VIROTOUR

Project Management Plan

University of Maryland Global Campus

SWEN 670: Team A

Spring 2023

Version 4.0

April 4, 2023

Document Control

Document History

Version	Issue Date	Changes
0.1	1/25/2023	Initial Draft.
1.0	1/28/2023	Milestone 1 Submission
2.0	2/12/2023	Milestone 2 Re-submission
3.0	3/25/2023	Milestone 3 Re-submission
4.0	4/4/2023	Milestone 4 Submission

Table of Contents

1. Project Information

ViroTour (VT) is a cutting-edge virtual tour software that enables users to showcase physical spaces in stunning detail. With immersive virtual tours featuring interactive panoramic images and informational hotspots, ViroTour offers a dynamic and engaging way for users to explore virtual spaces. ViroTour allows users to upload images, which will automatically create interactive tours and extract text from the images to create informational hotspots. Other features include applying a glow effect to tour images, with varying intensity levels, and giving users the ability to view tours using a virtual reality (VR) headset to take the virtual tour experience to the next level.

To achieve the highest user experience and customer satisfaction, our developing team have chosen Dart and Flutter as the programming language and library respectively. This allows us to create an all-in-on fast, beautiful, native quality applications that can perform on various operating systems and devices using just one codebase. At the end of this project, the users will be able to turn a space into an immersive digital version which can be used to design, build, promote, an understand any building or physical platform.

1.1.Purpose

This Project Management Plan or PMP will outline the approach and standards for each deliverable, how the project team will create, monitor, and deliver tasks. Every project milestone's delivery schedule will be outlined. The project stakeholders, including the client, project sponsor, senior leadership, and the project team made up of the project manager, product owner, software developers, quality assurance specialists, and business analysts, are the intended audiences of the ViroTour project management plan (PMP). The purpose of this PMP is to outline the particular processes that the ViroTour project team will adhere to throughout the project lifecycle for the benefit of the project team and all other stakeholders. This plan must be followed for all tasks carried out under ViroTour. Throughout the project's implementation, additions to the PMP will be required. The documents will offer the artifacts that support the details of the project, including requirement elicitation and validation, technical design choices, development of user acceptability testing criteria, and other documents required to support the application throughout its life cycle.

1.2.Statement of Need

Because of the fast and vast development of new techniques for collecting images, photo technology has advanced significantly over time. Nowadays, an application on a smartphone may be used to create totally immersive experiences and take lifelike photos at previously unheard-of resolutions. 3D camara is not a brand-new technology but they are not quite common for everyday use. ViroTour development team is on a mission to make the technology more user friendly and practical for the 3D camera fans and even the upcoming market.

1.3.Vision Statement

ViroTour development team has the vision to build a well-functioning application for the general 3D camera. The application includes panoramic image capture, manual placement of hotspots, cartesian and spherical image support, applying filters and reading text from images using Google Vision with user friendly interface. With the human face blurring feature and a virtual reality view at the end of the tour, we thrive to satisfy all of the client's requirements.

1.4.Stakeholders

"Stakeholder" are individuals directly or indirectly involved in the project and who ultimately may influence or be impacted by the project's outcome. The following individuals represent internal and external interested parties that will need to participate effectively throughout the project to achieve a successful outcome. Stakeholders can include but are not limited to signatories, customers, sponsors, project team members, independent testers, certifying organizations, external parties, and others with a vested interest in the project outcome.

Stakeholder Name	Project Role
Dr. Mir Assadullah	Client/Professor
Roy Gordon	Project Mentor
Robert Wilson	DevSecOps Mentor
Khoa Nguyen	Project Manager (PM)
Jacob Lynn	Product Owner (PO)
Viet Nguyen	Lead Developer (Dev Lead)
Tilahun Abreha	Business Analyst (BA)
Fedor Menchukov	Software Engineer (Dev)
Jude Ibe	Software Engineer (Dev)
Nicholas Platt	Software Engineer (Dev)
Samson Alemneh	Software Engineer (Dev)
Jeffrey Welch	Software Engineer (Dev)
Shawn Kagwa	Software Engineer (Dev)
Christian Dovel	Software Engineer (Dev)

Table 1.4 – Stakeholder Information

1.5.Project Methodology

This project will be executed using a modified Agile Scrum which will be adjusted to be less structured due to the nature of the project. All team members are supporting this project as a second job, and thus scheduled meetings will be more challenging to arrange. Teams primarily use the concept of cross-functional teams, where members can support activities like planning, developing code, testing, QA, and other tasks that the current milestone requires. This is because the team has all been through a similar level of training, and the level of effort required for specific roles will be inconsistent across milestones. A highly adaptable team will ensure a higher chance of success for the project.

1.6.Project Tools

The execution of the ViroTour application enhancements project will depend on using the following tools and services.

