

Tongue Twisters Project Team:  
Obinna Okonkwo, Andrew Rohn,  
Leelakrishnan Subramaniam, Joseph  
Kalfus, Firehiwot Chari, Eskedar  
Endashw & Malik Webster

# Natural Language Understanding (NLU) Module

## Project Plan

Approved By: Obinna Okonkwo

---

# Table of Contents

1. Introduction .....	4
1.1 Background Information .....	4
1.2 Statement of Need .....	4
1.3 Vision Statement .....	4
2. Project Overview .....	5
2.1 Project Objectives .....	5
2.2 Project Scope .....	5
2.3 Major Stakeholders .....	5
2.4 Project Deadline .....	5
2.5 Acceptance Criteria .....	6
2.6 Constraints and Assumptions .....	6
2.6.1 Constraints .....	6
2.6.2 Assumptions .....	7
2.7 Acronyms and Abbreviations .....	8
3. Risk Analysis .....	8
3.1 Risk Matrix .....	9
3.2 Risk Register .....	9
4. Organization of the Project .....	11
4.1 Software Development Methodology .....	11
4.1.1 Requirements Planning Phase .....	12
4.1.2 Design Phase .....	12
4.1.3 Iterative Development Phase .....	13
4.1.4 Testing & Deployment Phase .....	13
4.2 Project Team .....	13
4.2.1 Roles & Responsibilities .....	14
4.2.2 Project Tools .....	15
4.3 Stakeholder Communication Project Plan .....	16
4.4 Scope Management Plan .....	18
4.5 Project Schedule .....	18
4.6 RACI Matrix .....	22
4.7 Project Milestones & Deliverables .....	23
4.8 Project Cost .....	24
5. Software Test Plan .....	25
5.1 Unit Test .....	25
5.1.1 Approach .....	25
5.1.2 Proposed Test Cases .....	26
5.1.2.1 Test Case 1: TC-001 .....	26

5.1.2.2 Test Case 2: TC-002.....	26
5.1.2.3 Test Case 3: TC-003.....	26
5.1.2.4 Test Case 4: TC-004.....	26
5.1.2.5 Test Case 5: TC-005.....	27
5.1.2.6 Test Case 6: TC-006.....	27
5.1.2.7 Test Case 7: TC-007.....	27
5.1.2.8 Test Case 8: TC-008.....	27
5.1.2.9 Test Case 9: TC-009.....	27
5.1.2.10 Test Case 10: TC-010.....	28
5.1.2.11 Test Case 11: TC-011.....	28
5.1.2.12 Test Case 12: TC-012.....	28
5.1.2.13 Test Case 13: TC-013.....	28
5.1.2.14 Test Case 14: TC-014.....	28
5.1.2.15 Test Case 15: TC-015.....	29
5.1.2.16 Test Case 16: TC-016.....	29
5.1.2.17 Test Case 17: TC-017.....	29
5.1.2.18 Test Case 18: TC-018.....	30
5.1.2.19 Test Case 19: TC-019.....	30
5.1.2.20 Test Case 20: TC-020.....	30
5.1.2.21 Test Case 21: TC-021.....	30
5.1.2.22 Test Case 22: TC-022.....	30
5.1.2.23 Test Case 23: TC-023.....	31
5.1.2.24 Test Case 24: TC-024.....	31
5.1.2.25 Test Case 25: TC-025.....	31
5.1.2.26 Test Case 26: TC-026.....	31
5.1.2.27 Test Case 27: TC-027.....	31
5.1.2.28 Test Case 28: TC-028.....	32
5.1.2.29 Test Case 29: TC-029.....	32
5.1.2.30 Test Case 30: TC-030.....	32
5.1.2.31 Test Case 31: TC-031.....	32
5.1.2.32 Test Case 32: TC-032.....	32
5.1.2.33 Test Case 33: TC-033.....	33
5.1.2.34 Test Case 34: TC-034.....	33
5.2 Testing Criteria .....	33
5.2.1 Suspension Criteria.....	33
5.2.2 Resumption Criteria .....	33
5.2.3 Test Completeness .....	33
5.2.4 Test Deliverables .....	33
6. References .....	34

## Revision History

Date	Version	Description	Author(s)
August 31 <sup>st</sup> , 2021	1.0	Initial publication of Project Plan.	Obinna Okonkwo, Andrew Rohn, & Eskedar Endashw
September 10 <sup>th</sup> , 2021	1.1	Risk Matrix and Risk Register were added to section 3. Stakeholder Communication Project Plan, Scope Management Plan, and RACI Matrix were added to section 4.	Obinna Okonkwo & Andrew Rohn
September 14 <sup>th</sup> , 2021	2.0	Software Test Plan was added to section 5.	Obinna Okonkwo, Andrew Rohn, Eskedar Endashw, & Joseph Kalfus

# **1. Introduction**

## **1.1 Background Information**

This Project Plan provides a project management framework for the Natural Language Understanding (NLU) module being developed for the Memory Magic Application. This project plan describes the project's objectives, scope, development process, work schedule and timeline, development costs, and potential risks.

The NLU module is being developed to assist users who suffer from short-term memory (STM) loss. STM is defined as the capacity to store small amounts of information for immediate use, generally 7 items or less for a duration of 15 to 30 seconds (McLeod, 2009). Contributing factors to STM loss include aging, sleep deprivation, nutritional deficiencies, brain injury, illness, trauma, brain infections, or the effects from drugs or alcohol use (Brown, 2021). A person suffering from chronic STM loss requires support to accomplish everyday tasks, such as remembering to purchase groceries. The Memory Magic Application being developed by the Mesmerize project team seeks to solve this problem by creating an application that recognizes human speech, allowing them to set reminders for themselves so that they do not forget things that are important to their day-to-day lives. The NLU module specifically focuses on determining what features of the Memory Magic Application will be executed based on the user's voice input. It is this part of the application that the Tongue Twisters project team has been tasked to develop and it is the focal point of this project plan.

## **1.2 Statement of Need**

People who suffer from STM loss often worry about their ability to remember crucial information that is relevant to their everyday lives. The Memory Magic Application wishes to differentiate itself from other reminder applications by allowing users to speak directly with the application to set reminders for themselves and to ask questions about information they may have forgotten. Thus, the purpose of the NLU module is to process the user's speech to determine how the Memory Magic Application can best assist the user immediately, before they forget what they were thinking.

## **1.3 Vision Statement**

The vision for this project is to build and integrate an NLU module into the Memory Magic Application that transforms natural human dialog inputted from the user into structured information that the application can understand and use for its features. In so doing, individuals suffering from STM loss will possess the ability to recover from lapses in their memory to live more fulfilling lives.

## 2. Project Overview

### 2.1 Project Objectives

The primary objective of this project is to develop an NLU module that will be seamlessly integrated into the Memory Magic Application. This module will interpret the user's speech to derive information that can be used by the application's features. The addition of the NLU module will eliminate the need for the user to write notes and reminders for themselves, such as other similar applications. Once accomplished, the NLU module will allow the user to simply tell the application to set reminders and it will remind the user at specific times to accomplish some action. Furthermore, the user will be able to ask the application a question and the application will provide a response by searching the reminders to see if an answer can be given. These responses will be spoken back to the user.

### 2.2 Project Scope

This project entails the development of the NLU module for the Memory Magic Application, which will be integrated within the Memory Magic Application towards the end of this project's software development life-cycle (SDLC). The Memory Magic Application and its user interface (UI) are being developed by a separate project team, Team Mesmerize. As such, the scope of this particular project, and thus this document, is limited to the development of the NLU module and its integration with the Memory Magic Application and does not include the development of the Memory Magic Application.

### 2.3 Major Stakeholders

A stakeholder is any person, group, or organization that has a stake in the success of a given project. The project's major stakeholders are the ones who have the most to gain and the most to lose from the project's outcome. Thus, it is important that we identify who the major stakeholders are for this project, since they will be the ones from whom our development team shall be gathering the project's system requirements. It is for these people, that the various deliverables mentioned in section 4.7 of this document are intended. Also, it is their approval that will be sought to determine if the project has sufficiently satisfied those requirements. The major stakeholders for this project are:

- The NLU Module Development Team ("The Tongue Twisters")
- The Memory Magic Application Development Team ("Team Mesmerize")
- Dr. Mir Assadullah & University Maryland Global Campus (UMGC) assistant staff
- The customers (i.e., users) of the finished Memory Magic Application
- Endalkachew Girma (Overall Project Manager)

### 2.4 Project Deadline

The deadline for this project has been set for the date of **November 2nd, 2021**. This means that the NLU module must be planned, designed, developed, tested, and integrated into the Memory Magic Application no later than this date.

## 2.5 Acceptance Criteria

Acceptance criteria refers to a set of standards that are used to ensure the successful completion of a specific task, requirement, or deliverable. There are three parties, with their own acceptance criteria, who will be consulted to verify that the project is on track towards a successful conclusion.

