<b>Author</b>
06/07/2021	1.0.0	Initial Draft	(Team Bravo) Alec Baileys, Raul Benavides, Ben Cushing, Madison Dunning, Elshaday Mesfin, Tyler Puschinsky
06/11/2021	1.0.1	General revision: Addition of Risk mitigation in terms of PII	Team Bravo
06/19/2021	1.0.2	Added Use cases for Document Encryption, Notifications, and Favorites Page	Team Bravo

## Table of Contents

Background.....	1
Statement of Need .....	1
Vision Statement .....	1
Project Tools.....	2
Microsoft Office Suite .....	2
Microsoft Teams.....	2
Confluence.....	2
Jira .....	2
GitHub .....	2
Pico Voice .....	2
Project Deliverables and Milestones .....	3
Definitions of Acronyms and Abbreviations.....	4
Scope Management Plan .....	5
Project In-Scope Activities.....	5
Project Out-of-Scope Activities .....	6
Success Criteria .....	7
Project Assumptions .....	7
Project Dependencies .....	7
Project Approach.....	8
Project timeline (Gannt Chart) .....	8
Project Work Breakdown .....	10
Sprint 1 .....	10
Sprint 2 .....	10
Sprint 3 .....	11
Sprint 4 .....	11
Sprint 5 .....	11
Sprint 6 .....	11
Sprint 7 .....	11
Sprint 8 .....	12
Post-Development .....	12
Project Budget .....	12
Requirements Management Plan.....	13
Pending Functional Requirements.....	13

Integration Management Plan.....	14
Change Management Plan .....	15
Quality Management Plan .....	15
Resource Management Plan .....	16
Project Team Roles and Responsibilities .....	17
Responsibility Assigned Matrix (RACI) .....	18
Risk Management Plan .....	18
Risk Management Strategies .....	20
Risk Register .....	21
Risk Analysis.....	23
Risk Mitigation Strategies .....	23
Communication Management Plan .....	24
References .....	25

# Background

Our team has been tasked with developing an application to assist those with short-term memory impairment for our capstone project. As a class we separated into teams and our mentors assigned project managers (PMs) and a DevSecOps(DSO) team to assist in the creation and deployment of the project. This document will outline the project plan that will be used to accomplish the goal of deploying a functional application.

## Statement of Need

Approximately 14 million people (about twice the population of Arizona) in the United States suffer from a disability that impairs their short-term memory. This ranges from mild retention issues due to aging, to advanced memory loss such as Alzheimer's. There is a need to support those suffering from short-term memory loss or recognize cognitive decline.

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. After the completion of the project, anyone in the United States suffering from short-term memory (STM) will be able to download and use the application.

The project audience includes any of the approximately 14 million Americans that suffer from a disability that impairs their short-term memory that can work a smartphone and understand how to trigger the application.

Development and DevSecOps teams will be working remotely, selected after an application process conducted by UMGC consultants and United Global Masters Coders.

## Vision Statement

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

# **Project Tools**

The tools below will be used to manage the project as well as assist in creation the project deliverables create the project deliverables.

## **Microsoft Office Suite**

Microsoft Office applications will be used for various purposes for this project. Word will be used for creating deliverables such as reports and documentation, and PowerPoint will be used for creating the deliverables that require a presentation. Visio will be used for creating UML and other diagrams, and Project will be used for managing the project by creating Gantt charts and assigning team members to work on tasks.

## **Microsoft Teams**

Teams will be used as the primary method for team communication and allow for some collaboration.

## **Confluence**

Confluence is the primary method that team members will use to collaborate to create reports and documentation for the project. The application allows users to write and create without concern of formatting. Once a final draft is required, the information from Confluence will be downloaded and formatted in Word.

## **Jira**

Sprints and backlog will be managed by Jira. Tasks will be assigned a story score and team members must keep the Team up to date on the progress for their tasks. The creation of document deliverables will also use Jira manage the tasks involved.

## **GitHub**

Project code deliverables will be hosted on GitHub.

## **Pico Voice/Rhino**

This service is responsible for the talk to text services that our application will be responsible for. It provides customizable wake commands and allows for users to identify. Finally, this service is an open-source service allowing the business to not incur overhead.

## Azure

This service is responsible for the voice recognition portion of the application. This will provide user profiles while also maintaining user privacy by not saving PII information on the user. It provides a free testing service to allow users to interact with the service before purchasing.

## Project Deliverables and Milestones

The project will be broken into four distinct milestones, and two project presentations over a course of eleven weeks. The timeline for all deliverables can be seen in *Table 1 – Schedule of Deliverables*.














Milestone one contains the Project Plan and the Software Requirements Specification. The Project plan must include what is to be contained in the following three milestones. Milestone one also includes requirements gathering, resulting in a formal Scope Statement signed by the project stakeholders. After this milestone is delivered, a presentation will be conducted to Dr. Assadullah, including an overview of the Project Plan.

Milestone two will include delivery of the Technical Design Document. At this milestone, the Software Test Plan will also be added to the Project Plan developed for Milestone one, along with any other required updates to the Project Plan.

Milestone three will be composed of the Programmer Guide designed to aide future programmers working on the project software. It will also include a Deployment and Operations Guide, or runbook delivered by the Development team that guides the deployment of the project software.

Milestone four includes the finished project software. At this time, the User Guide will also be submitted, along with the Test Report. After this milestone is delivered, a final presentation will be conducted to Dr. Assadullah, including a demo of the software usage and lessons learned.

