DREAM TEAM

TECHNOLOGY SOLUTIONS

CogniOpen
Software Application

Project Plan

Project Name: CogniOpen Software Application
Document: Project Plan (PP)
Version: 4.0
Date: November 7, 2023
Prepared By: Team A

Team Lead / Project Manager:
Vincent Galeano
Dream Team Technology Solutions (DTTS)
vgaleano1@student.umgc.edu

Client:
Dr. Mir Assadullah
University of Maryland Global Campus (UMGC)
mir.assadullah@faculty.umgc.edu

Milestone Approvals

Milestone 1 (Version 1.0)

For Approval (Team Lead / Project Manager): Vincent Galeano
Date: 09/02/2023
Signature: Vincent Galeano

For Approval (Client): Dr. Mir Assadullah
Date:
Signature:

Milestone 2 (Version 2.0)

For Approval (Team Lead / Project Manager): Vincent Galeano
Date: 09/23/2023
Signature: Vincent Galeano

For Approval (Client): Dr. Mir Assadullah
Date:
Signature:

Milestone 3 (Version 3.0)

For Approval (Team Lead / Project Manager): Vincent Galeano
Date: 10/28/2023
Signature: Vincent Galeano

For Approval (Client): Dr. Mir Assadullah
Date:
Signature:

Milestone 4 (Version 4.0)

For Approval (Team Lead / Project Manager): Vincent Galeano
Date: 11/07/2023
Signature: Vincent Galeano

For Approval (Client): Dr. Mir Assadullah
Date:
Signature:

Leads Sign-off Sheet

Key Reviewer	Version	Date	Signature
Vincent Galeano Team Lead / Project Manger	1.0	02/Sep/2023	Vincent Galeano
Kavon Johnson Lead Technical Writer	1.0	02/Sep/2023	Kavon Johnson
Zach Bowman Lead Business Analyst	1.0	02/Sep/2023	Zach Bowman
David Bright Architect / Lead Software Developer	1.0	02/Sep/2023	David Bright
Juan Torres-Chardon Lead UI/UX Designer	1.0	02/Sep/2023	Juan Torres-Chardon
Laura Hamann Lead Test Engineer	1.0	02/Sep/2023	Laura Hamann
Vincent Galeano Team Lead / Project Manger	2.0	23/Sep/2023	Vincent Galeano
Kavon Johnson Lead Technical Writer	2.0	23/Sep/2023	Kavon Johnson
Zach Bowman Lead Business Analyst	2.0	23/Sep/2023	Zach Bowman
David Bright Architect / Lead Software Developer	2.0	23/Sep/2023	David Bright
Juan Torres-Chardon Lead UI/UX Designer	2.0	23/Sep/2023	Juan Torres-Chardon
Laura Hamann Lead Test Engineer	2.0	23/Sep/2023	Laura Hamann
Vincent Galeano Team Lead / Project Manger	3.0	28/Oct/2023	Vincent Galeano
Kavon Johnson Lead Technical Writer	3.0	28/Oct/2023	Kavon Johnson
Zach Bowman Lead Business Analyst	3.0	28/Oct/2023	Zach Bowman
David Bright Architect / Lead Software Developer	3.0	28/Oct/2023	David Bright
Juan Torres-Chardon Lead UI/UX Designer	3.0	28/Oct/2023	Juan Torres-Chardon
Laura Hamann Lead Test Engineer	3.0	28/Oct/2023	Laura Hamann

Key Reviewer	Version	Date	Signature
Vincent Galeano Team Lead / Project Manger	4.0	07/Nov/2023	Vincent Galeano
Kavon Johnson Lead Technical Writer	4.0	07/Nov/2023	Kavon Johnson
Zach Bowman Lead Business Analyst	4.0	07/Nov/2023	Zach Bowman
David Bright Architect / Lead Software Developer	4.0	07/Nov/2023	David Bright
Juan Torres-Chardon Lead UI/UX Designer	4.0	07/Nov/2023	Juan Torres-Chardon
Laura Hamann Lead Test Engineer	4.0	07/Nov/2023	Laura Hamann

Revision History

Date	Version	Description	Author
25/Aug/2023	0.1	Document created	DTTS
02/Sep/2023	1.0	Milestone 1 version	DTTS
23/Sep/2023	2.0	Milestone 2 version	DTTS
28/Oct/2023	3.0	Milestone 3 version	DTTS
07/Nov/2023	4.0	Milestone 4 version	DTTS

Table of Contents

Figures & Tables

1 Introduction

Welcome to the Project Plan document for the CogniOpen Software Application. This document serves as a compass that will navigate the development journey of CogniOpen; guiding stakeholders through the intricacies of its creation. It presents a strategic overview of the project's scope, objectives, and approach.

1.1 Purpose

The purpose of this Project Plan document is to provide a comprehensive framework that defines the direction of the CogniOpen project. It aims to establish a shared understanding amongst the stakeholders (including the development team, project manager, clients, and end-users), about the project's trajectory. By offering clear guidelines in a structured format, this document will serve as the cornerstone of the development process; steering design, coding, testing, and quality assurance endeavors.

1.2 Document Organization

This document is organized into several sections, each focusing on a specific aspect of the CogniOpen software project. The structure is designed to facilitate easy navigation and reference for stakeholders who seek detailed information on areas of interest.

1.3 Project Document Suite

This Project Plan is part of a suite of project documents that collectively provide comprehensive project documentation. The suite includes:

Document	Version	Date
Project Plan (PP)	4.0	07/Nov/2023
Software Requirements Specification (SRS)	4.0	07/Nov/2023
Technical Design Document (TDD)	3.0	07/Nov/2023
Test Plan (TP)	3.0	07/Nov/2023
Programmer Guide (PG)	2.0	07/Nov/2023
Deployment and Operations Guide (Runbook)	2.0	07/Nov/2023
User Guide (UG)	1.0	07/Nov/2023
Test Report (TR)	1.0	07/Nov/2023

Table 1: Project Document Suite

Additional project documents, including the UG and TR, will be developed as the project progresses.

1.4 References

Alzheimer.gov. (n.d.). *Alzheimer's & related dementias*. https://www.alzheimers.gov/alzheimers-dementias

UMGC (2020). *Capstone Project Guide.* SWEN 670: Software Engineering Capstone, University of Maryland Global Campus (UMGC). https://learn.umgc.edu/d2l/le/content/920456/viewContent/31091351/view

UMGC (2023). *Syllabus.* SWEN 670: Software Engineering Capstone, University of Maryland Global Campus (UMGC). https://umgc.campusconcourse.com/view_syllabus?course_id=255524

UMGC (2023). *Short-Term Memory System (STeMS) Project Plan*. SWEN 670: Software Engineering Capstone, University of Maryland Global Campus (UMGC). https://umgc-cappms.azurewebsites.net/download/10364d96-c438-493a-aa3e-903fdc07fad3----AlphaSoft%20Project%20Plan%20v4.pdf

1.5 Terms, Abbreviations, & Acronyms

This section will provide a glossary of terms, abbreviations, and acronyms used throughout the Project Plan document. It aims to ensure a shared understanding of terminology among all stakeholders involved in the project.

Term	Definition
AI	Artificial Intelligence
API	Application Programming Interface
App	Application
AQL	Acceptable Quality Level
AWS	Amazon Web Services
CCB	Change Control Board
ChatGPT	Chat Generative Pre-trained Transformer
DB	Database
DevOps	Development and Operations
DTTS	Dream Team Technology Solutions
FTD	Frontotemporal dementia
iOS	iPhone Operating System
LBD	Lewy body dementia
MCI	Mild cognitive impairment
MTTD	Mean Time to Detect
QA	Quality Assurance
SDK	Software Development Kit
SDLC	Software Development Life Cycle
SRS	Software Requirements Specifications
STML	Short-term memory loss
TDD	Technical Design Document
UI	User Interface
UMGC	University of Maryland, Global Campus
UML	Unified Modeling Language
UX	User Experience
WBS	Work Breakdown Structure

Table 2: Terms, Abbreviations, & Acronyms

2 Problem Definition

The CogniOpen Software Application is conceived as a novel solution to address the challenges faced by individuals with short-term memory loss (STML). Short-term memory loss can significantly impact daily life, making it difficult for affected individuals to remember conversations, locate misplaced items, and recall important information (Alzheimer.gov, n.d.). The traditional methods of using trigger phrases or relying on caregiver assistance to retrieve information are often limited and can lead to frustration and dependency.

The primary problem addressed by the CogniOpen project is the need for a more intuitive, efficient, and independent way for individuals with short-term memory loss to access and retain essential information without the reliance on memory or external assistance. The project aims to develop an Artificial Intelligence (AI) powered mobile application that leverages cutting-edge technologies, including Amazon Web Services (AWS) services and ChatGPT, to provide an innovative solution to this problem.