The first of these parties is Dr. Mir Assadullah and his assistant staff at UMGC. All of the deliverables mentioned in section 4.7 of this document shall be submitted to Dr. Mir Assadullah or his assistant staff for his approval. He will grade them based on his own professional acceptance criteria. Additionally, he will provide feedback as to what changes may need to be made to the deliverables so that the project progresses in the right direction.

The second party is the project manager. The project manager will check all work and documentation completed by the project team to ensure that the work is correctly fulfilling the project's requirements. Furthermore, the project manager will formulate well-written user stories for all software development tasks so that the software developers are developing the system according to their and Dr. Mir Assadullah's vision for the project. It is also their duty to enforce the adherence to proper software development best practices. Finally, the project manager shall be the one to submit the deliverables to Dr. Mir Assadullah. In so doing, they will have the opportunity to consult with him or his assistant staff to obtain feedback. This feedback shall be shared with the project team so that both parties are in agreement as to the project's requirements moving forward.

The third party is the customers themselves. Ultimately, it is the customers who determine whether the project is successful. Towards the end of the project's SDLC, the project will undergo acceptance testing, where customers will have a chance to use the Memory Magic Application and experiment with its features, including those offered by the NLU module. Their feedback will ultimately determine if the project has been successful and if any changes need to be made based on their own internal acceptance criteria.

## 2.6 Constraints and Assumptions

### 2.6.1 Constraints

The constraints associated with this project are:

- The software developers shall use the Dart programming language to develop the NLU module's functional code. The Memory Magic Application is being developed with this programming language and so it shall remain the same for the NLU module to ensure greater compatibility, ease of communication, and sharing of technical knowledge between the two respective development teams.
- The Magic Memory Application is being developed for the Android and iOS mobile operating systems. Therefore, the NLU module must operate on both mobile operating systems as well.
- The deadline for this project is November 2<sup>nd</sup>, 2021. Therefore, the project must be completed by a date that does not exceed this date.

- No budget has yet been set for this project. Nonetheless, all project-related costs must be documented and accounted for. It's important to note that development costs such as hardware, software, and services may add to the budget and should be taken into consideration. A budget may be established at a later date during the project's SDLC. Thus, it is important that costs are kept to a minimum to avoid exceeding a lower-than-expected budget at a later time.
- The project's SDLC consists of four distinct phases, each with their own timeframe and set of deliverables that will be submitted to Dr. Mir Abdullah for his approval. These deliverables must be completed before the end of each respective phase (see section 4.7).

### **2.6.2 Assumptions**

The assumptions associated with this project are:

- The Tongue Twisters project team has an adequate number of team members and experience needed to complete the NLU module project successfully.
- Should the Tongue Twisters project team lack the expertise needed to complete the project, they will be able to find assistance or learn what they need within a timely manner as to not interfere with the project's schedule.
- Each member of the project team will have access to adequate hardware and software resources to complete their respective work on the NLU module. Additional resources that must be purchased, will be included in the project's development costs. The project plan will then be updated to reflect these changes.
- Communication between the Tongue Twisters project team and Team Mesmerize will be frequent and adequate enough to ensure that integration of the NLU module with the Memory Magic Application will be successful and on schedule.
- The project manager will hold each of the team's members accountable for their respective work and ensure the project progresses in a timely manner and adheres to the systems requirements.



## 2.7 Acronyms and Abbreviations

Table 1 below contains a list of acronyms and abbreviations that are used throughout this document, paired with a description to improve the reader's understanding.

Table 1

### *Acronyms and Abbreviations*

Acronym / Abbreviation	Description
CI/CD	<b>Continuous Integration and Continuous Deployment</b> ; an automated process for building, testing, securing, and deploying software applications.
DevSecOps	<b>Development, Security and Operations</b> ; the integration of security in all phases of software development.
NLP/NLU	<b>Natural Language Processing/Understanding</b> ; the use of artificial intelligence to process common human speech into data that can be used by a computer system.
RACI	<b>Responsible, Accountable, Consulted, Informed</b> ; a RACI matrix assigns levels of responsibility for the completion of tasks to all team members.
SDLC	<b>Software Development Life Cycle</b> ; a process for planning, designing, developing, testing, and deploying a software system.
STM	<b>Short-Term Memory</b> ; the capacity to store a small amount of information for immediate use (generally 15 to 30 seconds).
SQL	<b>Structured Query Language</b> ; a standardized language used to store, modify, and access data stored in a relational database.
UI	<b>User Interface</b> ; the medium by which the user interacts with a computer system.
UMGC	<b>University of Maryland Global Campus</b> ; the organization that Dr. Mir Assadullah and his assistants work for.

## 3. Risk Analysis

There are several potential risks that are associated with a project such as this, which can be identified by undergoing a risk analysis. It's important to identify these risks, so that they may be avoided during the project's SDLC. Conducting a risk analysis is an essential part of good project management and ensures the project is delivered on time, within budget, and to the quality expected of the project's stakeholders (Nabawy & Khodeir, 2020).

### 3.1 Risk Matrix

To simplify the risk analysis process, a risk matrix was created (see Table 2) to determine a risk's level of importance by combining its probability and severity of occurrence. The risk matrix puts the probability that a risk may occur on the y-axis of the table with a range of Very Unlikely (1) to Very Likely (5). Likewise, the severity of the risk is placed on the x-axis of the table and has a range of Negligible (1) to Severe (5). A risk's level of importance can be calculated by multiplying the risk's probability rating with its severity rating. The higher the product of the two, the more important that risk is and thus requires more attention to mitigate it.

**Table 2**

***Risk Matrix***

		Severity				
		Negligible 1	Minor 2	Moderate 3	Significant 4	Severe 5
Probability	Very Likely 5	Moderate 5	High 10	Extreme 15	Extreme 20	Extreme 25
	Likely 4	Moderate 4	High 8	High 12	Extreme 16	Extreme 20
	Possible 3	Low 3	Moderate 6	High 9	High 12	Extreme 15
	Unlikely 2	Low 2	Moderate 4	Moderate 6	High 8	High 10
	Very Unlikely 1	Low 1	Low 2	Low 3	Moderate 4	Moderate 5

### 3.2 Risk Register

The project team has undergone a risk analysis and was able to identify nine distinct risks. These risks were placed into a risk register (see Table 3), which provides descriptions of these risks, their probability ratings, their severity ratings, their level of importance, the strategies that shall be used to mitigate them, and which team members are responsible for implementing these mitigation strategies. The identified risks in the risk register have been ordered from those of lowest importance to highest importance.

**Table 3**

***Risk Register***

Risk Description	Probability Rating	Severity Rating	Level of Importance	Mitigation Strategy	Risk Owner
Team members misunderstand the project's objectives and tasks.	Very Unlikely 1	Moderate 3	Low 3	Frequent team meetings shall be held to ensure everyone on the project team has a unified vision for the project.	Project Manager
The scope of the project increases overtime (i.e., scope creep).	Unlikely 2	Minor 2	Moderate 4	The scope shall be properly defined in this project plan so that additional, unnecessary features are not added to the project.	Project Manager
Insufficient software and hardware resources are provided to complete tasks.	Very Unlikely 1	Significant 4	Moderate 4	Open-source software shall be used so that it can be accessed by all team members without the need for payment.	Project Manager
Tasks are not completed on time or milestones are not reached on their projected date of completion.	Unlikely 2	Moderate 3	Moderate 6	The work schedule for the project shall be continuously monitored to ensure that team members are completing their tasks within their allotted time frame.	Project Manager
Team members become unavailable (e.g., sickness) and no one is left to complete their tasks.	Possible 3	Minor 2	Moderate 6	More than one team member shall be scheduled to work on tasks so that task completion is not dependent on a single team member's availability.	Project Manager
Insufficient technical knowledge to complete tasks.	Unlikely 2	Significant 4	High 8	The strengths and weaknesses of each team member shall be identified so that tasks are assigned appropriately.	Software Developers & SQL Database Engineer

Tasks are completed hastily due to time constraints and do not meet acceptance criteria.	Possible 3	Significant 4	High 12	The completion of all tasks shall be approved by the project manager to verify their quality.	Project Manager
NLU module is unable to properly understand or process the user's request.	Possible 3	Severe 5	Extreme 15	The NLU module shall be thoroughly tested to ensure that it functions properly across a wide range of use cases.	Software Developers & Software Tester
There are difficulties integrating NLU module with the Memory Magic Application.	Possible 3	Severe 5	Extreme 15	Meetings shall be held frequently between the two project teams to share technical knowledge. Furthermore, documentation shall be shared amongst teams including architectural diagrams.	Software Developers