Table 1- Schedule of Deliverables

Title	T/M		Start	End		%
Milestone 1	M		6/8/2021	6/8/2021	–	%
- Project Plan	T		5/19/2021	6/2/2021	2 weeks	%
- SRS	T		6/2/2021	6/8/2021	1 week	%
Milestone 2	M		6/29/2021	6/29/2021	–	%
- Technical Design Document	T		6/9/2021	6/22/2021	2 weeks	%
- Software Test Plan	T		6/23/2021	6/29/2021	1 week	%
Milestone 3	M		7/20/2021	7/20/2021	–	%
- Programmer Guide	T		6/30/2021	7/13/2021	2 weeks	%
- Deployment and Operations Guide	T		7/14/2021	7/20/2021	1 week	%
Milestone 4	M		8/3/2021	8/3/2021	–	%
- Test Report	T		7/20/2021	7/27/2021	1 week	%
- User Guide	T		7/28/2021	8/3/2021	1 week	%
- Finished Project Software	T		8/3/2021	8/3/2021	1 day	%

## Definitions of Acronyms and Abbreviations

AC: Acceptance Criteria

BA: Business Analyst

DSO: Development/Security/Operations aka DevSecOps

MS: Microsoft

PM: Project Manager

QA: Quality Assurance

RACI: Responsibility Assignment Matrix

SRS: Software Requirements Specification

STM: Short-term memory

TDD: Technical Design Document

UMGC: University of Maryland Global Campus

UML: Unified Modeling Language



# Scope Management Plan

In scope activities include project management, design validation, testing, the deployment of the application, and any project artifacts needed to meet milestones and requirements.

## Project In-Scope Activities

The following deliverables will be delivered before following deadlines by the end of the project:

- Project Plan
  - o June 8, 2021
  - o This document that details how the project will be organized, the scope of the projects, a breakdown of the work required, and various management plans.
- Software Requirements Specification (SRS)
  - o June 8, 2021
  - o This document is an agreement between the team and the customer detailing the requirements for the product to be designed by the project.
- Technical Design Document (TDD)
  - o June 29, 2021
  - o Details the entire design of the system being developed by the team [10].
  - o It specifies how the program will behave to those outside the development team and how the functionality will be implemented in code [10].
  - o Written by the entire team, it contains detailed class models, the signature of the interface, any algorithms employed by components, and Physical data models [10].
- Programmer Guide
  - o July 20, 2021
  - o Justifies design and technical decisions that went into working on the project.
  - o Can bring new developers or teams assigned to the project helping reduce the time needed to get up to speed with the project.

## Deployment and Operations Guide (Runbook)

- o July 20, 2021
- o Set of documents, references, and procedures created for giving step-by- step guidance to accomplish a specific task or troubleshooting a particular issue [12].
- User Guide
- o August 10, 2021
- o Allows users of the application to correctly use and utilize all the features of the application.
- Test Report
- o August 10, 2021
- o Summary of testing, objectives, activities, and results. It is used to help stakeholders determine the quality of the product, and decide whether the product, feature or defect resolution is on track for release [5].
- Presentations
- o Team must make a presentation to Dr. Assadullah after Milestone 1 and 4 (June 8, 2021, and August 10, 2021). Separate presentations for the customer may occur. Team may request to give presentations for Dr. Assadullah for Milestone 2 and 3 (June 29, 2021, and July 20, 2021).

## Project Out-of-Scope Activities

Out-of-Scope activities include those that will be accomplished by other teams. This includes but is not limited to:

- Deliverables created by other teams.
- Applications developed by other development teams.
- Project infrastructure hosting, setup, and maintenance.
- Project post development hosting.

## Success Criteria

Success criteria for this project are defined as follows:

- Complete all milestones on time with the deliverables required.
- Provide all documentation of the project to stakeholders.
- Deploy a final product that covers the requirements posed by the stakeholders.
- Project incorporates the required external services for speech synthesis and isolation.
- Stay within budget.
- Code coverage stays at or above 60%.
- Foster open communication and constant communication with stakeholders.

## Project Assumptions

Research and stakeholders have helped comprise project assumptions. The assumptions are:

- Access to the stakeholders varies by person but anyone will be available when needed.
- Team size will not change during the duration of project completion.
- The application will identify the user's voice and limit the recordings to only the user.
- Simplicity in the application and natural speech patterns benefit the target audience, dementia patients.
- The application will have a wake command to that will initiate a recording and this command must be customizable.
- The application shall be developed in Flutter and Dart.js as a mobile application for both Android and IOS smartphones.
- The application shall be designed specifically for its target demographic of elderly users suffering from memory loss.