Key Challenges and Limitations:

- Dependency on Trigger Phrases: Current voice-activated systems and virtual assistants rely on specific trigger phrases to initiate interactions. This approach is not always suitable for individuals with STML, as they may struggle to remember and articulate these trigger phrases accurately.
- Lack of Independence: The dependency on caregivers or family members to help locate misplaced items or recall important details can lead to a reduced sense of independence for individuals with STML.
- Complex Conversational Context: Short-term memory loss can result in confusion during conversations, making it challenging for individuals to engage meaningfully or remember discussions accurately.
- Accessibility: Existing solutions may not be designed with accessibility best practices, hindering the usability of these applications for individuals with disabilities.
- Privacy Concerns: Storing personal conversations and visual data in an application requires robust security measures to ensure users' privacy and data protection.

The CogniOpen project aims to overcome these challenges by offering a comprehensive solution that combines transcription, AI-powered analysis, and user-friendly interfaces. By providing real-time assistance, enabling conversational summaries, offering item location assistance, and promoting user independence, the CogniOpen app seeks to improve the quality of life for individuals with short-term memory loss while also alleviating the burden on caregivers.

The project's success will be measured by its ability to provide an efficient, accurate, and user-centric solution that meets the needs of individuals with STML and their caregivers. The CogniOpen app's impact will extend beyond mere technology; it has the potential to empower users, enhance communication, and foster a greater sense of autonomy for individuals who often struggle with memory-related challenges.

3 Project Assignment

This software application is intended to be hosted on mobile devices. It works in conjunction with the audio/video capabilities of the tablet or phone. Imagery and sounds are obtained from the device's camera and microphone. The content is an important surrounding of one's life.

When recall is desired, the person opens the application and asks it a question. The software will then attempt to answer it based on its database (DB) wisdom.

3.1 Goals & Objectives

CogniOpen intends to improve the lifestyle of a person with mental disabilities, for instance, the kind associated with aging. These include but are not limited to Alzheimer's disease, vascular dementia, Lewy body dementia (LBD), frontotemporal dementia (FTD), and mild cognitive impairment (MCI). Such conditions cause a person to forget details about his/her life and relationships with other people (Alzheimer.gov, n.d.).

CogniOpen draws information from a video feed to keep track of somebody's personal property and environment. The user poses simple questions via an invoked audio input mechanism. The program parses the chronicled pixels and tones and divulges a solution.

3.2 Project Scope

3.2.1 In Scope

- Project Software Application Code:
  - Baseline / Core Code (with Team B)
  - Modules:
    - Video
      - Integration with mobile device camera and microphone
      - Persistence of video across multiple sessions
    - Photo
      - Matching picture with recorded videographic material
      - Providing best guess answer in a clear and concise format
    - Gallery
- Project Documentation
  - Web site explaining how to download and install the software
  - Support material describing how to use the software's features
- Software Application (with Team B)

3.2.2 Out of Scope

- Project Software Application Code:
  - Modules:

        - Core Functionality
        - Virtual Assistant
        - Audio
- Hardware
    - Camera or microphone adjustments including troubleshooting
- Access the application (app) from anywhere except the device that has installed it. This includes web browsers, other mobile devices, desktop/laptop computers, television sets, or home appliances.
- Multiuser mode. The app operates from the perspective of a single individual.
- Intelligent suggestions or advice about what the user might consider doing with his/her information
- Troubleshooting bugs or enhancement requests with AWS AI application programming interface (API)
- Proactive discussion and iterative improvement of ChatGPT

3.3 Assumptions & Dependencies

Assumptions:

1. User can speak loud enough for the device (phone) to perceive them.
2. User has read/write access on the device. It is needed to install the app and save local data.
3. User should not be vision-impaired to the extent that they cannot make out objects presented to him/her on the screen.
4. Users are comfortable with their private information being saved on a local device.
5. Users are comfortable with their data being analyzed by remote AI providers.

Dependencies:

1. The mobile device is equipped with a camera and microphone.
2. User may provide access to camera and microphone from arbitrary applications
3. The app requires an internet connection.
4. ChatGPT must be available to the device.
5. AWS AI must be available to the device.

3.4 Deliverables

1. Project Plan
2. Software Requirements Specifications (SRS)
3. Technical Design Document (TDD)
4. Software Test Plan
5. Deployment and Operations Guide (Runbook)
6. Programmer Guide
7. User Guide
8. Test Report
9. GitHub Project (Code)
10. Mobile App (Android)

4 Project Organization

This section will reflect the organizational structure for the development of the software application, CogniOpen. This section will cover various aspects of the project development lifecycle including project management, phases, milestones and deliverables, tools, personnel, and communications. The project management model will introduce the software methodology that will be utilized to develop and manage the software application. This section will address the various phases of the software development lifecycle and the different milestones and deliverables that will be submitted throughout the project lifecycle. The many resources (software and hardware tools and staffing) that will be significant for the development of this project will be expressed in this section. Lastly, this section will detail the many forms of communication that the project team will use to meet time constraints, project costs and stakeholder expectations.

4.1 Project Management Model

The project team will implement a hybrid Agile methodology, Scrumban, to manage the software application that is being developed and maintained. This management model is the combination of Scrum and Kanban. This decision was made for the benefits of a defined organizational structure and the ability to develop in an environment with strict time constraints.

The Scrum project management model yields flexibility to the team as they can develop iteratively and incrementally. The project team will operate with shorter release cycles to consistently gather user feedback and make refinements as needed. The team will leverage the benefits of a product backlog to distribute and develop various functionalities of the system. Working in Sprints that align with Milestone deliveries will allow the team to organize and prioritize the many features of the system that need to be developed.

The Kanban project management model allows the team to organize and distribute work amongst the team with the implementation of a Kanban board. Like Scrum, this methodology gives the team flexibility with planning decisions. More importantly, it generates faster output with development as features will need to be produced at a much higher rate than normal.

4.2 Project Phases

- Planning – The team will incorporate project planning into every step of the development lifecycle. This will help the team estimate project costs (tools, materials and staffing), define a budget and establish a project schedule. Defining system requirements will also play a major part in this phase of development.

- Gathering Requirements & Analysis - This phase will mark the project team's efforts to gather user requirements and put together specification documentation for the application. The development team will then conduct a thorough analysis of the requirements to determine what tools and technologies they will implement to accomplish their deliverable goals. This stage is meant to gain a collective understanding of the requirements of the software system.

- Design - The design phase will highlight the various prototypes and solution suggestions of the development team as they attempt to deliver a product that meets stakeholder expectations. Feedback from the stakeholders on the different designs presented and documented will be essential before development can start.

- Development - This phase will encapsulate the actual development of the software application. The development team will collaborate to produce a front-end and back-end solution that meets the requirements and expectations of the stakeholders. System features will be distributed amongst the team to align with the project schedule and budget.

- Testing - This phase will co-exist with the development phase of the software project. As developers build the many features of the software, system testers will be vital in ensuring these features function as defined by user requirements. Testing will play a significant role in delivering a working product that meets the expectations of the end-user and project stakeholders.

- Implementation – After the testing phase, the software will then be ready to be deployed/release for consumption. This phase will also incorporate training and documentation that will assist users in operating the software system.

- Maintenance - This phase will determine how the software will be supported after it is released for consumption. This phase coexists with the rest of the lifecycle of the application after it has been deployed. The project team will need to establish guidance and support for the application as many things may change over time (environment, technology, staff, etc.)

4.3 Milestones & Deliverables

Milestone One
Deliverables: Project Plan and SRS document
Presentation

Milestone Two
Deliverables: Technical Design Document and Software Test Plan
Presentation

Milestone Three
Deliverables: Programmer Guide, Deployment and Operations Guide (Runbook),
Add Software Test Plan to Project Plan
Presentation

Milestone Four
Deliverables: User Guide, Test Report, Programmer Guide
Presentation

4.4 Tools

Communication, Documentation & Tracking

- Microsoft Teams 1.6.00.22378 (64-bit)
- Microsoft Word + Excel Version 2307 (Build 16626.20170)
- Microsoft SharePoint
- Microsoft Project
- Azure DevOps Wiki Pages
- Azure DevOps Board

Development Environment

- GitHub
- Argo UML v0.34
- Photo Shop 2033 Q2
- Pencil v3.1.0
- Dart v3.1.0
- Flutter v3.1.0
- Android Studio Giraffe 2022.3.1 Patch 1 with Android SDK v34.0.0
    - Compatible with Visual Studio Code 2022 v1.81.1
    - Compatible with Microsoft Visual Studio Community 2019 Version 16.11.0

4.5 Personnel

This section details the personnel associated with the CogniOpen project; it includes the personnel affiliations, roles & responsibilities, role assignments, and RACI (Responsible, Accountable, Consulted, and Informed) matrix.

4.5.1 Affiliations

Project personnel are logically grouped according to their affiliations within the project's context, which include the client, project mentors, Project Team A, and Project Team B. Affiliations are defined along with the names of associated members below.