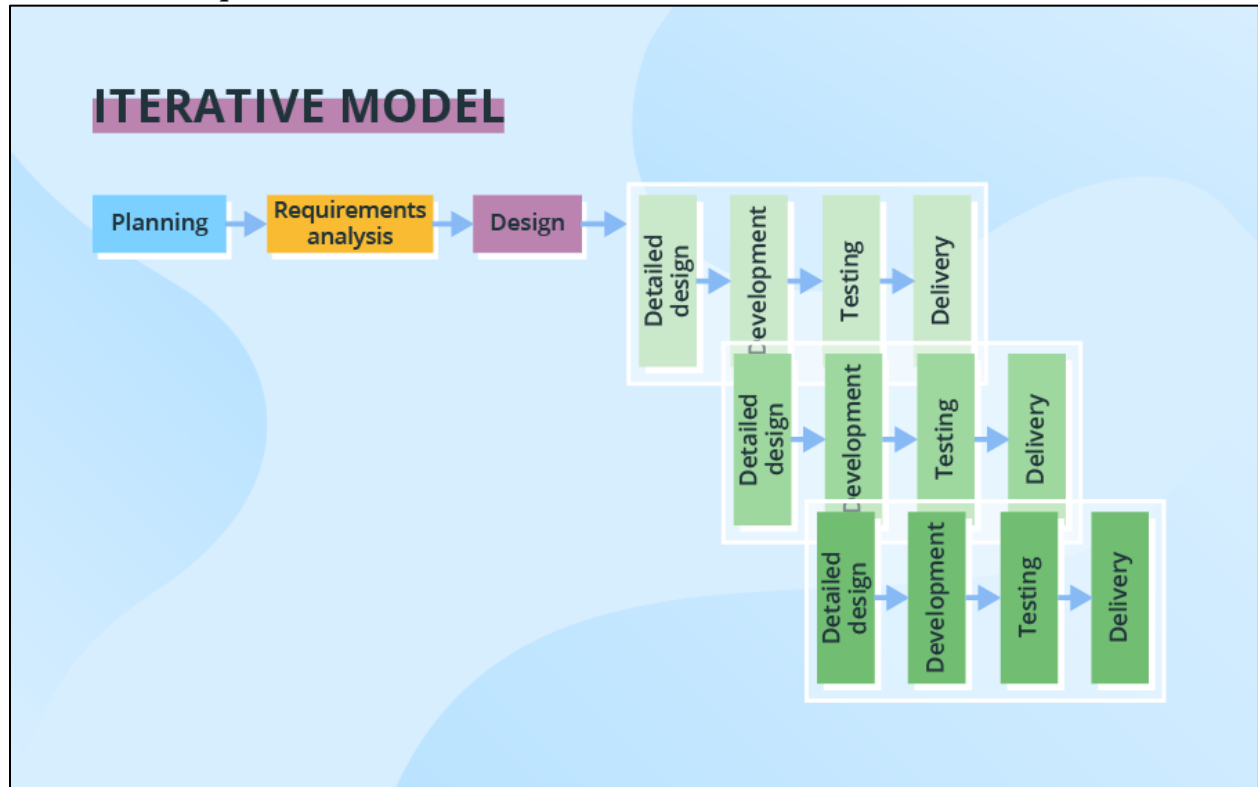
## 4. Organization of the Project

### 4.1 Software Development Methodology

The software development methodology that will be used for this project is our own derivation of the iterative development model (see Figure 1). With iterative development, the system is developed in cycles, or iterations. Each iteration is built on the previous one, so that the system's design remains relatively the same throughout the project's SDLC (Pedamkar, n.d.). Essential to this methodology, is the prioritization of the system's requirements so that earlier iterations develop the most essential requirements first, while less important requirements are developed in later iterations. This process ensures the final product has at minimum the highest-priority requirements developed. Furthermore, this methodology does allow for requirements to be changed slightly in between iterations, although not too much as to change the overall project's design. It is important to note that some of the phases and their integral steps have been altered from the original model to better fit this project's specific needs.

Figure 1

*Iterative Development Model*



Our rendition of the iterative development model consists of four distinct phases:

#### 4.1.1 Requirements Planning Phase

The first phase of our iterative development model is the requirements planning phase. During this phase, the project team is assembled and defines the project's scope, objectives, system requirements, and work schedule. Additionally, a risk analysis is done to identify potential risks associated with the project that may hinder the project's success. This phase is a combination of the planning and requirements analysis phases of the traditional waterfall SDLC.

#### 4.1.2 Design Phase

The second phase is the design phase. During this phase, the architecture of the system is agreed upon including programming languages, third-party services, data structures, and data storage. The system's architecture is illustrated in a work flow diagram showing how data moves throughout the system and how the user interacts with it. From this architectural model, a prioritized product backlog of system requirements is created that lists detailed software development actions that need to be completed to satisfy the requirements established in the previous phase.

### 4.1.3 Iterative Development Phase

The third phase is the iterative development phase. During this phase, the system is developed in iterative cycles. At the beginning of each cycle, the requirements in the product backlog are prioritized so that the highest-priority requirements are identified for that cycle's development period. These high-priority requirements are then developed during the week-long iterative cycle. At the end of the week, tests are created and executed to ensure all of the code functions correctly. Each subsequent cycle focuses on the requirements of the next level of prioritization until all of the requirements have been developed. This process ensures that the most important requirements are developed first, while less important requirements are developed last.

### 4.1.4 Testing & Deployment Phase

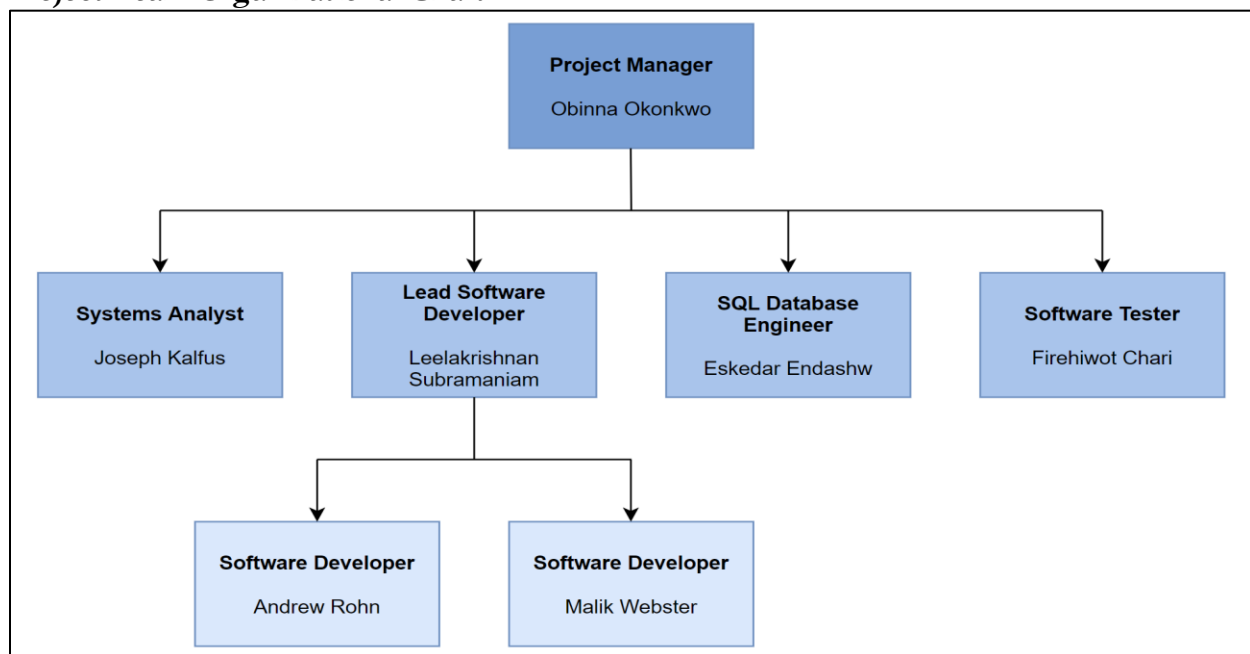
The fourth and final phase is the testing and deployment phase. At this point the system has been developed and should be ready for deployment. However, the system needs to undergo thorough unit, end-to-end, and integration testing to ensure there are no bugs. Furthermore, acceptance testing with the project's stakeholders needs to be done to ensure that they are satisfied with the final product. If all is well, the project is ready to be deployed.

## 4.2 Project Team

In order to complete this project, we have assembled a team of seven highly-competent experts including a project manager, a systems analyst, three software developers, an SQL database engineer, and a software tester. The team was assembled on August 21st, 2021 and has already started work on the project. Below is the organizational structure of the project team.

**Figure 2**

***Project Team Organizational Chart***



#### 4.2.1 Roles & Responsibilities

The following table shows the team roles, which team members are assigned to them, and what their responsibilities are.