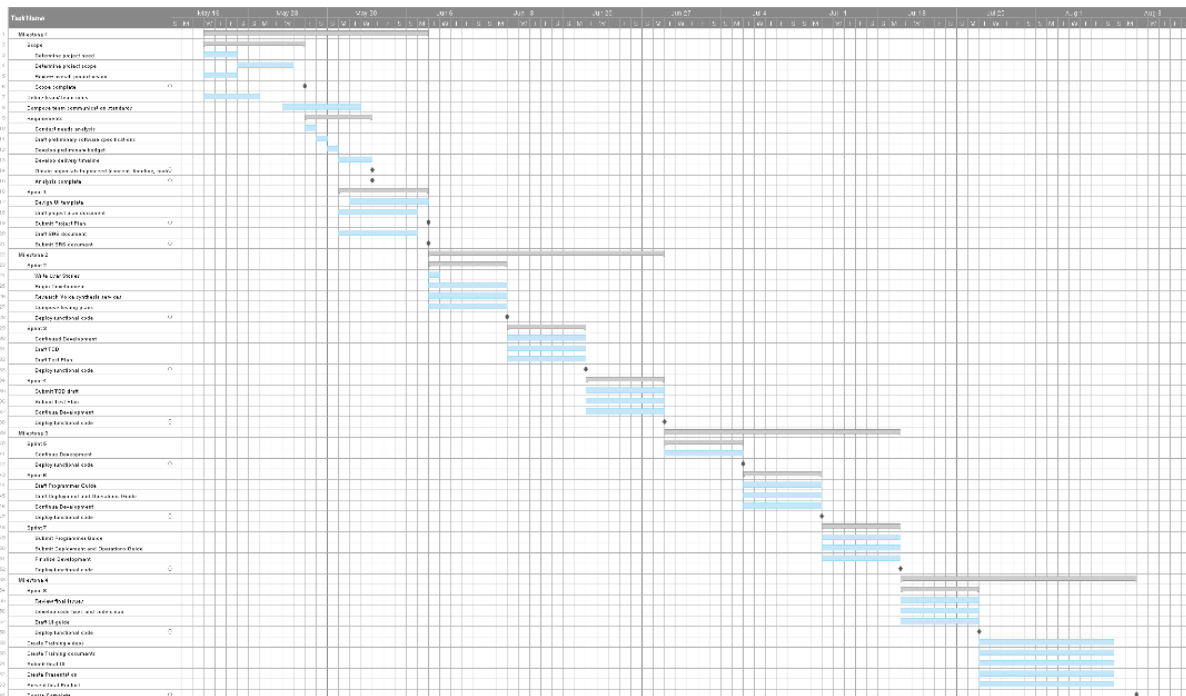
## Project Dependencies

The application depends upon the smartphone where it is installed for access to a microphone, speaker, and text note storage. If the smartphone does not provide access to these features, the application shall not function. Furthermore, given the project's nonexistent budget, application development is completely dependent upon free and open-source software.

# Project Approach

This project is intended to be completed in the 12-week capstone course. We must be able to deliver a functional application that fulfills the projects core requirements. To be able to complete the project we have decided to use an agile approach as our development methodology. By using Agile, we think it allows our team the flexibility to work on the project at the times that work best for us while being able to keep track of progress and have a full understanding of what has been completed in the project.

## Project timeline (Gantt Chart)



Task	Days to Complete	Start Date	End Date
Milestone 1	20d	5/19/2021	6/8/2021
Scope	9d	5/19/2021	5/28/2021
Define team/ team roles	5d	5/19/2021	5/23/2021
Compose team communication standards	7d	5/26/2021	6/1/2021
Requirements	6d	5/28/2021	6/3/2021
Sprint 1	8d	5/31/2021	6/8/2021
Design UI template	7d	6/1/2021	6/7/2021
Draft project plan document	7d	5/31/2021	6/6/2021
Submit Project Plan	0	6/8/2021	6/8/2021

Draft SRS document	7d	5/31/2021	6/6/2021
Submit SRS document	0	6/8/2021	6/8/2021
Milestone 2	21d	6/8/2021	6/29/2021
Sprint 2	7d	6/8/2021	6/15/2021
Write User Stories	1d	6/8/2021	6/8/2021
Begin Development	7d	6/8/2021	6/14/2021
Research Voice synthesis services	7d	6/8/2021	6/14/2021
Compose testing plans	7d	6/8/2021	6/14/2021
Deploy functional code	0	6/15/2021	6/15/2021
Sprint 3	7d	6/15/2021	6/22/2021
Continued Development	7d	6/15/2021	6/21/2021
Draft TDD	7d	6/15/2021	6/21/2021
Draft Test Plan	7d	6/15/2021	6/21/2021
Deploy functional code	0	6/22/2021	6/22/2021
Sprint 4	7d	6/22/2021	6/29/2021
Submit TDD draft	7d	6/22/2021	6/28/2021
Submit Test Plan	7d	6/22/2021	6/28/2021
Continue Development	7d	6/22/2021	6/28/2021
Deploy functional code	0	6/29/2021	6/29/2021
Milestone 3	21d	6/29/2021	7/20/2021
Sprint 5	7d	6/29/2021	7/6/2021
Continue Deveopment	7d	6/29/2021	7/5/2021
Deploy functional code	0	7/6/2021	7/6/2021
Sprint 6	7d	7/6/2021	7/13/2021
Draft Programmer Guide	7d	7/6/2021	7/12/2021
Draft Deployment and Operations Guide	7d	7/6/2021	7/12/2021
Continue Development	7d	7/6/2021	7/12/2021
Deploy functional code	0	7/13/2021	7/13/2021
Sprint 7	7d	7/13/2021	7/20/2021
Submit Programmer Guide	7d	7/13/2021	7/19/2021
Submit Deployment and Operations Guide	7d	7/13/2021	7/19/2021
Finalize Development	7d	7/13/2021	7/19/2021
Deploy functional code	0	7/20/2021	7/20/2021
Milestone 4	21d	7/20/2021	8/10/2021
Sprint 8	7d	7/20/2021	7/27/2021
Review final issues	7d	7/20/2021	7/26/2021
Develop code fixes and code clean	7d	7/20/2021	7/26/2021
Draft UI guide	7d	7/20/2021	7/26/2021
Deploy functional code	0	7/27/2021	7/27/2021
Create Training videos	12d	7/27/2021	8/7/2021