Affiliation	Description	Name(s)
Client	Initiator of the project request, communicates objectives, defines requirements and scope, and provides feedback as the primary stakeholder	▪ Dr. Mir Assadullah
Project Mentors	Provides general project guidance, answers to guidance inquiries or issues needing clarification, and feedback	▪ Roy Gordon ▪ Rob Wilson

Affiliation	Description	Name(s)
Project Team A	Primary full-stack project team, responsible for design, development, and maintenance of the software application, and works in tandem with Project Team B	▪ Vincent Galeano ▪ Kavon Johnson ▪ Zach Bowman ▪ David Bright ▪ Juan Torres-Chardon ▪ Laura Hamann ▪ Vivek Goel ▪ Selam Biru ▪ Benjamin Sutter ▪ Richard Tsang ▪ Andrea Pellot
Project Team B	Secondary full-stack project team, responsible for design, development, and maintenance of the software application, and works in tandem with Project Team A	▪ Edward Devine ▪ Zachary Cappella ▪ Gabriel Gomes ▪ John Hamilton ▪ Malachi Jamison ▪ Sean Mirani ▪ Abebe Natea ▪ Alexis Shannon ▪ Eyerusalme Gebrehiwot

Table 3: Affiliations

4.5.2 Roles & Responsibilities

To ensure the success of the project, it is crucial to identify, assign, and leverage varied roles with distinct responsibilities. These roles and responsibilities more effectively and efficiently empower specialized skills, streamline task management, ensure quality, facilitate communication, and collectively address various project aspects. With respect to Project Team A, internal and external roles and responsibilities are detailed below.

***Internal* – Project Team A:**

Role	Responsibilities
Team Lead / Project Manager (Team A)	▪ Leads team organization and coordination efforts to achieve project goals and objectives ▪ Acts as the team representative and liaison with the client, project mentors, and Project Team B ▪ Attends the weekly project manager meeting to review status, issues / barriers, and determine next steps

Role	Responsibilities
	▪ Finalizes, compiles, and submits all project milestone documentation and materials to the client ▪ Initiates milestone presentations with a status update on activities ▪ Guides, coordinates, and supervises team Role Leads to successfully complete all team project deliverables ▪ Provides final team approval on all team deliverables
Deputy Team Lead / Project Manager	▪ Supports the Team Lead / Project Manager (Team A) with all responsibilities
Secretary	▪ Organizes all team communication standards, protocols, and resources ▪ Schedules, coordinates, and drives team meetings and agendas ▪ Keeps detailed records of all pertinent discussions, decisions, and action items ▪ Posts relevant announcements and reminders to the team as necessary
Lead Technical Writer	▪ Role Led – Guides, coordinates, and supervises team Technical Writers to successfully complete technical documentation for all team project deliverables ▪ Determine documentation strategies, styles, and formats ▪ Revise and refine documentation for accuracy, consistency, and style ▪ Provides role approval on all technical documentation for all team project deliverables ▪ All responsibilities of the Technical Writer role
Technical Writer	▪ Creates clear and user-friendly technical documentation, including project design and requirement documents as well as user, code, and API guides / manuals
Lead Business Analyst	▪ Role Led – Guides, coordinates, and supervises team Business Analysts to successfully determine the necessary business needs, requirements, and solutions for all team project deliverables

Role	Responsibilities
	▪ Engages with the client to analyze needs and use cases, and collaborates with stakeholders to ensure seamless communication and alignment across all team levels ▪ Sets the work backlog priority ▪ Provides role approval on all determined business needs, requirements, and solutions for all team project deliverables ▪ All responsibilities of the Business Analyst role
Business Analyst	▪ Documents use cases and intended functionality ▪ Creates user stories with acceptance criteria ▪ Validates use cases against business rules ▪ Understands requirements, prepares documentation, and grooms the work backlog
Architect / Lead Software Developer	▪ Role Led – Guides, coordinates, and supervises team Software Developers to successfully design and develop software for all team project deliverables ▪ Establishes the technical direction, architecture blueprints, and standards for the software solution for the project ▪ Breaks down complex software development goals and efforts into smaller tasks, sizes development efforts, and then delegates tasks the Software Developers ▪ Enforces and ensures code quality and standards ▪ Communicates business requirements, criteria and needs to the Software Developers ▪ Provides role approval on all software for all team project deliverables ▪ All responsibilities of the Software Developer role
Software Developer	▪ Develops and modifies software as per architecture blueprints and requirements ▪ Ensures adherence to standards, design patterns, and writes well-documented code ▪ Tests and debugs code in coordination with the Test Engineers to ensure software quality and functionality ▪ Deploys, monitors, and maintains software
Lead UI/UX Designer	▪ Role Led – Guides, coordinates, and supervises team UI/UX designers to successfully design the UI/UX for all team project deliverables

Role	Responsibilities
	▪ Set UX vision and strategy for the software application, aligning with business goals and user needs ▪ Organize software application content into intuitive hierarchies for easy navigation and logical interactions ▪ Conduct usability testing to make data-driven design decisions enhancing software application usability ▪ Determine and enforce modern UI/UX and accessibility best practices standards ▪ Provides role approval on all UI/UX designs for all team project deliverables ▪ All responsibilities of the UI/UX Designer role
UI/UX Designer	▪ Designs visually appealing, user-friendly interfaces ▪ Conducts user research and develops mocks and prototypes ▪ Collaborates for design consistency, user feedback incorporation, and seamless experiences. ▪ Adapts designs for diverse devices and platforms, organizing information into user-friendly layouts with visual design elements enhancing aesthetics and usability
Lead Test Engineer	▪ Role Led – Guides, coordinates, and supervises team Test Engineers to successfully test and ensure quality control for all team project deliverables ▪ Collaborate with role leads to align testing with development and project objectives ▪ Lead the development and execution of comprehensive test plans and automation ▪ Provides role approval on all Test / QA plans, tests, and test automation for all team project deliverables ▪ All responsibilities of the Test Engineer role
Test Engineer	▪ Develop and execute comprehensive test plans and automation ▪ Identify and report bugs and defects ▪ Analyze test results and provide recommendations

Role	Responsibilities
	▪ Write tests and debugs code in coordination with the Software Developers to ensure software quality and functionality

Table 4: Internal Roles and Responsibilities

External:

Role	Responsibilities
Client Representative / Primary Stakeholder	▪ Initiates the project request ▪ Communicates objectives and defines requirements and scope (from the perspective of the client) ▪ Provides feedback and approvals on project deliverables ▪ Collaborates with project team lead / project managers at milestone presentations
Project Management Mentor	▪ Provides general project guidance, answers to guidance inquiries or issues needing clarification, and feedback with respect to project management ▪ Facilitates the weekly project manager meeting to review status, issues / barriers, and determine next steps
Business Administration Mentor	▪ Provides general project guidance, answers to guidance inquiries or issues needing clarification, and feedback with respect to product design and development ▪ Acts as a secondary project stakeholder and provides feedback on project progress and direction
DevOps Mentor	▪ Provides general project guidance, answers to guidance inquiries or issues needing clarification, and feedback with respect to DevOps ▪ Assists the team with external technical integration requests ▪ Organizes team and project shared technical resources for collaboration and software development
Team Lead / Project Manager (Team B)	▪ Acts as the Project Team B representative and liaison with the client, project mentors, and Project Team A

Table 5: External Roles and Responsibilities

4.5.3 Role Assignments

Personnel role assignments are detailed below. Note that individuals may be assigned one-to-many roles. Project Team A roles are listed in priority order, with the first listed role as their primary role. Additionally, Project Team A roles and assignments may change as the project progresses to best meet its needs and ensure success.

Affiliation	Name	Role(s)
Client	Dr. Mir Assadullah	▪ Client Representative / Primary Stakeholder
Project Mentor	Roy Gordon	▪ Project Management Mentor ▪ Business Administration Mentor
Project Mentor	Rob Wilson	▪ DevOps Mentor
Project Team A	Vincent Galeano	▪ Team Lead / Project Manager (Team A) ▪ Software Developer ▪ Technical Writer
Project Team A	Kavon Johnson	▪ Deputy Team Lead / Project Manager ▪ Lead Technical Writer ▪ Team Secretary
Project Team A	Zach Bowman	▪ Lead Business Analyst ▪ Test Engineer ▪ Technical Writer
Project Team A	David Bright	▪ Architect / Lead Software Developer ▪ Technical Writer
Project Team A	Juan Torres-Chardon	▪ Lead UI/UX Designer ▪ Software Developer ▪ Technical Writer
Project Team A	Laura Hamann	▪ Lead Test Engineer ▪ Technical Writer
Project Team A	Vivek Goel	▪ Business Analyst ▪ Test Engineer ▪ Technical Writer
Project Team A	Selam Biru	▪ Business Analyst ▪ Software Developer ▪ Technical Writer
Project Team A	Benjamin Sutter	▪ Software Developer ▪ Test Engineer ▪ Technical Writer
Project Team A	Richard Tsang	▪ UI/UX Designer ▪ Software Developer

Affiliation	Name	Role(s)
		▪ Test Engineer ▪ Technical Writer
Project Team A	Andrea Pellot	▪ UI/UX Designer ▪ Software Developer ▪ Technical Writer
Project Team B	Edward Devine	▪ Team Lead / Project Manager (Team B)

Table 6: Role Assignments

4.5.4 RACI Matrix

The following RACI matrix was created to help clarify and define roles and responsibilities for the CogniOpen project. Each role in the matrix is assigned one or more of these labels to specify their level of involvement and authority in decision-making and task execution.