**Table 4**

***Team Roles & Responsibilities***

Team Role	Team Member(s)	Responsibilities
Project Manager	Obinna Okonkwo	Oversees the project's progression and ensures timely completion of tasks. Facilitates communications within the team and with clients. Manages the project's work schedule. Ensures deliverables are adequate for release to clients.
Systems Analyst	Joseph Kalfus	Communicates with clients to determine the system's functional and non-functional requirements. Designs the system's architecture and writes technical documentation for the project.
Lead Software Developer	Leelakrishnan Subramaniam	Identifies software solutions for system requirements. Develops the software the system uses. Ensures good development practices such as code reviews, refactoring, and unit testing. Explains technical concepts to project team.
Software Developer	Andrew Rohn & Malik Webster	Works with the lead software developer to develop software solutions. Learns and uses software development best practices.
SQL Database Engineer	Eskedar Endashw	Designs, creates, and manages the system's database. Ensures data is well organized, secure, and accessible only to those who need it.
Software Tester	Firehiwot Chari Leelakrishnan Subramaniam	Creates and executes unit tests, end-to-end tests, and integrations tests to ensure the system functions properly. Documents the passage and failure of individual tests. Identifies potential problems for the software developers to fix.

#### 4.2.2 Project Tools

The project team will utilize a variety of tools in order to communicate and coordinate development actions. The first of these tools is Microsoft Teams. The project team will use Microsoft Teams in order to maintain communication with each other. They will be able to text message, call, and video chat with each other as need be. This is important because the project team members are working remotely from across the globe and thus all communication will have to be via telecommunications. Weekly standup meetings will be hosted on Microsoft Teams, where the team will reflect on the work that has been done over the week prior, what work needs to be done during the following weeks, and what areas of weakness can be improved.

Another tool that team will use is GitHub. All software code written by the project team will be uploaded to a single GitHub repository, where the code can be accessed by any team member at any time. Code that is uploaded to the GitHub repository will be verified by other team members before it is merged into the main branch. GitHub also includes a Kanban board, where development tasks can be grouped into sections that correspond with the tasks level of completion (e.g., not started, in progress, or completed). The Kanban board will serve as the project's product backlog. The project's requirements will be added to the Kanban board so that all team members can see which tasks need to be completed, which ones are being developed, who is developing them, and if they have been completed.

Finally, work on the project will be uploaded to a continuous integration and continuous deployment (CI/CD) pipeline setup by the development, security, and operations (DevSecOps) team. This pipeline will ensure all code being pushed up has adequate test coverage, does not contain security vulnerabilities, does not contain outdated or unused dependencies, and that the code is formatted in a clean manner.



### 4.3 Stakeholder Communication Project Plan

It is important that communication is maintained with the project's various stakeholders so that expectations are shared. The stakeholder communication project plan (Table 5) identifies how communication will be maintained, with whom, at what frequency, and what will be discussed.

**Table 5**

#### *Stakeholder Communication Project Plan*

Communication Type	Objective of Communication	Medium	Frequency	Audience	Owner	Deliverables
Kickoff Meetings	<ul style="list-style-type: none"> <li>Introduce the project team</li> <li>and the project.</li> <li>Review the project objectives and management approach.</li> </ul>	Via Microsoft Teams	Once	<ul style="list-style-type: none"> <li>Project Sponsor</li> <li>Project Teams</li> <li>Stakeholders</li> </ul>	Dr. Mir Assadullah	<ul style="list-style-type: none"> <li>Agenda</li> <li>Meeting Minutes</li> </ul>
Project Managers Team Meeting	<ul style="list-style-type: none"> <li>Discussing the status of the team at the start of the week.</li> </ul>	Via Microsoft Teams	Weekly	<ul style="list-style-type: none"> <li>Overall Project Manager</li> <li>Roy Gordon</li> <li>Robert Wilson</li> <li>Team Project Managers</li> </ul>	Overall Project Manager	<ul style="list-style-type: none"> <li>Agenda</li> <li>Meeting Minutes</li> </ul>
Project Team Meeting	<ul style="list-style-type: none"> <li>Discussing the status of the project.</li> <li>Expectation for the next milestone.</li> <li>Entrusting of roles and responsibilities.</li> </ul>	Via Microsoft Teams	Daily	<ul style="list-style-type: none"> <li>Project Team</li> </ul>	Team's Project Manager	<ul style="list-style-type: none"> <li>Agenda</li> <li>Meeting Minutes</li> <li>Roles and Responsibilities</li> </ul>

Technical Design Meeting	<ul style="list-style-type: none"> <li>Discussing the aspects of the Project's Technical Design as it relates to the Team.</li> <li>Entrusting the roles and responsibilities to be completed prior to submission</li> </ul>	Via Microsoft Teams	Daily	<ul style="list-style-type: none"> <li>Project Team</li> </ul>	Team's Project Manager	<ul style="list-style-type: none"> <li>Agenda</li> <li>Meeting Minutes</li> <li>Roles and Responsibilities</li> </ul>
Cross-Team Meeting	<ul style="list-style-type: none"> <li>Discussing the requirements between both Project Teams and the DevSecOps teams.</li> </ul>	Via Microsoft Teams	Weekly	<ul style="list-style-type: none"> <li>Project Team</li> <li>Application Development Team</li> <li>DevSecOps Team</li> <li>Overall Project Manager</li> </ul>	Overall Project Manager	<ul style="list-style-type: none"> <li>Agenda</li> <li>Meeting Minutes</li> </ul>
Milestone Approval Presentation Meeting	<ul style="list-style-type: none"> <li>Presentations performed to conclude the Milestone phase.</li> <li>Receive review and criticism for presentations.</li> <li>Introduce the next phase.</li> </ul>	Via Microsoft Teams	End of Milestone	<ul style="list-style-type: none"> <li>Project Sponsor</li> <li>Project Teams</li> <li>Stakeholders</li> </ul>	Dr. Mir Assadullah	<ul style="list-style-type: none"> <li>Agenda</li> <li>Meeting Minutes</li> </ul>

## 4.4 Scope Management Plan

It is important that as the project progresses, the scope of the project is not expanded beyond what can be accomplished within the short timeline it encompasses. It is an unfortunate trend where the number of system requirements for a project increases gradually over its SDLC. This is a common trap that many software projects fall into known as scope creep, or feature creep.

Scope creep extends the project's SDLC and diverts the project team's attention away from more important system requirements. For a project with an SDLC as short as this one, it is important that the system requirements are kept only to the absolute essentials and those that the customer is likely to utilize. Other non-essential requirements can be added in later renditions of the project. For now, it is more important to release a viable product that has the most essential requirements.

During the creation of the initial product backlog, requirements that are deemed unnecessary, impractical, or outside of the project's initial scope will be eliminated. The iterative development model that the project team is using calls for the product backlog to be re-prioritized at the beginning of each development iteration, where requirements will once again be scrutinized.

Should the project's stakeholders request new requirements be added to the project, they shall be analyzed to determine if they are compatible with the current existing scope of the project. If not, the project manager will have to explain to them that due to the lack of time, such requirements cannot be added during the project's current SDLC. Instead, they should be added in a later version of the application when the current project deadline is no longer relevant.

## 4.5 Project Schedule

In order to complete the project by the established deadline of November 2<sup>nd</sup>, 2021, the project team will need to adhere to a work schedule created by the project manager, Obinna Okonkwo, and agreed to by the individual team members. The work schedule in Figure 3, lists all the necessary tasks that will need to be completed, who has been assigned to work on them, and the time frame in which these tasks need to be done. It's important that tasks are completed within their respective allotted time frame, since most tasks are dependent on the completion of one or more prior tasks before they can be started.