Create Training documents	12d	7/27/2021	8/7/2021
Submit final UI	12d	7/27/2021	8/7/2021
Create Presentation	12d	7/27/2021	8/7/2021
Present final Product	12d	7/27/2021	8/7/2021
Course Complete	0	8/10/2021	8/10/2021

## Project Work Breakdown

We have broken down our 12 weeks by making the first two weeks requirements collection and project planning weeks, followed 8 weeks of 8 sprints and finalizing our project with two weeks for maintenance, documentation, and presentations. Each team member is dedicating 10 hours a week, for both development and documentation, and staying open for communication throughout the day. Each Story Point has been assigned an hour, with 6 team members, our team has a sprint velocity of 60 story points with the total project costing 480 story points. The entire project will be tracked using Jira and progress will be maintained through Jira's table system. The sprint breakdown are as follows:

### Sprint 1

The first sprint will be dedicated to designing, creating wireframes, and UI design walkthroughs to have a clear understanding of the project at hand. The team will also dedicate time to finish deliverables and clear up any misunderstanding that we may have before moving forward. On the development side, the development team will focus on creating a stable application infrastructure and uploading the code to our team GitHub repository. This way the team can become familiar with flutter and dart basics while the set up for the environment is already created. Finally, if the UI design is created, we will breakdown the design into user stories and use the requirements to build ac's for the user stories. The success of the sprint will be determined by the completion of Milestone 1 deliverables and stable flutter application creation.

### Sprint 2

The goal of this sprint is to be able to start development of user stories. Using Jira, the team will be able to see the status of the user story and leave their notes as comments to the story. The path forward will be a feature driven development where all aspects of a feature will be developed before moving onto other features. This will allow us to have fully functional code without splitting the team thin working on multiple areas at a time. This will also allow other developers time to learn as much as possible of voice synthesis and integration services before choosing the ideal one for our project. Success of Sprint 2 is determined by the completion of user stories comparable to the total story point workload of the project.

## **Sprint 3**

The goal of sprint 3 will be development of the project as well as beginning work on the TDD and test plan. After reviewing the user stories, requirements and wireframes, a testing plan can be created and begin writing. The team will also simultaneously document their code and decisions so that they can be added to the TDD. Decisions of the speech to text services will be made by the end of this sprint. If any issues or complications arise with this it will be shared with other teams, but final decisions of design plans will be made at the end of this sprint.

## **Sprint 4**

Sprint 4 will be dedicated to the deliverables that are required for Milestone 2. This fits into our development plan as we have incorporated the need to focus on documentation into sprint expectations. By completing some development before the completion of the deliverables, Technical Design Document and test plan, the team will be fully aware of the team design decisions so that everyone can then contribute to the document. Extended goals will be to continue development on the project.

## **Sprint 5**

Development will continue for the full team in Sprint 5. This is the beginning of Milestone 3 which will require the programmer guide and the Deployment and Operations Guide. the Milestone deliverables requires that the project be completed by then. To be able to accomplish this development and notes will be tracked and compiled as we go through development. Sprint 5 will allow us to have a week of focused development.

## **Sprint 6**

In sprint 6 the programmers guide as well as the deployment and operations guide will start being worked on. Our DSO Engineer will begin to lead in the documentation efforts for the deployment and operations guide and begin to summarize his findings. the team will continue to finish development and make the effort to finish final development user stories. User stories that could not be finished will be addressed as a team and narrow down the required stories.

## **Sprint 7**

In sprint 7, Milestone 3 deliverables will be the focus of the sprint. The team will focus on getting the deliverables and separating labor as needed. With the conclusion of the project, we will write up everything that we can about the project that we can present. Anything that was not able to be submitted will have to be reviewed and analyzed in the final sprint.

## Sprint 8

Finally, sprint 8 will be dedicated to going over the final product and seeing if anything can be addressed in the application. By this point the documentation is submitted, however, if there are any minor issues or functionality issues that still need to be addressed, they will be covered in this sprint. This sprint will be for code clean up or anything that can be accomplished before presentation and UI guide. The team will also begin work on UI guide, training videos and presentation.

## Post-Development

The conclusion of Sprint 8 would be the end of development for the team as the full project should be deployed. The last two weeks of the project will be dedicated to finalizing the Milestone 4 deliverables and creating the necessary training guides and walkthroughs as well as the final presentation. In agile, the final stage would be maintenance however, since the life of the application is only contained in the parameters of the class, maintenance will not be required.

## Project Budget

This section breaks down the budget for the project. The team is going to make the effort in creating minimal overhead by only using open-source services in their development. This means that the IDE, voice recognition services, code repositories, and testing systems will all be open to the public. By doing so the only charge that will be created is for the positions that will be filled to complete the project. The salary for each role was determined by using Glassdoor average estimates for each position. Even though each position has varying salaries due to title, all team members will be involved in the development of the project and assuring that the project deliverables are submitted on time.