RACI Matrix RACI Matrix Key R Responsible A Accountable C Consulted I Informed	Client	Team Lead / Project Manager	Lead Technical Writer	Technical Writers	Lead Business Analyst	Business Analysts	Architect / Lead Software Developer	Software Developers	Lead UI/UX Designer	UI/UX Designers	Lead Test Engineer	Test Engineer
Project Plan	C	I	A	R	R	R	I		I		I	
Software Requirements Specifications (SRS)	C	I	A	R	R	R	I		I		I	
Architecture Design		I			C	C	A	R	C	I	C	I
UI/UX Design		I			C	C	C	I	A	R	C	I
Test & Automation Plan		I			C	C	C	I	C	I	A	R
Technical Design Document (TDD)	I	I	A	R	I	I	A	R	R	R	R	R
User Stories & Backlog		I			A	R	C	I	C	I	C	I
Software Application Development	I	I			C		A	R	C		C	
Software Application Test & Automation Development		I			C		R	R	C		A	R
Test Report	I	I	A	R	I	I	I	I	I	I	A	R
Software Application Deployment & Maintenance	I	A					R	R			C	C
Deployment & Operations Guide (Runbook)	I	I	A	R			R	R			C	C
Programmer Guide	I	I	A	R			R	R			R	R
User Guide	I	I	A	R	C	C			R			

Table 7: RACI Matrix

4.6 Communications

The project teams will utilize Microsoft Teams as the primary method of communication within the project teams, among the project teams, and with the customer and mentors. Because none of the people involved in the project are collocated, the success of this project hinges on rapidly and effectively communicating with one another. The Microsoft Teams product addresses this need through its various chats, channels, and tele video conferencing meetings; these functionalities will all be utilized to quickly collaborate on the project and bring all parties to the same understanding. Additionally, email will be utilized on a more formal basis to reinforce the decisions and work completed in the Teams product (i.e., sending an email reminder of an upcoming Teams meeting).

As discussed in the previous section, there are various teams utilizing the Teams application for this project. There is Team A, Team B, the customer, and the project mentors; additional teams may exist that are subsets or combinations of the above teams, but the definition of these is outside the realm of what Team A sets. The following information pertains to what Team A is doing.

Microsoft Teams allows for multiple conversations among peers along with dedicated Channels for intrapersonal communication. For this project, we have set up separate Channels for the various discussions to keep communication focused on the correct members. Currently, we have the following dedicated Channels promoted for the discussions listed below.

Channel Name	Channel Members	Topic of Discussion
General	All project members, including customer, mentors, and both teams	General comments regarding the overall course structure and the involved project
Team A	Project Members of "Team A"	General location for meeting activities and project planning for the Team A project team
Team B	Project Members of "Team B"	General location for meeting activities and project planning for the Team B project team

Table 8: Discussion Channel List

Besides the Channels, Team A also utilizes a few internal chat rooms. Each chat room is dedicated to a specific topic, and for the most part, each room includes all the members of Team A. There is one exception being the "Team A Leads Chat" which includes the leads of the various functional areas. The current chat rooms are listed in the following table.

Chat Room Name	Members	Topic of Discussion
Dream Team (Team A -- UMGC SWEN 670 Fall 2023)	Team A project members	Overall Course discussions and specific chat room for the Project Plan
Team A Leads Chat	Vincent Galeano, Kavon Johnson, Zach Bowman, Laura Hamann, Juan Torres-Chardon, David Bright	Major project initiatives and project decisions Coordination between the project areas

Chat Room Name	Members	Topic of Discussion
SRS discussion	Team A project members	Specific channel dedicated to the Software Requirements Specification deliverable
Teams A + B Chat	All project members	Forum for coordinating work scope among the project teams General discussion of overall course and project

Table 9: Chat Room List

4.6.1 Conduct

Information Distribution

Information is shared openly with all group members regardless of topic. Meetings are recorded, and the recorded videos are available for all team members after the meeting has concluded. Decisions are best made through the chat rooms, due to the speed at which information can be shared and the condensed timeframe of this project; however, items which put the team members at an impasse are tabled and discussed during the team meeting.

If the impasse is foreseen to be something that will not be settled with a meeting, the team project lead will reach out to the other team project lead for insight and clarification on the situation. If an answer still cannot be found, the issue is escalated to the course instructor and the project mentors for priority resolution.

Lastly, generally, messages can be marked as “Important!” to stand out to all members in the list. Conversely, when singling out a single person, the “@mention” feature is helpful for attracting the person’s attention and delivering the information directly.

Subject Moderation

In the specific chats on a deliverable, the team members are to keep discussion to that topic. A chat about the technical design document should not be asking questions about where the peer review form (that is in the Project Overview/General discussion). The Lead for that deliverable is responsible for moderating the chat and corralling the members to discuss the topic at hand. For example, in the SRS chat, the Lead Business Analyst for the team should minimize any inappropriate discussion about the Project Plan that does not pertain to the software requirements.

4.6.2 Meetings

As previously stated, the success of this project hinges on the ability to synchronize clearly and quickly everyone’s understanding of the project and their role in the project. A challenge to this goal is the asynchronous and non-collocated nature of our project members. Weekly tele video conference meetings will be held to overcome this challenge. Again, Microsoft Teams supports this feature; meetings will be created for routine synchronization sessions, and invitations sent to the prospective members. The current schedule for these meetings can be seen in the following table.

Meeting Name	Meeting Time and Schedule	Purpose of the Meeting
Team A Weekly Sync	Wednesday evenings; 8PM-9PM EST (Eastern Standard Time)	Discuss any open items not settled in the Team A chat
Team A Sync (follow-up)	Per needed basis	A follow-up meeting to the weekly sync meeting for those unable to attend
Project Team Weekly Sync	Monday evenings; 7PM-8PM EST	Weekly sync with the other project team to discuss challenges, obstacles, and plans moving forward
Project Leads Weekly Meeting	Monday evenings; 8PM-8:30PM EST	Weekly sync with the project leads and the project mentors to discuss pain points, methods to overcome, and request advice
Milestone Kick-off	Same time as the Team A Weekly Sync	Special meeting at the start of the new Milestone to identity the deliverables for the current Milestone and discuss role assignments to complete current Milestone deliverables

Table 10: Meeting Schedule

These meetings must follow certain guidelines to maximize effectiveness. To begin, when scheduling a meeting, the meeting organizer must provide an agenda for the meeting. This agenda will enumerate the topics to discuss in a bulleted format; this will communicate to the attendees the items that will be addressed in this meeting to keep the conversation limited to the topics at hand. If an item is missing, an attendee may send a message to the organizer to add the item, or as needed, schedule a separate meeting to address that topic.

General meeting etiquette applies to our meetings. Attendees should arrive at the meeting promptly at the scheduled time to not waste the other’s time. Attendees not speaking are to remain muted; when an attendee wants to speak and another is talking, they are to use to the “raise hand” feature to gather the attention of the speaker; the speaker will then address the person with the raised hand when yielding the floor. All meetings are to be recorded for the invitees not in attendance; permission is requested to record (and attendance implies consent to being recorded). Attendees are not required to have their web cameras on, but it is encouraged to facilitate team cohesiveness. Discussion is encouraged to be lively, with not one person or group of people dominating the discussion; however, discussion shall be limited to the topics on the agenda, and any deviation from this list shall be corrected by the meeting organizer.

4.6.3 Email

Email communication will be supplemental to the Microsoft Teams communication. Emails are reserved for the formal communications of questions and ideas. For instance, scheduling a meeting will begin in a Teams chat to quickly determine everyone's availability; the meeting will then be scheduled with an email reminder sent out to formally declare the scheduled meeting time (that was pre-arranged in the Teams chat). Alternatively, questions to the customer will be sent via email. The response is expected via email and disseminated to the team via email as well. As a fail-safe, this information will also be summarized and disseminated to the team via a Teams chat for the quickest dispersal of the information.

4.6.4 Azure DevOps Backlog

Work item tracking for this project will be completed using the Azure DevOps project tasks available in the Azure DevOps Team backlog board[1]. The general workflow for work items in this backlog is to follow a Kanban method of "to do", "in progress" and "done". Items will be created in the project view and added to the backlog. From there, they will be prioritized, scheduled, and assigned by the project leads to the team members completing the work. Development items can be converted to "issues" and tracked/completed with the GitHub issue tracking. Completed issues are to be auto closed with the commit.

Newly created items are to follow a general format. To begin, it is imperative that team members search to see if the work item or issue already exists before duplicating the work item. When creating a work item, a title for the work item is required. Administratively, a description of the work item must be provided. This work item description must contain within it a "definition of done", also known as acceptance criteria; this detail will explain what the work product must be to close the work item and accomplish the intended goal. The "assigned to" and "status" fields can be populated during backlog grooming, and they may change over the work item lifecycle. If the work item is a development item, it can be converted easily into an Issue.