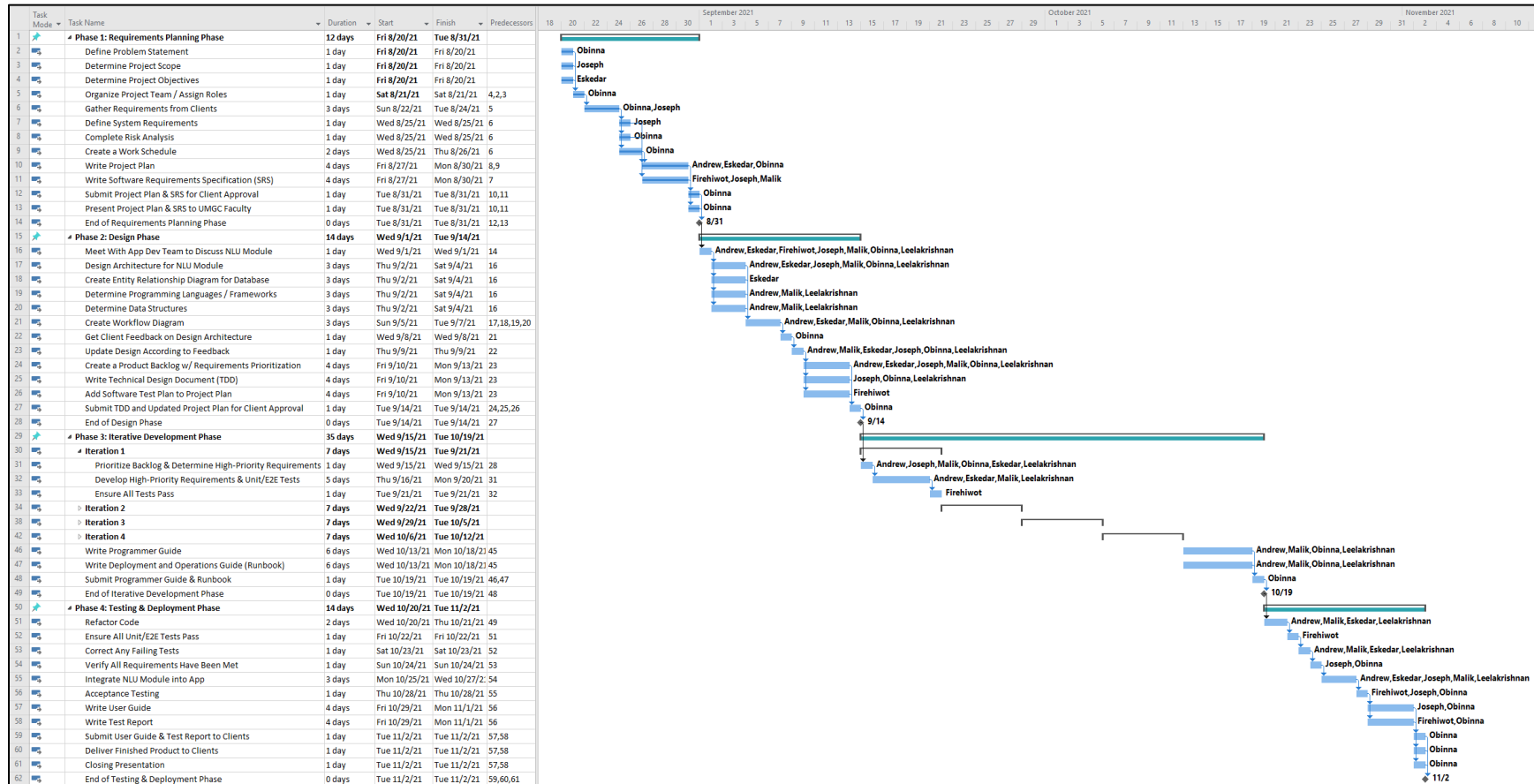
Figure 3

### Project Team Work Schedule

Task Mode	Task Name	Duration	Start	Finish	Predecessors	Resource Names
1	★ Phase 1: Requirements Planning Phase	12 days	Fri 8/20/21	Tue 8/31/21		
2	Define Problem Statement	1 day	Fri 8/20/21	Fri 8/20/21		Obinna
3	Determine Project Scope	1 day	Fri 8/20/21	Fri 8/20/21		Joseph
4	Determine Project Objectives	1 day	Fri 8/20/21	Fri 8/20/21		Eskedar
5	Organize Project Team / Assign Roles	1 day	Sat 8/21/21	Sat 8/21/21	4,2,3	Obinna
6	Gather Requirements from Clients	3 days	Sun 8/22/21	Tue 8/24/21	5	Obinna,Joseph
7	Define System Requirements	1 day	Wed 8/25/21	Wed 8/25/21	6	Joseph
8	Complete Risk Analysis	1 day	Wed 8/25/21	Wed 8/25/21	6	Obinna
9	Create a Work Schedule	2 days	Wed 8/25/21	Thu 8/26/21	6	Obinna
10	Write Project Plan	4 days	Fri 8/27/21	Mon 8/30/21	8,9	Andrew,Eskedar,Obinna
11	Write Software Requirements Specification (SRS)	4 days	Fri 8/27/21	Mon 8/30/21	7	Firehiwot,Joseph,Malik
12	Submit Project Plan & SRS for Client Approval	1 day	Tue 8/31/21	Tue 8/31/21	10,11	Obinna
13	Present Project Plan & SRS to UMG Faculty	1 day	Tue 8/31/21	Tue 8/31/21	10,11	Obinna
14	End of Requirements Planning Phase	0 days	Tue 8/31/21	Tue 8/31/21	12,13	
15	★ Phase 2: Design Phase	14 days	Wed 9/1/21	Tue 9/14/21		
16	Meet With App Dev Team to Discuss NLU Module	1 day	Wed 9/1/21	Wed 9/1/21	14	Andrew,Eskedar,Firehiwot,Joseph,Malik,Obinna,Leelakrishnan
17	Design Architecture for NLU Module	3 days	Thu 9/2/21	Sat 9/4/21	16	Andrew,Eskedar,Joseph,Malik,Obinna,Leelakrishnan
18	Create Entity Relationship Diagram for Database	3 days	Thu 9/2/21	Sat 9/4/21	16	Eskedar
19	Determine Programming Languages / Frameworks	3 days	Thu 9/2/21	Sat 9/4/21	16	Andrew,Malik,Leelakrishnan
20	Determine Data Structures	3 days	Thu 9/2/21	Sat 9/4/21	16	Andrew,Malik,Leelakrishnan
21	Create Workflow Diagram	3 days	Sun 9/5/21	Tue 9/7/21	17,18,19,20	Andrew,Eskedar,Malik,Obinna,Leelakrishnan
22	Get Client Feedback on Design Architecture	1 day	Wed 9/8/21	Wed 9/8/21	21	Obinna
23	Update Design According to Feedback	1 day	Thu 9/9/21	Thu 9/9/21	22	Andrew,Malik,Eskedar,Joseph,Obinna,Leelakrishnan
24	Create a Product Backlog w/ Requirements Prioritization	4 days	Fri 9/10/21	Mon 9/13/21	23	Andrew,Eskedar,Joseph,Malik,Obinna,Leelakrishnan
25	Write Technical Design Document (TDD)	4 days	Fri 9/10/21	Mon 9/13/21	23	Joseph,Obinna,Leelakrishnan
26	Add Software Test Plan to Project Plan	4 days	Fri 9/10/21	Mon 9/13/21	23	Firehiwot
27	Submit TDD and Updated Project Plan for Client Approval	1 day	Tue 9/14/21	Tue 9/14/21	24,25,26	Obinna
28	End of Design Phase	0 days	Tue 9/14/21	Tue 9/14/21	27	
29	★ Phase 3: Iterative Development Phase	35 days	Wed 9/15/21	Tue 10/19/21		
30	Iteration 1	7 days	Wed 9/15/21	Tue 9/21/21		
31	Prioritize Backlog & Determine High-Priority Requirements	1 day	Wed 9/15/21	Wed 9/15/21	28	Andrew,Joseph,Malik,Obinna,Eskedar,Leelakrishnan
32	Develop High-Priority Requirements & Unit/E2E Tests	5 days	Thu 9/16/21	Mon 9/20/21	31	Andrew,Eskedar,Malik,Leelakrishnan
33	Ensure All Tests Pass	1 day	Tue 9/21/21	Tue 9/21/21	32	Firehiwot
34	Iteration 2	7 days	Wed 9/22/21	Tue 9/28/21		
38	Iteration 3	7 days	Wed 9/29/21	Tue 10/5/21		
42	Iteration 4	7 days	Wed 10/6/21	Tue 10/12/21		
46	Write Programmer Guide	6 days	Wed 10/13/21	Mon 10/18/21	45	Andrew,Malik,Obinna,Leelakrishnan
47	Write Deployment and Operations Guide (Runbook)	6 days	Wed 10/13/21	Mon 10/18/21	45	Andrew,Malik,Obinna,Leelakrishnan
48	Submit Programmer Guide & Runbook	1 day	Tue 10/19/21	Tue 10/19/21	46,47	Obinna
49	End of Iterative Development Phase	0 days	Tue 10/19/21	Tue 10/19/21	48	
50	★ Phase 4: Testing & Deployment Phase	14 days	Wed 10/20/21	Tue 11/2/21		
51	Refactor Code	2 days	Wed 10/20/21	Thu 10/21/21	49	Andrew,Malik,Eskedar,Leelakrishnan
52	Ensure All Unit/E2E Tests Pass	1 day	Fri 10/22/21	Fri 10/22/21	51	Firehiwot
53	Correct Any Failing Tests	1 day	Sat 10/23/21	Sat 10/23/21	52	Andrew,Malik,Eskedar,Leelakrishnan
54	Verify All Requirements Have Been Met	1 day	Sun 10/24/21	Sun 10/24/21	53	Joseph,Obinna
55	Integrate NLU Module into App	3 days	Mon 10/25/21	Wed 10/27/21	54	Andrew,Eskedar,Joseph,Malik,Leelakrishnan
56	Acceptance Testing	1 day	Thu 10/28/21	Thu 10/28/21	55	Firehiwot,Joseph,Obinna
57	Write User Guide	4 days	Fri 10/29/21	Mon 11/1/21	56	Joseph,Obinna
58	Write Test Report	4 days	Fri 10/29/21	Mon 11/1/21	56	Firehiwot,Obinna
59	Submit User Guide & Test Report to Clients	1 day	Tue 11/2/21	Tue 11/2/21	57,58	Obinna
60	Deliver Finished Product to Clients	1 day	Tue 11/2/21	Tue 11/2/21	57,58	Obinna
61	Closing Presentation	1 day	Tue 11/2/21	Tue 11/2/21	57,58	Obinna
62	End of Testing & Deployment Phase	0 days	Tue 11/2/21	Tue 11/2/21	59,60,61	