### Weekly Budget summary

Week	Start Date	End Date	Total Hours	Billable Rate	Budget per week
1	5/19/2021	5/25/2021	60	\$ 265	\$15,908.65
2	5/26/2021	6/1/2021	60	\$ 265	\$15,908.65
3	6/2/2021	6/8/2021	60	\$ 265	\$15,908.65
4	6/9/2021	6/15/2021	60	\$ 265	\$15,908.65
5	6/16/2021	6/22/2021	60	\$ 265	\$15,908.65
6	6/23/2021	6/29/2021	60	\$ 265	\$15,908.65
7	6/30/2021	7/6/2021	60	\$ 265	\$15,908.65
8	7/7/2021	7/13/2021	60	\$ 265	\$15,908.65
9	7/14/2021	7/20/2021	60	\$ 265	\$15,908.65
10	7/21/2021	7/27/2021	60	\$ 265	\$15,908.65
11	7/28/2021	8/3/2021	60	\$ 265	\$15,908.65
12	8/4/2021	8/10/2021	60	\$ 265	\$15,908.65
				<b>Total</b>	\$190,903.85



## Billable rate estimation breakdown

Role	Salary per Year	Hourly rate
Project Manager	\$ 100,000	\$ 48
Lead Developer	\$ 110,000	\$ 53
Test Engineer	\$ 73,500	\$ 35
DSO Engineer	\$ 108,000	\$ 52
Business Analyst	\$ 80,000	\$ 38
Business Analyst	\$ 80,000	\$ 38
<b>Total</b>	\$ 551,500	\$ 265

# Requirements Management Plan

It is Bravo Team's responsibility to successfully implement the pending functional requirements in the finished product. Due to the limited time available to completing this project, all team members work to ensure that all requirements are implemented properly. Sprints will be planned using tools such as Jira and Microsoft Project. The product shall be systematically tested to determine if it meets the functional requirements.

## Pending Functional Requirements

1. Use Case: Start Application  
**Summary:** This feature allows a user to start the application so that it begins to listen for trigger phrases and is ready to accept commands.
2. Use Case: Train Voice Recognition  
**Summary:** This feature allows a user to train the application to recognize the user's voice for commands.
3. Use Case: Record Save Trigger  
**Summary:** This feature allows a user to record a trigger phrase so that the application initiates the "Save Note" use case when the trigger is heard.
4. Use Case: Record End Trigger  
**Summary:** This feature allows a user to record a trigger phrase so that the application stops recording a note during "Save Note" use case when the trigger is heard.
5. Use Case: Record Search Trigger  
**Summary:** This feature allows a user to record a trigger phrase so that the application initiates the "Search Notes by Voice" use case when the trigger is heard.
6. Use Case: Save Note  
**Summary:** This feature allows a user to record speech as a text note by speaking the save trigger phrase or selecting the "Save Note" option.
7. Use Case: View Note  
**Summary:** This feature allows a user to employ the GUI to view, edit or delete an existing note on the screen.
8. Use Case: View Settings  
**Summary:** This feature allows a user to view current application settings, edit those settings, record speech triggers or train voice recognition.

9. Use Case: Search Notes by Voice  
**Summary:** This feature allows the user of the software to search a specific event of the recorded conversations using key phrases provided.
10. Use Case: View Notes  
**Summary:** This feature allows the user of the software to view existing text notes and search for them using key words displayed on the GUI.
11. Use Case: Stop Application  
**Summary:** This feature allows the user of the software to end the session /close out of the application.
12. Use Case: Welcome New User  
**Summary:** This feature allows a user to configure application settings after starting the application for the first time.
13. Use Case: Encrypt saved document  
**Summary:** This feature allows a user to provide personal information to the application with knowledge of security.
14. Use Case: Favorites Page  
**Summary:** This feature allows a user to store notes that cannot be deleted directly from this page or from the application deleting settings.
15. Use Case: Notifications  
**Summary:** This feature reminds a user of the application and to remember to record notes.
16. Use Case: Training videos  
**Summary:** This feature demonstrates training videos on how to use the application.

## Integration Management Plan

Most tasks in the project will be managed through Jira where they can be easily assigned and progress towards goals can be tracked. Team members who have completed their assignments can check Jira to see where they can pick up slack. GitHub allows for collaboration between developers, where changes can be easily tracked. To do so each team member will create a branch off of the main branch in Github. Once the developer has completed their story, they will be able to create a pull request to merge their code into the main branch. This code will be reviewed by another person in the team and by a member of the DSO team. Once the code is approved by both reviewers, the code will be brought into the main branch. This will reduce the likelihood of deployment failure and improve code quality. For deliverable documents, changes can also be tracked using Confluence. Using Confluence allows the team to easily view what other team members have completed and assign sections that have not been claimed. Once the full scope of writing is completed, the document will be transferred to word where it will be formatted for final submission.

Using these systems, the PM can see where the team stands with their tasks, and they reduce the need for additional meeting times during the week. Based on the information displayed to the PM on Jira, they may modify the sprints and make any necessary changes to their plans. Regarding building and testing the software, a member of the team has been designated to

be a contact between Bravo Team and the DSO team. The DSO team are responsible for ensuring that the software is being built and tested properly.

## **Change Management Plan**

Since our project has chosen to use the Agile methodology, our change management plan will be light on process. An explicit Change Advisory Board (CAB) is not necessary and formally documented change requests are not required. Instead, our change management process will rely on common agile principles and practices.