Every code commit must be tied to a work item issue. When committing code to the GitHub repository, developers need to clearly communicate with their peers what is being committed and for which issue. To do this effectively, developers must follow proper etiquette. The following information provides the best practices guidelines that are expected to be followed for this project.

- Use imperative commands at the beginning of your commit (*e.g.*, fix, refactor, add, remove, etc.).

[1] Azure DevOps Team - https://dev.azure.com/ProjectCogniOpen/CogniOpen/_boards/board/t/CogniOpen%20Team%20A/Stories

- A commit message should not exceed 50 characters
  - What work did you complete?
  - Why did you do it?
  - What effect does it have?
- Commit the smallest amount of code for a ticket/issue
  - Do not combine changes for tickets unless necessary
  - Changes should be as small as possible to make reverting the code base to a previous state less problematic
- Closing issues using `commit` comment
  - Special syntax of "`Closes #issue_number`" or "`Resolves #issue_number`"
  - Issue will be auto closed when the pull request has been merged to main branch
  - This effort will resolve the anti-pattern of leaving issues open when they have in fact been solved
  - Reduces duplicate efforts from other developers solving a non-issue

4.6.5 Approvals

In adherence to our project methodology, all work is initiated only upon obtaining approval from the customer. Work does not begin without approval from the customer. This approval process is meticulously followed to ensure alignment with customer expectations and project goals. For instance, the project was initiated by a meeting with the customer where it was requested that a Project Plan be created along with a Software Requirements Specification document. This meeting was verbal approval and the catalyst for the effort.

Upon the delivery of this Project Plan, we will present additional Milestones to the customer. The commencement of work on these milestones is contingent on the customer's explicit approval. The Project Plan and subsequent deliverables will be presented to the customer via the UMGC assignment submission portal. The customer will then approve (or possibly disapprove) with feedback on what needs to be modified in the deliverable for further work to continue.

The project team will only work on the agreed upon functionalities presented in the Software Requirements Specification document. This approach ensures that the project scope remains aligned with customer expectations and minimizes scope creep. If the project team is to take on any more work, or an existing agreed-upon functionality needs to be modified, such changes must follow the Change Management policies defined in Section 5.4 Change Management. By meticulously adhering to this customer-centric approval process, we ensure that our project remains responsive, aligned, and attuned to the evolving needs and expectations of our valued customers.

5 Work Management

5.1 Requirements

5.1.1 Project Methodology Requirements

The project shall adopt the Scrum methodology for the development of the CogniOpen software application. The Agile Scrumban methodology was chosen due to its iterative and incremental nature, resulting in better adaptability during development. For more details, refer to Section 4.1 “Project Management Model”.

5.1.2 Task Tracking and Management Requirements

The project shall utilize a project management tool, such as Jira, to track and manage tasks. These tasks will be categorized by priority, complexity, and phase. Each task will include a description, owner, and due date. The project manager will keep track of task progress and potential bottlenecks throughout the project.

5.1.3 Communication Requirements

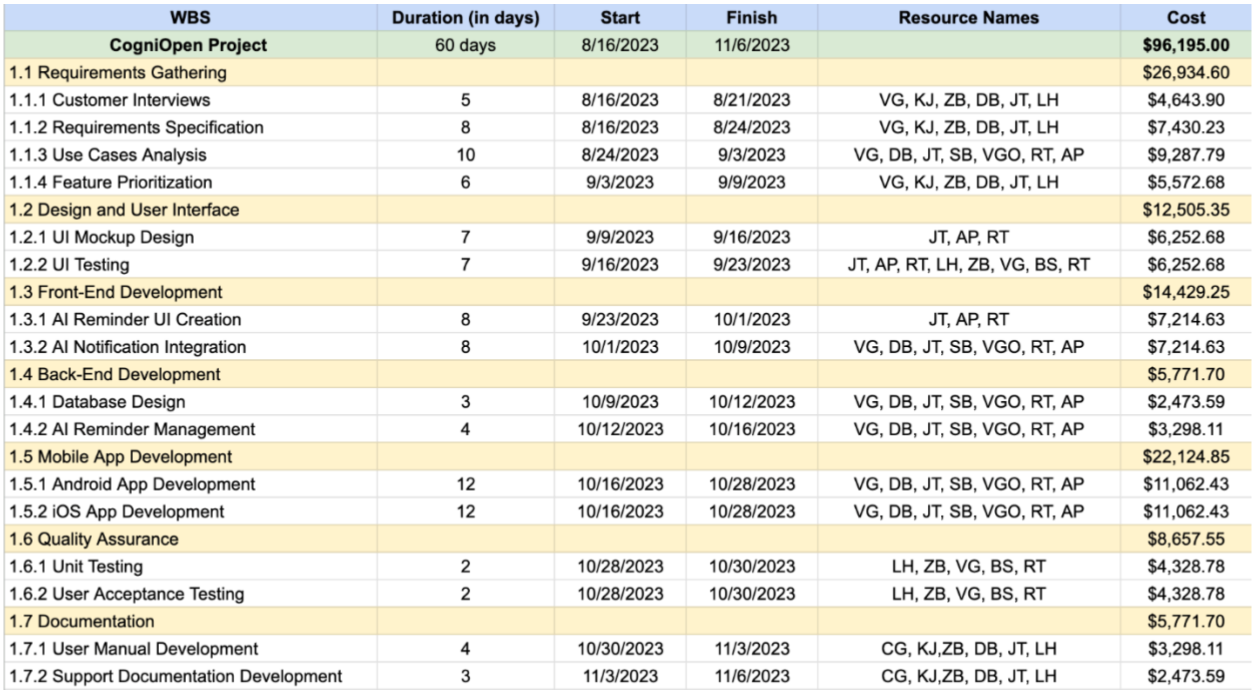
A weekly status meeting will be held every Wednesday at 8 p.m. EST to discuss project progress, challenges, and upcoming tasks. Microsoft Teams will be used to share day-to-day updates, task issue resolution, and information sharing amongst team members. For more details, refer to Section 4.6 “Communications” on the topic of communication management.

5.1.4 Risk Management Requirements

Potential risks will be identified and assessed during project initiation. These risks will then be reviewed thoroughly throughout the development lifecycle. A risk register will detail identified risks, potential impact, probability, and mitigation strategies. High priority risks will be brought to the project leadership’s attention. The team will ensure risk assessment is integrated throughout the project’s development. For more details, refer to section 10.2 Risk Management of this document.

5.2 Work Breakdown Structure (WBS)

This section outlines the fundamental sections of project components through the Work Breakdown Structure. It will serve to guide project planning and address progress, challenges, and requirements. It provides the hierarchical structure of major and minor phases, providing a comprehensive framework for project management and execution.



WBS	Duration (in days)	Start	Finish	Resource Names	Cost
CogniOpen Project	60 days	8/16/2023	11/6/2023		**$96,195.00**
1.1 Requirements Gathering					$26,934.60
1.1.1 Customer Interviews	5	8/16/2023	8/21/2023	VG, KJ, ZB, DB, JT, LH	$4,643.90
1.1.2 Requirements Specification	8	8/16/2023	8/24/2023	VG, KJ, ZB, DB, JT, LH	$7,430.23
1.1.3 Use Cases Analysis	10	8/24/2023	9/3/2023	VG, DB, JT, SB, VGO, RT, AP	$9,287.79
1.1.4 Feature Prioritization	6	9/3/2023	9/9/2023	VG, KJ, ZB, DB, JT, LH	$5,572.68
1.2 Design and User Interface					$12,505.35
1.2.1 UI Mockup Design	7	9/9/2023	9/16/2023	JT, AP, RT	$6,252.68
1.2.2 UI Testing	7	9/16/2023	9/23/2023	JT, AP, RT, LH, ZB, VG, BS, RT	$6,252.68
1.3 Front-End Development					$14,429.25
1.3.1 AI Reminder UI Creation	8	9/23/2023	10/1/2023	JT, AP, RT	$7,214.63
1.3.2 AI Notification Integration	8	10/1/2023	10/9/2023	VG, DB, JT, SB, VGO, RT, AP	$7,214.63
1.4 Back-End Development					$5,771.70
1.4.1 Database Design	3	10/9/2023	10/12/2023	VG, DB, JT, SB, VGO, RT, AP	$2,473.59
1.4.2 AI Reminder Management	4	10/12/2023	10/16/2023	VG, DB, JT, SB, VGO, RT, AP	$3,298.11
1.5 Mobile App Development					$22,124.85
1.5.1 Android App Development	12	10/16/2023	10/28/2023	VG, DB, JT, SB, VGO, RT, AP	$11,062.43
1.5.2 iOS App Development	12	10/16/2023	10/28/2023	VG, DB, JT, SB, VGO, RT, AP	$11,062.43
1.6 Quality Assurance					$8,657.55
1.6.1 Unit Testing	2	10/28/2023	10/30/2023	LH, ZB, VG, BS, RT	$4,328.78
1.6.2 User Acceptance Testing	2	10/28/2023	10/30/2023	LH, ZB, VG, BS, RT	$4,328.78
1.7 Documentation					$5,771.70
1.7.1 User Manual Development	4	10/30/2023	11/3/2023	CG, KJ,ZB, DB, JT, LH	$3,298.11
1.7.2 Support Documentation Development	3	11/3/2023	11/6/2023	CG, KJ,ZB, DB, JT, LH	$2,473.59

Figure 1: Work Breakdown Structure

5.3 Deployment Plan

This section serves as a fundamental framework for the CogniOpen application's development project. The development of the CogniOpen application will involve thorough requirement analysis, design of a user-friendly interface that takes into consideration user limitations, and diligent testing. Its key aspects are:

- Requirements Gathering: Careful collection and prioritization of user needs to shape the app's functionalities.
- Design: Creating an intuitive interface that conforms to the user’s limitations.
- Testing: Rigorous testing phases, including unit and integration testing.
- Deployment: A well-structured plan to ensure a seamless transition from development to production.