Figure 4

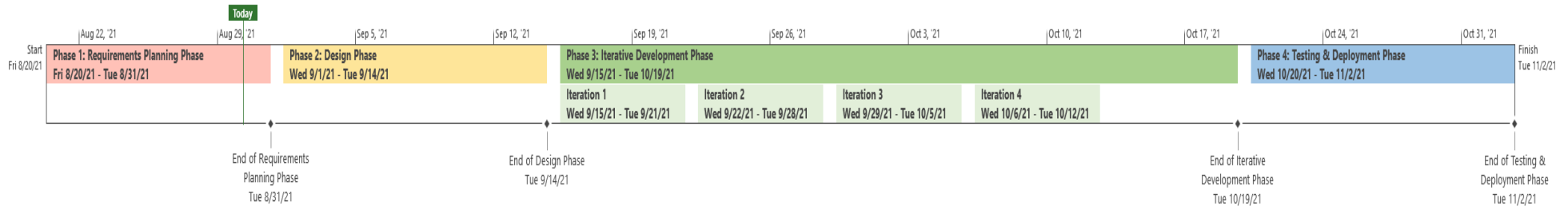
Gantt Chart



While a work schedule is useful for properly managing the project team's time, it does not show the dependencies between tasks. The above Gantt chart (Figure 4) makes it easier for the viewer to visualize the dependencies between tasks. The arrows connecting tasks indicate the order in which tasks need to be completed. Each task has a description, duration, start date, and end date (Westland, 2019). This Gantt chart will allow the project manager to monitor the project's progression from beginning to end. As you can see, there is no room for leeway, since our team is working within a short deadline. Therefore, it is important that all tasks are completed in their allotted time frame, so that the time frames for other tasks are not cut short.

**Figure 5**

***Project Timeline***



The project timeline in Figure 5 shows the four phases of this project's SDLC. The project began on August 20<sup>th</sup>, 2021 and the deadline has been set for November 2<sup>nd</sup>, 2021, meaning the project is scheduled to last 75 days in total. As was mentioned in section 4.1 of this document, our iterative development model consists of four distinct phases: a requirements planning phase, a design phase, an iterative development phase, and a testing & deployment phase. The above timeline shows the date ranges for each of these phases. Furthermore, the iterative development phase includes four, weekly iterative development cycles. Due to the short timeline for this project, our team was only able to schedule four iterative cycles. An additional week is included in this phase that does not include development, so that the team can write and publish the Programmer Guide and the Deployment and Operations Guide for this project. The start date, end date, and duration of each of the four phases are listed in the table below (Table 6).

**Table 6**

***Phases of Development***

Phase of Development	Start Date	End Date	Duration
Requirements Planning Phase	August 20 <sup>th</sup> , 2021	August 31 <sup>st</sup> , 2021	12 days
Design Phase	September 1 <sup>st</sup> , 2021	September 14 <sup>th</sup> , 2021	14 days
Iterative Development Phase	September 15 <sup>th</sup> , 2021	October 19 <sup>th</sup> , 2021	35 days
Testing & Deployment Phase	October 20 <sup>th</sup> , 2021	November 2 <sup>nd</sup> , 2021	14 work days

## 4.6 RACI Matrix

To ensure that the project team completes its tasks, they have all been assigned varying degrees of responsibility for the completion of tasks. The RACI matrix (see Table 7) displays the different levels of responsibility that all project team members have towards the completion of various tasks within the project. Each team member has been assigned one or more of the following degrees of responsibility:

**Responsible (R)** – Team member is responsible for actively completing the task. Every task needs at least one team member who is responsible for it.

**Accountable (A)** – Team member is the owner of the task and is the approval authority to determine if the task has been successfully completed. Every task needs one and only one team member to be accountable for the task's completion.

**Consulted (C)** – Team member needs to be consulted before task can be started and signed-off.

**Informed (I)** – Team member needs to be updated on the task's completion, but does not need to be consulted or contribute to the task.

Table 7

*RACI Matrix*

Task	Project Manager	Systems Analyst	Software Developers	SQL Database Engineer	Software Tester
Determine Project Scope & Objectives	A / R	R	I	I	I
Define System Requirements	R	A / R	C	C	I
Facilitate Communication w/ Stakeholders	A / R	C	I	I	I
Manage Work Schedule	A / R	C	C	C	C
Prioritize Product Backlog	A / R	C	C	C	I
Publish Documentation for Project	A	R	R	C	R

Design System Architecture	C	A / C	R	R	I
Create & Monitor User Stories	A / R	I	C	C	I
Develop System Following User Stories	A / C	I	R	R	C
Write, Execute, & Document Tests	I	I	C	C	A / R
Integrate System w/ Memory Magic Application	A / C	I	R	R	C
Acceptance Testing	A	R	I	I	I

## 4.7 Project Milestones & Deliverables

The SDLC for this project consists of four milestones that correspond with the end of each of the four phases of the SDLC. Each milestone also entails the release of deliverables to the project's stakeholders. The milestones for this project, their projected dates of completion, and their deliverables are listed in the table below (Table 8).

**Table 8**

### *Project Milestones & Deliverables*

	Significance	Projected Date of Completion	Deliverable	Description
Milestone 1	End of Requirements Planning Phase	Aug 31 <sup>st</sup> , 2021	Project Plan 1.0	Describes the purpose, scope, objectives, and organization of the project.
			Software Requirements Specification	Specifies the functional and non-functional requirements that the system must have for the project to be successful.
Milestone 2	End of Design Phase	Sep 14 <sup>th</sup> , 2021	Technical Design Document	Described the design of the system in detail including data types and data structures.
			Project Plan 2.0 w/ Software Test Plan	Details the tests and testing strategy that will be used for the project.
			Workflow Diagram	Displays how tasks are completed within the system in a logical manner.



Milestone 3	End of Iterative Development Phase	Oct 19 <sup>th</sup> , 2021	Programmer Guide	A guide written to assist future software developers who may be tasked to work on the system.
			Deployment and Operations Guide (Runbook)	Describes how the system should be deployed and operated.
			Project Plan 3.0 (if applicable)	Addition of any missing information in the previous project plan publication.
Milestone 4	End of Testing & Deployment Phase	Nov 2 <sup>nd</sup> , 2021	User Guide	Describes how users should use the system.
			Test Report	Results from executing the tests outlined in the Software Test Plan.
			Project Plan 4.0 (if applicable)	Addition of any missing information in the previous project plan publication.
			NLU Module	The successful integration of the NLU Module into the Memory Magic Application.
			Closing Presentation	The project manager's closing brief with the stakeholders summarizing the project.

## 4.8 Project Cost

Our team has completed an initial cost analysis to determine how much the project will cost. A budget for this project has not yet been provided, but it is important that should a budget be discussed, that it is equal to or greater than the estimated amount. Generally speaking, there are three types of costs that are associated with a software project: labor, equipment, and services (Rad, 2001). As of now, we have yet to determine if any equipment or services need to be paid for this project, that our team does not already have access to or can acquire free of charge. We can however calculate an approximate amount for the labor costs, which are displayed in Table 9.

**Table 9**

### *Projected Labor Costs*

Team Member	Hours Worked	Hourly Pay Rate	Labor Cost
Obinna Okonkwo	352 hours	\$64.00/hr	\$22,528.00
Joseph Kalfus	248 hours	\$35.00/hr	\$8,680.00
Leelakrishnan Subramaniam	384 hours	\$54.00/hr	\$20,736.00
Andrew Rohn	424 hours	\$43.00/hr	\$18,232.00
Malik Webster	424 hours	\$43.00/hr	\$18,232.00
Eskedar Endashw	376 hours	\$42.00/hr	\$15,792.00
Firehiwot Chari	152 hours	\$33.00/hr	\$5,016.00
<b>TOTAL</b>	2,360 hours	\$46.28/hr (average)	\$109,216.00

The project is projected to last 75 work days, including a total of 2,360 hours of combined work. The individual labor costs for each team member in Table 9 were calculated using the number of hours each team member is scheduled to work, multiplied by their hourly pay rate. Their hourly pay rate was determined by their role within the team and their level of expertise. These individual labor costs were then aggregated to calculate the total labor costs for the project, which comes out to \$109,216.00.