Weekly status meetings are the appropriate time for team members to mention roadblocks or potential changes where the team can discuss them. Together the project team will decide if changes should be implemented or abandoned. Smaller changes will be resolved in this fashion with JIRA tickets created, as necessary. Larger issues will be brought back to the other project PMs and the client for discussion, resulting in JIRA tickets when needed. At the end of each milestone sprint, sprint review meetings will be held to discuss the team's progress. Here, more changes may be discussed to affect future sprints.

When the project teams designate changes for inclusion in the project, the team will determine whether those tickets should be added to the current sprint, a future sprint, or remain in the project backlog for implementation at an indeterminate date. JIRA tickets shall be used to document any notable change performed by a project team member.

## **Quality Management Plan**

Testing will be performed throughout the development of the application, broken into 4 test categories: unit, integration, acceptance, and regression testing. The Software Test Plan will define the scope and content of each category of testing in detail. The primary goals of testing are to confirm the functionality defined in the requirements document are present and the functionality in the system performs correctly. Frequency of test execution will depend on the type of test, automated or manual. Automated tests will be executed with every build of the application, manual tests will be executed when the associated functionality is delivered, as well as at the conclusion of the project to produce a final test report.

# Resource Management Plan

Although team members have been assigned to roles, all members are expected to wear multiple hats and assist the project in other capacities, as necessary. Staffing resources have been distributed for this project as follows:

- Alec Baileys – DevSecOps Engineer
- Raul Benavides – Project Manager
- Ben Cushing – Business Analyst
- Madison Dunning – Developer
- Elshaday Mesfin – Business Analyst
- Tyler Puschinsky – Test Engineer
- Dr. Mir Mohammed Assadullah – Product Owner
- Roy Gordon – Mentor
- Jonny Lockhard – Mentor
- Betsy Weitz – Customer

Regarding space and equipment, no team space has been allocated and no new equipment purchased. All team members shall continue to work remotely using their personal equipment for all project work. This is in keeping with the lack of budget for this project.

In terms of technological needs, the project team shall persist with free and open-source software for development purposes and collaboration. These include but are not limited to Visual Studio Code, Flutter, Confluence and JIRA. For team coordination, the university's student access to MS Teams shall be used to minimize project costs.

## Project Team Roles and Responsibilities

**Project Manager:** The project manager will coordinate all project operations while developing and maintaining the project timeline. The project manager will also ensure that all project tasks are appropriately staffed and handle the presentation of deliverables.

**Business Analyst:** The business analyst will draft project requirements based on the product owner's requests and act as a liaison between the product owner and developers in communicating what the application should do.

**Developer:** The developer will be primarily responsible for evaluating potential technical solutions and creating the software to run the proposed application.

**Test Engineer:** The test engineer will focus on ensuring the validation and verification of project requirements. Ultimately, the test engineer confirms that all requirements have been satisfied.

**DevSecOps Engineer:** The DevSecOps engineer will coordinate continuous integration and continuous deployment of the project's software application with the overall DevSecOps team.

**Product Owner:** The product owner establishes the vision for what the product should be while prioritizing all proposed features. The product owner represents the application's customer in establishing their needs and confirming that those needs have been met.

**Customer:** The customer will provide insight and feedback of the product as a representative of the target audience. Their feedback and suggestions will be used to be able to create a better product for those that will be using the application.

## Responsibility Assigned Matrix (RACI)

BravoTeam RACI Matrix	Dr. Mir Assadulla	Roy Gordon	Jonny Lockhard	Betsy Weitz (customer)	Raul Benavides (PM)	Madison Dunning (LD)	Alec Baileys (DSO/DEV)	Benjamin Cushing (BA)	Elshaday Mesfin (BA)	Tyler Puschinsky(TST)
R - Responsible A - Accountable C - Consulted I - Informed										
Deliverable	Stakeholder/Leadership				Project team					
<b>Milestone 1</b>	C	C	C	C	A	R	R	A	R	R
Requirements Gathering					R	C	C	R	R	C
Project Scope Planning					R	C	C	R	R	C
Project Plan					R	R	R	R	R	R
SRS Document					I	I	I	R	R	I
Internal-Intra Team Meetings					R	R	R	R	R	R
Client/Project Team Meetings					R	I	I	C	I	I
<b>Milestone 2</b>	C	C	C	C	A	R	R	A	R	R
Technical Design Document					R	R	R	R	R	R
Software Test Plan					C	C	C	R	R	R
Internal-Intra Team Meetings					R	R	R	R	R	R
Client/Project Team Meetings					R	I	I	C	I	I
<b>Milestone 3</b>	C	C	C	C	A	R	R	A	R	R
Programmer Guide					R	R	R	R	R	R
Deployment and Operations Guide					R	R	R	C	C	C
Internal-Intra Team Meetings					R	R	R	R	R	R
Client/Project Team Meetings					R	I	I	C	I	I
<b>Milestone 4</b>	C	C	C	C	A	R	R	A	R	R
User Guide					R	R	R	R	R	R
Test Report					C	C	C	R	R	R
Training Videos					R	R	R	I	I	I
Presentation					R	R	R	R	R	R
Final Deployment					C	R	R	C	C	R
Internal-Intra Team Meetings					R	R	R	R	R	R
Client/Project Team Meetings					R	I	I	C	I	I

## Risk Management Plan

The main goal of developing or undertaking a project requires identifying the challenges associated with risk management. The processes of risk management could use subjective or quantitative methods, by estimating the processability during a particular procedure, to be evaluated and used for further improvement through the length of the project. A substantial portion of the project will be focusing on specifying the correct methodology and implementation procedures for adjusting and picking the correct organizationally focused difficulties for executing any goals set. The purpose of developing a risk management process is to examine and assess, besides distinguishing any risk that needs evaluating, analyzing, treatment, and monitoring options for it to have a better outcome. [1] An effective risk management process follows the steps of recognizing the risk, investigating the recognized risk,

assessing the risk against characterized criteria, choosing the risk for treatment, and finally treating the chosen risk.