5.4 Change Management

Changes may be requested by clients, the project team, or Team B in the dynamic environment of project development. A structured change management procedure is in place to make sure these modifications are implemented without causing any disruption to the project alignment. Following the processes mentioned below, this process includes documenting, assessing, and prioritizing each change request.

5.4.1 Change Control Board

A Change Control Board (CCB) is set up to guarantee thorough assessment and decision-making for substantial changes that potentially affect the project’s objectives, feasibility, and risks. A detailed evaluation procedure is conducted by the CCB when a proposed change is determined to have significant ramifications. The project's goals, potential risks, and overall impact of the modification are all carefully considered throughout the CCB assessment. When deciding whether to approve or reject these modifications, the CCB uses its combined expertise to reach a well-informed conclusion. The CCB members are:

Project Role	Name	Responsibility
Team Lead	Vincent Galeano	▪ Oversees team organization, coordination, and representation ▪ Manages project milestones and deliverables
Deputy Team Lead	Kavon Johnson	▪ Supports the Team Lead (Team A) with all responsibilities
Secretary	Kavon Johnson	▪ Establishes effective team communication ▪ Manages team information flow
Technical Writer	Kavon Johnson Zach Bowman Richard Tsang David Bright Juan-Torres Chardon Laura Hamann Selam Biru Vivek Goel Benjamin Sutter Andrea Pellot	▪ Develops documentation strategies and content ▪ Produces comprehensive technical documentation
Business Analyst	Zach Bowman Vivek Goel Selam Biru	▪ Client-centric coordination ▪ Prioritizes, validates, and documents
Software Developer	David Bright Benjamin Sutter Richard Tsang Andrea Pellot Selam Biru	▪ Defines technical direction and oversees development ▪ End-to-end software development and management
UI/UX Designer	Juan-Torres Chardon Richard Tsang Andrea Pellot	▪ Shapes user experience strategy ▪ End-to-end UX design and implementation
Test Engineer	Laura Hamann Zach Bowman Richard Tsang Vivek Goel Benjamin Sutter	▪ Testing coordination and execution

Table 11: Change Control Board

5.4.2 Internal Change Requests

Changes identified within the project team, client, or Team B follow a structured process for evaluation and potential integration:

1. Initiation: Any project team member, client, or Team B can initiate an internal change request by submitting the request in Microsoft Teams.
2. Review and Assessment: The change request is reviewed by Project Manager and Team Leads
3. Impact Assessment: The potential impact of the change on project objectives, scope, schedule, budget, and resources is thoroughly assessed.

5.4.3 External Change Requests

Requests for external changes that come from the client or Team B outside the core project team are evaluated according to a specific process:

1. Submission: Through established channels, outside parties can submit change requests.
2. Initial Review: The Team Lead or Co-Team Lead conducts an initial review to understand the request's context and urgency.
3. Assessment: The project team evaluates the change's alignment with project objectives, feasibility, and potential impact.

5.4.4 Change Submittal Process

To ensure careful consideration, the process for submitting, reviewing, and evaluating change requests is designed:

1. Request Initiation: By filling out the appropriate form, any team member, stakeholder, or member of Team B can start a change request.
2. Review Team Assignment: The change coordinator assigns the request to a review team with the relevant expertise.
3. Impact Assessment: The review team evaluates the potential impact of the change on project objectives, scope, schedule, budget, and resources.
4. CCB Review (if applicable): If there are major modifications, the CCB reviews the request to determine its feasibility, alignment with the project's objectives and any potential impact.
5. Decision and Communication: The requester is informed of the decision, together with any pertinent feedback, whether to accept or reject the change request.
6. Documentation: A change log keeps track of accepted and rejected change requests for historical reference.

5.5 Schedule

This section highlights the key milestones and tasks of the project. It is a guideline for the project schedule and may be adjusted based on the project's challenges and needs that arise in the duration of the project lifecycle.

5.5.1 Milestones

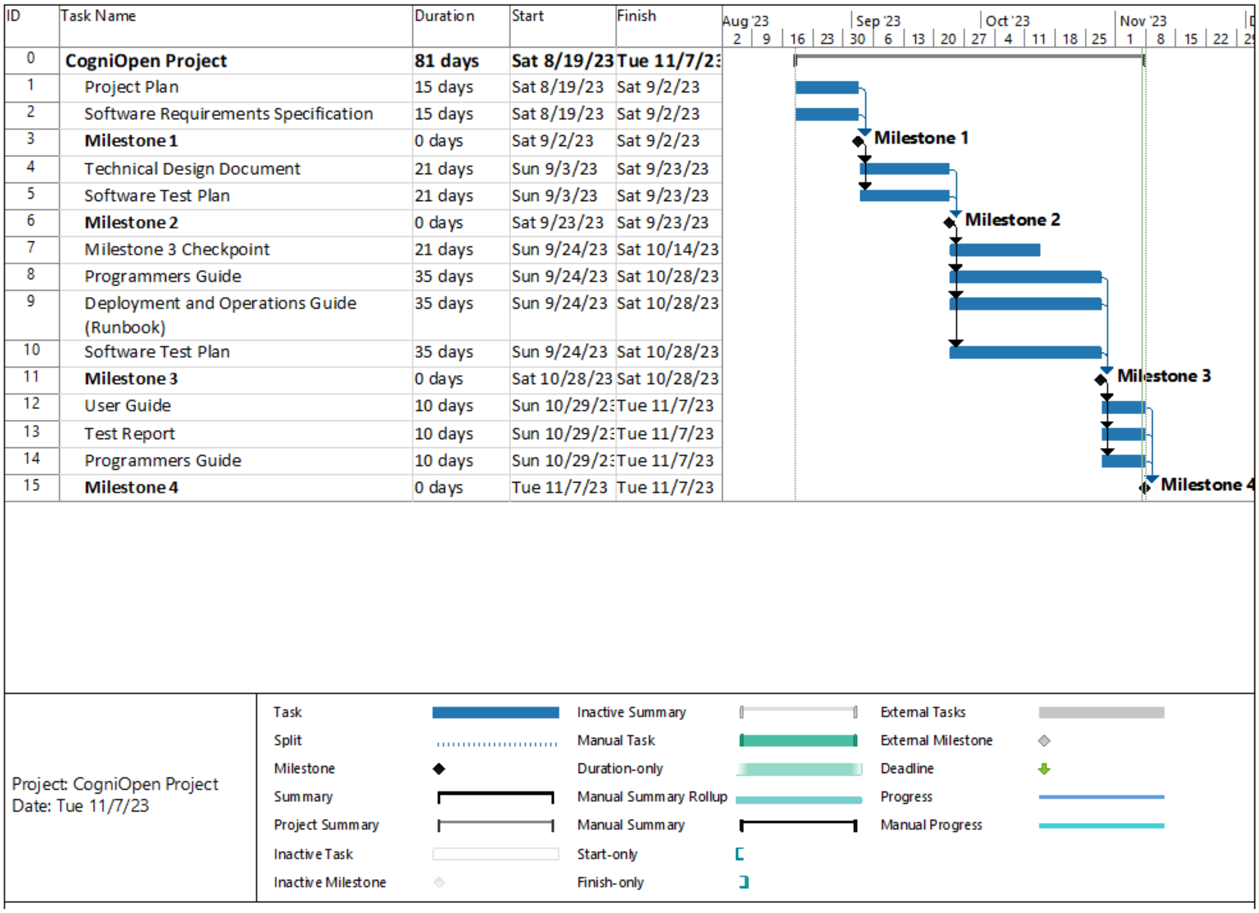
The project lifecycle is broken down into 4 major milestones over the course of an 11-week period. Each milestone is presented in the table below with its respective deliverables and due dates.

Milestone	Deliverables	Due Date
1	Project Plan Software Requirements Specification (SRS)	August 29, 2023 (Week 3)
2	Technical Design Document Software Test Plan Source Code	September 19, 2023 (Week 6)
3	Programmer Guide Deployment and Operations Guide (Runbook) Source Code	October 28, 2023 (Week 11)
4	User Guide Test Report Programmer Guide Final Code Delivery	November 7, 2023 (Week 12)

Table 12: Milestone Breakdown

5.5.2 Gantt Chart

The Gantt chart for the CogniOpen application provides a more detailed breakdown of the project schedule. Each deliverable is associated with an estimated duration as well as an approximate start and finish date. The Gantt chart will be monitored and updated at the weekly Dream Team meetings to maintain effective project progression and tackle setbacks promptly.