It's important to mention that the number of hours worked by each team member was calculated by multiplying the number of days they have been scheduled to work by eight hours per day. All team members will be paid for their work at the fixed, hourly rate listed in Table 9. Furthermore, team members will only work on the days that they are scheduled to work, meaning that not all team members will be working every day of the project's duration. On the days that they do work, they will be working a standard eight hours per day. There will be no scheduled overtime or overtime pay. Team members will not be paid for the days that they do not work.

## **5. Software Test Plan**

### **5.1 Unit Test**

#### **5.1.1 Approach**

The approach to unit tests is to feed input into a module of code and record the output. The programmer or quality assurance (QA) team would then compare those inputs and outputs to what is expected done either by hand, or with a calculator. In this scenario, we would have to find the method to black-box.

The goal of a unit test is to test simple modules ensuring certain inputs will give the correct outputs. Another goal is to ensure that errors are handled gracefully and prevent the program from crashing.

The key criteria for unit testing are to test the most important things first. As mentioned, correct output from valid input. The key criteria is not to have a goal of 100% bug free code unless it's a mission critical system where lives may be at stake or the budget and time restraints is unlimited. The goal of 100% bug free code gets exponentially more expensive as the last 1% of testing is complete. We will know when a unit test is done by testing the various valid and invalid inputs, and in arithmetic algorithms.

Since this is a black-box test, expected outputs will be tested to see if they align with what was input. For instance, we will test for saved notes, language input, and GPS location. More of this is detailed in section 5.1.2 of this document.

To determine if a unit test passes, or fails, the programmer must know what the correct output will be from a valid input. Also, any invalid input must be handled to ensure that program does not crash. If a unit test does not pass, then the programmer must figure out why it didn't pass and fix their module. Finally, once a fix has been applied, regression testing must be completed (if the unit module is integrated with other modules) to ensure that one bug fix doesn't break other code.

## 5.1.2 Proposed Test Cases

### 5.1.2.1 Test Case 1: TC-001

*Test case name:* Is query a question

*Method being tested:* questionOrStatement()

*Short Description:* Upon the Memory Magic App providing input text to the NLU Module, the NLU Module shall identify if it is a question.

*Input Data:* “Do I need to pick up something?”

*Expected Results:* “Question”

### 5.1.2.2 Test Case 2: TC-002

*Test case name:* Is query a statement

*Method being tested:* questionOrStatement()

*Short Description:* Upon the Memory Magic App providing input text to the NLU Module, the NLU Module shall identify if it is a statement

*Input Data:* “I need to pick up eggs today.”

*Expected Results:* “Statement”

### 5.1.2.3 Test Case 3: TC-003

*Test case name:* Valid query (Requirement IDs: REQ-1.3, REQ-1.4, REQ-2.12)

*Method being tested:* isValid()

*Short Description:* Upon the Memory Magic App providing input text to the NLU Module, the NLU Module shall identify if it is invalid.

*Input Data:* “Isfhew ghreuh.”

*Expected Results:* “Sorry, I do not understand what you are saying.”

### 5.1.2.4 Test Case 4: TC-004

*Test case name:* Identify pronouns and professions

*Method being tested:* whatNamePronounProfession()

*Short Description:* Upon the Memory Magic App providing valid input text to the NLU Module, the NLU Module shall identify all persons in the text like names, pronouns, and occupations.

*Input Data:* “John is a plumber, and he likes fishing.

What is the plumbers name again?

What is John’s pronoun?

What is John’s profession?”

*Expected Results:* “Name: John, Pronoun: He, Profession: Plumber”

#### **5.1.2.5 Test Case 5: TC-005**

*Test case name:* Identify locations

*Method being tested:* whereLatLong?

*Short Description:* Upon the Memory Magic App providing valid input text to the NLU Module, the NLU Module shall identify all locations in the text (e.g., cities, streets, countries, addresses, home, landmarks).

*Input Data:* 41.833046697790245, -87.7251429041244

*Expected Results:* “3348 S Pulaski Rd, Chicago, IL 60623”

#### **5.1.2.6 Test Case 6: TC-006**

*Test case name:* Identify Objects

*Method being tested:* identifyObjects()

*Short Description:* Upon the Memory Magic App providing valid input text to the NLU Module, the NLU Module shall identify all objects in the text (e.g., medication, toy, remote, banana).

*Input Data:* “Where are my house keys located?”

*Expected Results:* “Object: house keys”

#### **5.1.2.7 Test Case 7: TC-007**

*Test case name:* Identify Quantity

*Method being tested:* identifyQuantity()

*Short Description:* Upon the Memory Magic App providing valid input text to the NLU Module, the NLU Module shall identify all quantities in the text (e.g., 6, six, half a dozen).

*Input Data:* “We have two cars.

How many cars do we have?”

*Expected Results:* “Two”

#### **5.1.2.8 Test Case 8: TC-008**

*Test case name:* Identify Organizations

*Method being tested:* identifyOrganizations()

*Short Description:* Upon the Memory Magic App providing valid input text to the NLU Module, the NLU Module shall identify all organizations in the text.

*Input Data:* “I have to go to Walmart today.

What store do I have to go to?”

*Expected Results:* “Walmart”

#### **5.1.2.9 Test Case 9: TC-009**

*Test case name:* Identify Major Events

*Method being tested:* identifyMajorEvents()

*Short Description:* Upon the Memory Magic App providing valid input text to the NLU Module, the NLU Module shall identify all events in the text (e.g., anniversary, graduation, meeting).

*Input Data:* “Tim’s graduation is in one month.

What event is coming up?”

*Expected Results:* “Graduation”

#### **5.1.2.10 Test Case 10: TC-010**

*Test case name:* Identify dates

*Method being tested:* identifyDates()

*Short Description:* Upon the Memory Magic App providing valid input text to the NLU Module, the NLU Module shall identify all dates in the text (e.g., Friday, yesterday, today, tomorrow, January, January 21st, the 23rd).

*Input Data:* “What day of the week is January First, 2022?”

*Expected Results:* “Saturday”

#### **5.1.2.11 Test Case 11: TC-011**

*Test case name:* Identify times

*Method being tested:* identifyTimes()

*Short Description:* Upon the Memory Magic App providing valid input text to the NLU Module, the NLU Module shall identify all times in the text (e.g., 4:00pm, 3 in the morning, noon, evening, tonight).

*Input Data:* “My next appointment is tomorrow at noon.

When is my next appointment?”

*Expected Results:* “Tomorrow at noon.”

#### **5.1.2.12 Test Case 12: TC-012**

*Test case name:* Identify Actions

*Method being tested:* identifyActions()

*Short Description:* Upon the Memory Magic App providing valid input text to the NLU Module, the NLU Module shall identify all actions in the text (e.g., pick up, attend, go to, meet).

*Input Data:* “I need to pick up eggs”

*Expected Results:* Notes: Action: pick up

#### **5.1.2.13 Test Case 13: TC-013**

*Test case name:* Identify PII

*Method being tested:* identifyPII()

*Short Description:* Upon the Memory Magic App providing valid input text to the NLU Module, the NLU Module shall identify all PII in the text (e.g., phone number, email address).

*Input Data:* “Tom’s phone number is 706-455-1234

What is Tom’s number?”

*Expected Results:* “706-455-1234”

#### **5.1.2.14 Test Case 14: TC-014**

*Test case name:* Identify App Features

*Method being tested:* identifyAppFeatures()

*Short Description:* Upon the Memory Magic App providing valid input text to the NLU Module, the NLU Module shall identify all app features in the text.

*Input Data:* App Features: user profile, login, theme

*Expected Results:* App Features: user profile, login, theme

#### **5.1.2.15 Test Case 15: TC-015**

*Test case name:* Store note (Requirement IDs: REQ-2.13, REQ-3.1)

*Method being tested:* upsert()

*Short Description:* If a sufficient number of tokens can be parsed from input text to provide value, the NLU Module shall store tokens into a JSON object for further processing.

*Input Data:*

POST /upsert HTTP/1.1

```
{
    "inputText": description,
    "latitude": x,
    "longitude": y,
    "userId": userid,
    "isChecklist": false,
    "apptTime": epochTime,
    "noteId": noteId
}
```

*Expected Results:*

HTTP/1.1 200 OK

Content-type: application/json

```
{
    "result": "OK"
}
```

#### **5.1.2.16 Test Case 16: TC-016**

*Test case name:* Display new note

*Method being tested:* displayNewNote()

*Short Description:* Upon a new note being created, the Memory Magic App shall direct the user to the Notes screen, displaying the newly created note.

*Input Data:* Any note from Memory Magic App

*Expected Results:* Instruct Memory Magic App to display new note

#### **5.1.2.17 Test Case 17: TC-017**

*Test case name:* Is the question about Memory Magic app feature

*Method being tested:* questionOrStatementaboutMemoryMagicappfeature()

*Short Description:* Upon the Memory Magic App providing input text to the NLU Module, the NLU Module shall identify if the question is about Memory Magic app feature

*Input Data:* “How do I change my profile picture?”