[6]

Even though there is no requirement for the use of different methodologies in the risk management process, there are present measures featured in the risk management comprehensive pathways offered by the international standards to be used as a guideline when developing organizationally specific methods for integrating the organizational resources for implementing the framework and treating any risk selected for treatment.

The project risk management process should embrace proactively dealing with the overall risk associated with the organization/project and not solely focus on individual risks. The project manager should be able to identify any opportunities, impacts, probability and not just focus on the time, the scope of the project, and the costs. Each project has its own risk associated with it and there is no universal method to manage it all. The best ways to visualize that there are risks associated with every step of the project life cycle, thus it is important to know which organizational resources to utilize and plan, to minimize or mitigate any risk that might develop. It is also important to involve the stakeholders during the planning phase. There are multiple types and categories of risk, and identification of the risks is the second step after the planning phase.

A few risks associated with projects include lack of planning, lack of knowledge of both team and stakeholders, requirements/scope of the project, project schedule, internal and external factors, communication issues, management, unclear roles of the team members, decision-making process, security measures, and system structures. The fundamental reason for the risk management process is not to avoid those risks but to be able to plan and manage them so that the outcome of the project would not be affected.

## Risk Management Strategies

Initially, risks of the software development life cycle are cataloged into four distinct types. This risk included process, project, organizational and technical risks. This risk management plan outlines the responses to mitigate in case of occurrences of these risks. [8] The main focus of the risk management plan consists of a risk register. The risk register defines and outlines probable threats to the SDLC. The attributes of a risk register are described in table 1.

Risk Attributes	Definition
Description of risk	Summary description of the risk—easy to understand.
Recognition Date	Date on which stakeholders identify and acknowledge the risk.
Probability of occurrence	Estimate of probability that this risk will materialize (%)
Severity	The intensity of undesirable impact to the project—if the risk materializes.
Owner	This person monitors the risk and acts if necessary.
Action	The contingent response if the risk materializes.
Status	current team view of the risk: potential, monitoring, occurring, or eliminated.
Loss Size	Given in hours or days, this is a measure of the negative impact to the project.
Risk Exposure	Given in hours or days, this is a product of <i>probability</i> and <i>loss size</i> .
Priority (optional)	This is either an independent ranking, or the product of probability and severity. Typically, a higher-severity risk with high probability has higher relative priority.



# Risk Register

Project name: Short-Term Memory Assistant

Project Manager: Raul Benavides

ID	Date Raised	Risk Description	Probability of Occurrence	Severity	Owner	Mitigation Action	Status	Priority
1	[enter date]	Project purpose and need not well defined	Medium	High	Project sponsor	Complete business case	open	High
2	[enter date]	Incomplete project design and deliverable definition	Low	High	Project sponsor	Define detail scope from subject matter experts.	open	High
3	[enter date]	Misunderstood and ambiguous requirements	Low	High	Project manager	Multiple checkpoints with stakeholders, early delivery of drafts.	open	High
4	[enter date]	Project Schedules not clearly defined and understood	Low	Medium	Project manager	Hold scheduling meetings with project team.	open	Low
5	[enter date]	System development and integration complexity	Medium	High	Software architect/Developers	Early development of integration plan with formal entry of requirements for components entering integration.	open	High

6	[enter date]	Availability of finance	High	High	Project sponsor	Own resources, provide necessary tools for development and production.	open	High
7	[enter date]	Inadequate team members available	Low	High	Project sponsor	Multi-role/ versatile team members to get involved on various tasks as needed.	open	Medium
8	[enter date]	Lack of control of team priorities	Low	Medium	Project manager	Provide schedules with clear deadlines for tasks and are easily adjustable giving flexibility to team members.	open	Low
9	[enter date]	Insufficient time for stakeholders' feedback	Low	Medium	Project manager	Early development and deliverables.	open	Medium
10	[enter date]	Insufficient QA and end user validation	Medium	High	Project manager and Team lead	Agile development allowing for preliminary assessment to complete validation and verification.	open	High

[11]

## Risk Analysis

- 1) **Scope risks:** Team Bravo will submit a scope of work to the client and upon written agreement commence the project to prevent poor specification of the scope of work or unnecessary and unexpected requests from the client during the project development.
- 2) **Technical risks:**
  - a) **Application is not secure:** All the operations on the database will be performed via HTTPS request using JSON web token to secure the operations. [4] Industry-standard methods to protect the database from SQL injection and brute force attacks will be adopted to prevent any malicious input to the database that jeopardizes the system's response. [2,7]
  - b) **Application is not scalable:** To address the increased load or features to be included in the system later in the future, a scalable product will be designed and deployed. [3]

## Risk Mitigation Strategies

Much of the focus during software development projects should be the software project tasks which are continuously monitored by checking the project progress during critical activities and in advance scheduled meetings. Effective communication between managers, developers, and stakeholders is a key factor for reducing risk and as a main component of risk mitigation planning. Risk monitoring activities incorporate revision of risk plan, publication of project report, evaluation of possible new risks after modification, modification of project timeline or project scope. There will be informational guide to user not to store personal identifiable information (PII) through the application in order to prevent risks associated with securing and managing sensitive data. In situations where the user enters any PII inadvertently, some of the risk mitigation tactics for securing the PII are to reduce attack surface, remember to encrypt data, depersonalize sensitive data and delegate compliance to third party [13]. Some mitigation options relating to all risks include accepting, avoiding, controlling, transferring, and ongoing monitoring. The options with details in the table below [9].