ID	Task Name	Duration	Start	Finish
0	**CogniOpen Project**	**81 days**	**Sat 8/19/23**	**Tue 11/7/2**
1	Project Plan	15 days	Sat 8/19/23	Sat 9/2/23
2	Software Requirements Specification	15 days	Sat 8/19/23	Sat 9/2/23
3	**Milestone 1**	0 days	Sat 9/2/23	Sat 9/2/23
4	Technical Design Document	21 days	Sun 9/3/23	Sat 9/23/23
5	Software Test Plan	21 days	Sun 9/3/23	Sat 9/23/23
6	**Milestone 2**	0 days	Sat 9/23/23	Sat 9/23/23
7	Milestone 3 Checkpoint	21 days	Sun 9/24/23	Sat 10/14/23
8	Programmers Guide	35 days	Sun 9/24/23	Sat 10/28/23
9	Deployment and Operations Guide (Runbook)	35 days	Sun 9/24/23	Sat 10/28/23
10	Software Test Plan	35 days	Sun 9/24/23	Sat 10/28/23
11	**Milestone 3**	0 days	Sat 10/28/23	Sat 10/28/23
12	User Guide	10 days	Sun 10/29/2	Tue 11/7/23
13	Test Report	10 days	Sun 10/29/2	Tue 11/7/23
14	Programmers Guide	10 days	Sun 10/29/2	Tue 11/7/23
15	**Milestone 4**	0 days	Tue 11/7/23	Tue 11/7/23

Project: CogniOpen Project
Date: Tue 11/7/23

Figure 2: CogniOpen Gantt chart

5.5.3 Project Schedule

The following high-level linear project schedule provides an overview of the project's milestones. It offers a simplified view of the project's progression and timeline.

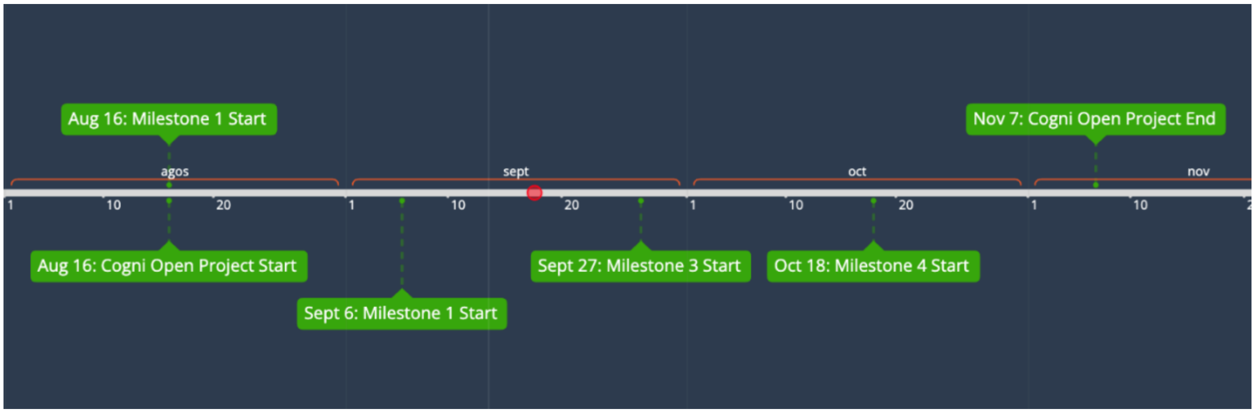

Aug 16: Milestone 1 Start
Nov 7: Cogni Open Project End
agos
sept
oct
nov
1
10
20
1
10
20
1
10
20
1
10
Aug 16: Cogni Open Project Start
Sept 27: Milestone 3 Start
Oct 18: Milestone 4 Start
Sept 6: Milestone 1 Start

Figure 3: CogniOpen Linear Timeline

6 Budget

6.1 Personnel

The expected average weekly time commitment for team members is 10-15 hours a week. Over the course of the 11-week project lifecycle, this equates to a maximum of 110 hours of work per team member.

For predictive cost analysis, a notional budget reference is based on the national average for each role from ZipRecruiter. The table below provides a prediction of the maximum notional project cost, based on each role working 15 hours per week for 11 consecutive weeks:

Role	Hourly Rate	Weekly Cost	Total Cost
Team Lead	$ 85.00	$ 1,275.00	$ 14,025.00
Lead Technical Writer	$ 50.00	$ 750.00	$ 8,250.00
Lead Business Analyst	$ 55.00	$ 825.00	$ 9,075.00
Lead Tester	$ 60.00	$ 900.00	$ 9,900.00
Lead Architect	$ 65.00	$ 975.00	$ 10,725.00
Lead UI/UX Designer	$ 63.00	$ 945.00	$ 10,395.00
UI/UX	$ 45.00	$ 675.00	$ 7,425.00
Business Analyst	$ 45.00	$ 675.00	$ 7,425.00
Tester	$ 35.00	$ 525.00	$ 5,775.00
Developer	$ 40.00	$ 600.00	$ 6,600.00
Developer	$ 40.00	$ 600.00	$ 6,600.00
Total	**$ 583.00**	**$ 8,745.00**	**$ 96,195.00**

Table 13: Notional Project Budget

6.2 Equipment

As a remote organization, DTTS members are expected to supply their own equipment in terms of hardware, mobile devices, and network end points. Any member who requires equipment of any kind is allotted a stipend up to $750 over the course of the project to use on any equipment deemed relevant to the project’s success. Upon project completion, members are expected to return any equipment purchased with the stipend to DTTS management or the team member will be expected to repay the expenses incurred out of pocket.

6.3 Licenses and External Services

As of milestone 1 and in conjunction with the first draft of the initial project plan, there are currently no required licenses identified as many of the design decisions still need to be made and approved internally. In future versions of this document, any licenses or external services utilized will be appended in this section.

6.4 Maintenance

After all deliverables are given to the customer during milestone 4, DTTS will no longer actively be involved in the project. The goal of these deliverables is to present as complete of a product as possible. However, DTTS realizes that this is rarely possible and will be eager and able to help with any work requested after the product is live. Once work is identified, the customer will contact the project manager who will then discuss with relevant department leads how to best complete the work in terms of time and talent. Once the proper personnel have been identified, the customer will be billed at any previously defined rates discussed in section 7.1.

6.5 Overhead Costs

The Dream Team greatly values the members of the team and the work that they do, no matter where they live. Since the company is entirely remote, management passes the savings of not having a building with rent and utilities to its members. These savings will reimburse members for their phone and internet plans (up to $50 for phones and $100 for internet).

6.6 Internal Training

This application is the culmination of various new, emerging technologies. As such, DTTS project members will have to familiarize themselves with these new frameworks and tools as part of their initial training/project onboarding. The labor costs associated with this training is already factored into the personnel budget in section 7.1 as part of the team’s regular work duties.

6.7 External Training

One objective of project completion will be ensuring that expansive documentation and training is available and easily accessible for the customer and any new prospective clients or users. During the final stages of the product, more detailed training to provide extensive education on the product to any relevant parties will be supplied by members of the Dream Team at no cost to the customer. After project completion, any further training sessions can be arranged at the cost of $100 per participant per session.

7 Time & Cost Management

7.1 Time management

To ensure the efficient and timely progression of the project, the schedule will be reviewed and monitored at the weekly team meetings. Delays in the project schedule will be discussed amongst the team to determine the best course of action for realignment. In conjunction with other time management tools, the Dream Team will utilize a Gantt chart to monitor the project schedule. Stakeholders will get status updates on the progression of the project. The project stakeholders will be made aware of any major setbacks and/or adjustments in the project schedule and the mitigation strategies decided by the team to get back on track.

7.2 Cost management

The Dream Team will ensure that any current or future budgets garner the approval of both internal management and any relevant parties associated with the customer. To ensure that these agreed upon budgets are fully respected, all budgetary expenses will be passively monitored on a weekly basis and actively assessed/analyzed on a milestone basis. In the event of any budgetary anomalies, the team will have an internal discussion with the department leads to review current tasking and logistics to determine how best to divert allotted funds. If the anomaly is ever to exceed $5,000, there will be a direct discussion with the customer to determine how best to proceed. These direct discussions allow the team to give the customer detailed insight as to why more funding may be required and allow them full autonomy over the current state of the project and whether a shift in direction is required to properly manage costs.

8 Quality Management

It is the goal of DTTS to ensure that all activities and deliverables meet and/or exceed the expectations of our customer throughout the project duration. Therefore, to promise this level of excellence, a detailed Quality Management Plan shall be detailed below. The purpose of this plan is to provide details on roles and responsibilities, quality objectives, procedures, and activities to help maintain the project's quality. Acceptable Quality Levels (AQL) shall be identified within subsections for documentation, software and coding activities and deliverables. In addition, each AQL shall have an associated inspection plan to ensure quality management and customer satisfaction.