*Expected Results:* “Question about Memory Magic app feature”

#### **5.1.2.18 Test Case 18: TC-018**

*Test case name:* Is the question about the last thing said

*Method being tested:* questionOrStatementaboutLastThingSaid()

*Short Description:* Upon the Memory Magic App providing input text to the NLU Module, the NLU Module shall identify the intent of the question is about the last thing said

*Input Data:* What was I talking about?

*Expected Results:* “Question”

#### **5.1.2.19 Test Case 19: TC-019**

*Test case name:* Is the question about user’s location

*Method being tested:* questionOrStatementaboutLocation()

*Short Description:* Upon the Memory Magic App providing input text to the NLU Module, the NLU Module shall identify the intent of the question is about user’s Location

*Input Data:* Why am I here?

*Expected Results:* “Location Question”

#### **5.1.2.20 Test Case 20: TC-020**

*Test case name:* Is the question about user’s note

*Method being tested:* questionOrStatementaboutNote()

*Short Description:* Upon the Memory Magic App providing input text to the NLU Module, the NLU Module shall identify the intent of the question is about user’s note

*Input Data:* Do I need to pick up eggs today?

*Expected Results:* “Question”

#### **5.1.2.21 Test Case 21: TC-021**

*Test case name:* Is the question miscellaneous

*Method being tested:* miscellaneousQuestionOrStatement()

*Short Description:* Upon the Memory Magic App providing input text to the NLU Module, the NLU Module shall identify the intent of the question is miscellaneous

*Input Data:* What is a baby goose called?

*Expected Results:* “Question”

#### **5.1.2.22 Test Case 22: TC-022**

*Test case name:* Query Memory Magic app feature

*Method being tested:* questionOrStatementaboutMemoryMagicappfeature()

*Short Description:* Upon the Memory Magic App providing input text to the NLU Module, the NLU Module shall identify if the question is about Memory Magic app feature

*Input Data:* “How do I change my profile picture?”

*Expected Results:* “Direct the user to a dedicated support page for that feature with step-by-step instructions.”

#### **5.1.2.23 Test Case 23: TC-023**

*Test case name:* Query the last thing said

*Method being tested:* questionOrStatementaboutLastThingSaid()

*Short Description:* Upon the Memory Magic App providing input text to the NLU Module, the NLU Module shall identify the intent of the question is about the last thing said

*Input Data:* What was I talking about?

*Expected Results:* “Last valid input text as output text”

#### **5.1.2.24 Test Case 24: TC-024**

*Test case name:* Query user’s location

*Method being tested:* questionOrStatementaboutLocation()

*Short Description:* Upon the Memory Magic App providing input text to the NLU Module, the NLU Module shall identify the intent of the question is about user’s Location

*Input Data:* Why am I here?

*Expected Results:* “query the GPS to determine the user’s location and return it as output text.”

#### **5.1.2.25 Test Case 25: TC-025**

*Test case name:* Query user’s note

*Method being tested:* questionOrStatementaboutNote()

*Short Description:* Upon the Memory Magic App providing input text to the NLU Module, the NLU Module shall identify the intent of the question is about user’s note

*Input Data:* Do I need to pick up eggs today?

*Expected Results:* “identify the user note whose tokenized data most closely matches the tokenized input text data (e.g., Yes, pick up eggs today)”

#### **5.1.2.26 Test Case 26: TC-026**

*Test case name:* Query user’s unavailable note

*Method being tested:* questionOrStatementaboutNote()

*Short Description:* Upon the Memory Magic App providing input text to the NLU Module, the NLU Module shall identify the intent of the question is about user’s note that is not existing

*Input Data:* Where is my car key?

*Expected Results:* “Sorry, but I could not find anything in your notes that matches your question.”

#### **5.1.2.27 Test Case 27: TC-027**

*Test case name:* Query user’s 30% to 60% matching note

*Method being tested:* questionOrStatementaboutNote()

*Short Description:* Upon the Memory Magic App providing input text to the NLU Module, the NLU Module shall identify the intent of the question is about user’s note that is 30% to 60% matching note

*Input Data:* Do I need to pick up eggs today?

*Expected Results:* “return output text asking the user for confirmation (e.g., “Are you asking about picking up eggs tomorrow?”



#### **5.1.2.28 Test Case 28: TC-028**

*Test case name:* Query user's 60% or greater matching note

*Method being tested:* questionOrStatementaboutNote()

*Short Description:* Upon the Memory Magic App providing input text to the NLU Module, the NLU Module shall identify the intent of the question is about user's note that is 60% or greater matching note

*Input Data:* Do I need to pick up eggs today?

*Expected Results:* "identify the user note whose tokenized data most closely matches the tokenized input text data (e.g., Yes, pick up eggs today)"

#### **5.1.2.29 Test Case 29: TC-029**

*Test case name:* identify the source of the answer

*Method being tested:* questionOrStatement()

*Short Description:* Upon the Memory Magic App providing input text to the NLU Module, the NLU Module shall identify the questioning method and source of the answer

*Input Data:* Where?

*Expected Results:* "Location"

#### **5.1.2.30 Test Case 30: TC-030**

*Test case name:* identify the source of the question

*Method being tested:* questionOrStatement()

*Short Description:* Upon the Memory Magic App providing input text to the NLU Module, the NLU Module shall identify the question as being miscellaneous

*Input Data:* Where is Maryland?

*Expected Results:* "NLU Module shall make a web search with the input text and return the first search result as output text. (e.g., Maryland is a state in the Mid-Atlantic region of the United States...)"

#### **5.1.2.31 Test Case 31: TC-031**

*Test case name:* identify the Language

*Method being tested:* questionOrStatementLanguage()

*Short Description:* Upon the Memory Magic App providing input text to the NLU Module, the NLU Module shall identify the language of the input text

*Input Data:* Ola

*Expected Results:* "Brazilian Portuguese"

#### **5.1.2.32 Test Case 32: TC-032**

*Test case name:* identify the Language

*Method being tested:* questionOrStatementLanguage()

*Short Description:* Upon the Memory Magic App providing input text to the NLU Module, the NLU Module shall identify the language and translate to U.S English for internal NLU processing

*Input Data:* Ola

*Expected Results:* "Hello"

### **5.1.2.33 Test Case 33: TC-033**

*Test case name:* identify the Language

*Method being tested:* questionOrStatementLanguage()

*Short Description:* Upon the Memory Magic App providing input text to the NLU Module, the NLU Module shall identify the input language is not U.S English and translate to the localization language

*Input Data:* Ola

*Expected Results:* “Ola”

### **5.1.2.34 Test Case 34: TC-034**

*Test case name:* correct output statement grammar

*Method being tested:* correctGrammar()

*Short Description:* Upon the NLU Module determining the output text that shall be returned to the Memory Magic App, it shall verify the output text is grammatically correct

*Input Data:* My son like fish.

*Expected Results:* “My son likes fish”

## **5.2 Testing Criteria**

This section lists all identified defects.

### **5.2.1 Suspension Criteria**

The suspension of testing will occur when:

- the Mesmerize App change their input or output API call format requirements
- 95% of the test cases have passed
- 50% of the test cases have failed
- it is two days before a release date deadline (to allow for the completion of integrations tests)

### **5.2.2 Resumption Criteria**

The resumption of testing will occur when:

- The requirements of the format for API calls have been set by Mesmerize app
- Enough of the codebase passes at least 50% of tests
- Another software development cycle has started to release new version

### **5.2.3 Test Completeness**

Tests will be complete when all API calls have been completed for both the NLU API and the DynamoDB API.

### **5.2.4 Test Deliverables**

The test deliverables will include this Test Plan which will match the requirements traceability matrix in the technical design document.

## 6. References

Nabawy, M., & Khodeir, L. M. (2020). Achieving efficiency in quantitative risk analysis process Application on infrastructure projects. *Ain Shams Engineering Journal*, 12(2).  
<https://doi.org/10.1016/j.asej.2020.07.032>

Westland, J. (2019, November 14). How to Create a Project Management Schedule. Retrieved August 29, 2021, from <https://www.projectmanager.com/blog/create-project-management-schedule>

Rad, P. F. (2001). *Project Estimating and Cost Management (Project Management Essential Library)*. Oakland, CA: Berrett-Koehler.

McLeod, S. (2009). Short Term Memory. Retrieved August 29, 2021, from <https://www.simplypsychology.org/short-term-memory.html>

Brown, B. (2021, July 28). The Most Common Causes of Memory Loss. Retrieved August 30, 2021, from <https://www.forbes.com/health/healthy-aging/memory-loss-causes/>

Pedamkar, P. (n.d.). Iterative Methodology. Retrieved August 28, 2021, from <https://www.educba.com/iterative-methodology/>