Mitigation Option	Detail
Accept	Recognize how the risk impact the project.
Avoid	Adapt and adjust the project scope, timeline, or issue to reduce the risk effects.
Control	Act on issues to reduce the impact or risk intensity.
Transfer	Regarding accountability, authority or responsibility to other stakeholders that accept the risk.
Ongoing monitoring	For low-impact risks.
Project progress	For potentially high-risk impacts.

# Communication Management Plan

To be able to complete this project, The team must be able to always stay in communication. The team must be able to share ideas, concerns, and updates on a day-to-day basis as well as a full group meeting every week where everyone is focused on the deliverables. Ideas that go beyond our group and need to be shared with other groups or project managers, also must have a simple and central location for communication so that those with similar thoughts can be directed to that conversation. This way decided outcomes can be shared with everyone and reducing the amount of repetition of discussing the same topic. Reducing repetitive conversations is important due to the limited time given for this project.

For this reason, the team and class has decided to use Microsoft teams as our main form of communication with email to distribute and present important concepts. Microsoft teams allows us to schedule weekly online meetings as well as create group chats to be able to communicate as a team. There are also two separate accounts, the UMGC account and the United Global Master Coders account. The UMGC account is set up to allow the team to communicate, schedule meetings, and organize our deliverables, while the United Global Master Coders account allows us to communicate with other teams, project managers and stakeholders using a thread-based discussion system. This means that if a team member wishes to pose a discussion visible to everyone in the class, a discussion thread in the United Global Master Coders account can be created and those who respond can reply directly to the thread. If a conversation is being had in the UMGC account that pertains to more people or requires time sensitive answers, an email or a discussion thread can be created to reach out for more information. Finally, the customer, Betsy Weitz, is a family member of one of our team members, Benjamin Cushing. To communicate with her we have decided to allow all communication to go through Benjamin as to not overwhelm with questions.

Overall, Microsoft teams, multiple account types, and email will allow everyone the flexibility to communicate in whatever manner they choose and exposing the conversation to those that need to know.

# References

- [1] Barafort, B., Mesquida, A.-L., & Mas, A. (2018). Integrated risk management process assessment model for IT organizations based on ISO 31000 in an ISO multi-standards context. *Computer Standards and Interfaces*, 60, 57–66.  
<https://doi.org/10.1016/j.csi.2018.04.010>
- [2] Brute Force Attack: Definition and Examples (n.d.). Kaspersky. Retrieved from <https://www.kaspersky.com/resource-center/definitions/brute-force-attack>
- [3] Development, R. (n.d.). Risk Management in Web Development. Retrieved from <https://rubygarage.org/blog/risk-management-in-development>
- [4] Introduction to JSON Web Tokens. [JWT.IO](https://jwt.io/). (n.d.). Retrieved from <https://jwt.io/>
- [5] Kinsbruner, E. (2020, August 13). *Test Reporting: What It Is & How to Make It Work for Continuous Testing* | by Perforce. Perfecto by Perforce.  
<https://www.perfecto.io/blog/test-reporting>
- [6] Project Management (2012). Project Risk Management Handbook: A Scalable Approach. Caltrans. Retrieved from [http://www.dot.ca.gov/projmgmt/documents/Project\\_Risk\\_Management\\_Handbook.pdf](http://www.dot.ca.gov/projmgmt/documents/Project_Risk_Management_Handbook.pdf)
- [7] SQL Injection. (n.d.). W3 Schools. Retrieved from [https://www.w3schools.com/sql/sql\\_injection.asp](https://www.w3schools.com/sql/sql_injection.asp)
- [8] Software Development Risk Management Plan with Examples (n.d.). Cast. Retrieved from <https://www.castsoftware.com/research-labs/software-development-risk-management-plan-with-examples>
- [9] Strategies in Mitigating Risks in Software Development Projects. Fly Outsourcing. Retrieved from <https://flyoutsourcing.com/blog/strategies-in-mitigating-risks-in-software-development-projects.html>
- [10] Technical Design | University IT. (n.d.). Stanford | University IT. Retrieved June 2, 2021, from <https://uit.stanford.edu/pmo/technical-design>
- [11] 45 Useful Risk Register Templates. (n.d.). Template LAB. Retrieved from <https://templatelab.com/risk-register-templates/>
- [12] Watts, S. (2020, May 21). An Introduction to Operations Runbooks [Blog]. *BMC Blogs*.  
<https://www.bmc.com/blogs/operations-runbook/>
- [13] Krush, A (2020, November 30). Securing PII in a software system. Object style. Retrieved from <https://www.objectstyle.com/security/securing-personally-identifiable-information-in-a-software-system>