8.1 Quality Management Roles & Responsibilities

To ensure adherence to the Quality Management Plan, the table below identifies members of the project team and stakeholders, along with their associated roles and responsibilities.

Role	Responsibility
Client/Project Manager	Review and approve all milestone deliverables.
Quality Lead	Notify Project Manager of quality issues. Work with appropriate Lead to resolve identified quality issues.
Quality Group	Perform reviews/inspections of activities and deliverables. Notify Quality Lead of identified quality issues. Maintain project quality logs.
Project Team and Stakeholders	Work with Quality Lead and Project Manager to address and resolve quality issues as needed.

Table 14: Role and Responsibilities Table

8.2 Quality Assurance Plans

This section shall identify individual project activities and deliverables, how each will manage quality, and associated AQLs and inspection plans.

8.2.1 Documentation Quality Assurance

Documentation quality shall be managed using peer reviews, final review by the Lead Technical Writer, and finally through approval of each document by the Project Manager and Client. Identified activities, AQLs and inspection information is documented in the table below.

Activity	AQL	Inspection
Approval of final documentation	100% of documentation must be approved by the Project Manager and Client	Quality team inspects final document for signatures
Peer review of documentation	1 peer shall review documentation before submittal to Lead Technical Writer	Quality team inspects document history for reviews

Table 15: Quality Assurance Inspection Plan

8.2.2 **Software Quality Assurance**

Software quality shall be managed throughout the entire Software Development Life Cycle (SDLC), starting from requirement definition, and continuing all the way through release. Quality will be measured based on product performance and customer satisfaction. Functional and Non-functional system requirements will be identified in detail within the Software Requirements Specification (SRS) Document. A detailed test plan will be created to ensure 100% of the requirements documented within the SRS are met. Identified activities, AQLs and inspection information is documented in the table below.

Activity	AQL	Inspection
Requirement definition	100% of requirements shall be documented	Quality team inspects SRS prior to final submission
Testing	100% of requirements shall be tested	Quality team will inspect Test Plan prior to final submission
Defect Identification	Mean Time to Detect (MTTD) less than 24 hours	Quality team will perform testing on each new release

Table 16: Software Quality Insurance Inspection Plan

8.2.3 **Code Quality Assurance**

Code quality shall be managed using code reviews and unit testing. Code for a new feature or function shall be committed to a separate feature branch. Prior to the branch being merged with master, a peer-review must be done by a senior development team member to ensure coding standards were followed, testing covered 100% of code and that the requested feature or functionality is behaving as expected. Identified activities, AQLs and inspection information is documented in the table below.

Activity	AQL	Inspection
Code Review	100% of code must be tested	For each new commit
Testing	100% of tests must be passing	For each new commit

Table 17: Code Quality Insurance Inspection Plan

9 Risk Analysis & Management

The primary objective of the CogniOpen project is to enhance the possibility of successful project completion and achieving its goals by proactively identifying, evaluating, and mitigating potential risks through implementing a Risk Analysis & Management strategy.

9.1 Risk Analysis

Risk analysis is a dynamic and iterative procedure that aids project teams within CogniOpen in anticipating, assessing, and reducing potential obstacles and uncertainties. The implementation of efficient project planning and management, guided by comprehensive risk analysis, significantly enhances the quality of decision-making procedures, hence facilitating the successful implementation of projects.

9.2 Risk Management

Risk management is a crucial aspect of any project. It involves identifying, assessing, and mitigating potential risks that could impact the success or outcome of the project.

Risk Monitoring: involves the ongoing observation and evaluation of identified risks to assess their potential impact on a project or organization. The continuous monitoring of identified risks by the project team will involve assessing their status and subsequently updating the Risk Assessment Matrix.

Risk Response Planning: involves identifying potential risks and developing strategies to mitigate or address them. The risk management approach will prioritize High risks, ensuring that prompt action is taken, and contingency plans are developed. Mitigation strategies will be formulated and executed to address Moderate risks.

Risk Ownership: refers to the identification and assignment of responsibility for managing and mitigating risks within an organization. It involves recognizing that risks exist, determining who is accountable for addressing them.

Contingency Planning: involves identifying potential risks and developing strategies to mitigate their impact on the business. The project timeline and deliverables will be safeguarded by developing contingency plans specifically designed to mitigate the impact of critical and high-level risks.

Risk Communication: involves the exchange of information about potential risks, their likelihood, and potential impacts to relevant stakeholders. Regular communication of risk status and mitigation efforts will be ensured to project stakeholders and team members through the means of status reports and meetings.

Periodic Review: is a crucial aspect of risk management that involves regularly assessing and evaluating potential risks and their impact on an organization. It will be conducted to assess the effectiveness of the Risk Assessment Matrix and risk management strategies in mitigating project risks.

9.3 Risk Assessment Matrix

As a crucial component of the risk management strategy for the CogniOpen project, the project team has crafted a comprehensive Risk Assessment Matrix. This matrix is a structured framework for systematically identifying, assessing, and evaluating possible risks that could affect the success of the project. The risks identified in this matrix have been subjected to a thorough analysis and classification procedure. They have been classified into four distinct levels based on probability and impact.

Low Risk: Risks in this category are regarded to have a low probability of occurring and a low potential impact on the project's objectives. Although they may still require attention and monitoring, it is not anticipated that they will pose a significant hazard to the project's success.

Medium Risk: Risks with a moderate likelihood of occurrence and a moderate prospective impact are classified as moderate risk. These risks are recognized as having a reasonable likelihood of materializing and could have a negative impact on certain aspects of the project if they are not managed appropriately.

High Risk: Risks in this category have a high probability of occurring and the potential to significantly impact the project. These risks necessitate proactive management strategies to mitigate their effects and lower their occurrence probabilities.

Likelihood Level	Description
Low	Unlikely to occur
Medium	Possible but not certain
High	Highly likely to occur

Figure 4: Risk Matrix Key

9.3.1 Risk Matrix

The justification for the risk matrix, where risks have been categorized based on their potential impact and likelihood. In categorizing these risks, a balance has been struck between their potential impact and likelihood. Mitigation and contingency plans have been put in place to minimize their overall impact on the project, ensuring a proactive approach to risk management.

Data Security and Privacy: While the risk of a data breach is low due to robust encryption and compliance measures, the project manager has implemented mitigation and contingency plans to address this potential threat.

User Acceptance: While there is a possibility of encountering challenges related to user adoption, it's not considered highly probable. User adoption is critical to project success, and to mitigate this risk, user-centric design and testing are being carried out.

Integration Challenges: This risk is categorized as high impact with moderate probability since integration issues can affect project schedules and functionality.

ChatGPT Efficiency: Continuous optimization and monitoring of ChatGPT are in place to address performance issues, with backup plans and alternate approaches available to reduce the likelihood of major impacts.

ChatGPT and AWS Service Availability: Although service interruptions are unlikely, the impact on project operations is significant.

Compliance with Regulations: Noncompliance with regulations can lead to legal and reputational consequences.

Technical Complicatedness: Technical complexity can result in project delays and issues. Mitigation includes training and skill development, with the Project Manager prepared to respond immediately to technical challenges.

User Acceptance Testing: To maintain software integrity, rigorous testing and feedback incorporation are carried out.

Data Loss: Even with minimal probability, data loss can be detrimental. The Software Developer shall implement proper storage, data backup, and recovery protocols.

ID	Risk Description	Likelihood	Impact	Mitigation Plan	Contingency Plan	Owner
1	Data Privacy and Security	Low	High	Data encryption, compliance, and audits that are robust	Plans will be in place for data recovery and breach response.	Project Manager
2	User Adoption	Med	Med	User-centric design and comprehensive user testing	UX Designer employ user training tactics.	Lead UI/UX Designer
3	Integration Difficulties	Med	High	Integration testing and monitoring on a comprehensive scale	Parallel development for important components is led by the Lead Software Developer	Lead Software Developer
4	ChatGPT Performance	Low	Med	Continuous ChatGPT model optimization and monitoring	Back-up plans and alternate approaches to ChatGPT	Lead Software Developer
5	ChatGPT and AWS	Low	High	Backup solutions for moderate to	Service-level agreements with providers	Lead Software Developer

ID	Risk Description	Likelihood	Impact	Mitigation Plan	Contingency Plan	Owner
	Service Availability			high service outages		
6	Regulatory Compliance	Low	High	Legal compliance checks and upgrades	counsel for issues of noncompliance	Project Manager
7	Technical Complexity	Med	High	team training and skill development	Project Manager rapid response to team for technical concerns.	Project Manager
8	User Acceptance Testing	Med	Med	Rigorous user testing, feedback incorporation	Lead Tester will be responsible for testing environments and teams	Lead Tester
9	Data Loss	Low	High	Redundant storage, data backup, and data recovery protocols all need to be considered	Lead Software Developer provides real-time data synchronization and auditing.	Lead Software Developer

Figure 5: Risk Assessment Matrix