Name	Description	Version
Microsoft Teams	A messaging application the project team uses to collaborate, hold meetings, and share files.	N/A
SharePoint	A web-based Microsoft platform that allows the project team to manage and store documents.	N/A
Microsoft Word	A software-as-a-service (SaaS) tool used for creating and editing word documents.	N/A
Microsoft Excel	A software-as-a-service (SaaS) tool used for creating and editing excel documents.	N/A
Pencil	A GUI prototyping tool used to create mockups.	3.1.0
GitHub	A software development internet tool used for version control using Git.	N/A
Dart	An object-oriented programming language developed by Google	N/A
Flutter	An open-source framework developed by Google	N/A
Android Studio	An integrated development environment (IDE) used to develop and test mobile applications for Google Android devices.	N/A

Table 1.6 – Project Tools

2. Scope Management

2.1.Scope

The goal of this project is to develop a virtual tour web and mobile application to be delivered by the Software Engineering Project Capstone class of Spring 2023. Our team, Team

A, along with Team B, have split the work and will be collaborating throughout the project's development. This project plan has been created to outline the deliverables, tasks, and constraints needed to ensure successful completion of the project from the standpoint of Team A.

The deliverables include the following:

1. Project Plan
2. Software Requirements Specification
3. Technical Design Document
4. Software Test Plan
5. Deployment and Operations Guide
6. Programmer Guide
7. User Guide
8. Test Report
9. Application Code

In Scope for Team A:

- User Interface foundational design.
- Ability to display a location.
- Ability to navigate from one location to another.
- Ensuring smooth transitions between locations.
- Ability to zoom and pan the view.
- Displaying UI elements of the search results.
- Displaying UI elements of hotspot editing feature.
- Ability to add a "glow" effect to the image when uploading.
- Ability to view using VR viewer.

Out of Scope for Team A:

Team B's Responsibilities:

- Database to store Images, Locations, Hotspots, and Text.
- Core Module: Image processing to enable tour navigation.
- Editing Module: Perform modifications on Images, Locations and Hotspots.
- Search Module: Perform text-based search.
- Backend support for any number of Virtual Tours.

Other:

- User profile management.

- User authorization.

2.2.Work Breakdown Structure (WBS)

Work Breakdown Structure (WBS)

WBS	Task Name	Work Hours	Duration	Start	Finish	Resource Names
1	ViroTour Application	1185	83	1/11	4/4	
1.1	Scope and Project Management	50	4	1/11	1/15	
1.1.1	Determine Project Scope	20	4	1/11	1/15	KN, JL
1.1.2	Define Preliminary Resources	10	4	1/11	1/15	KN
1.1.3	Team Meeting	20	4	1/11	1/15	KN, JL, TA, VN, CD, FM, JI, NP, SA, JW
1.2	Requirements	120	14	1/14	1/28	
1.2.1	Review Software Specifications (internally)	10	7	1/14	1/21	KN, TA, JL, VN
1.2.2	Review Software Specifications (externally with Team B and client)	10	4	1/17	1/21	KN, JL, VN
1.2.3	Draft Preliminary Software Specifications	30	4	1/21	1/25	KN, TA, SA, VN, JW, CD, NP, JL, SK, FM
1.2.4	Develop Preliminary Schedule and Budget	10	4	1/21	1/25	KN
1.2.5	Review Schedule and Budget	10	3	1/25	1/28	KN
1.2.6	Review Project Plan and SRS	10	3	1/25	1/28	KN, JL, VN, SA, TA, NP
1.2.7	Milestone 1 Deliverables (Project Plan, SRS, Presentation)	30	3	1/25	1/28	KN, JL, TA, VN, CD, FM, NP, SA, JW, SK

1.2.8	Team Meeting	10	1	1/27	1/28	KN, JL, TA, VN, CD, FM, JI, NP, SA, JW
1.3	Design	165	13	1/29	2/11	
1.3.1	Review Product Backlog	5	13	1/29	2/11	KN, JL
1.3.2	System Architecture Design	10	13	1/29	2/11	VN
1.3.3	Component 1 Design: Tour Navigation	30	13	1/29	2/11	VN, NP, SA, KN
1.3.4	Component 2 Design: View Customization	30	13	1/29	2/11	VN, CD, TA, JW
1.3.5	Component 3 Design: Hotspot Customization	30	13	1/29	2/11	VN, NP, FM, CD, JW, JL
1.3.6	Cross-team Discussion	20	13	1/29	2/11	KN, JL, TA, VN, CD, FM, JI, NP, SA, JW
1.3.7	Milestone 2 Deliverables (TDD, Test Plan)	30	13	1/29	2/11	KN, JL, TA, VN, CD, FM, JI, NP, SA, JW
1.3.8	Team Meeting	10	13	1/29	2/11	KN, JL, TA, VN, CD, FM, JI, NP, SA, JW
1.4	Development	700	41	2/12	3/25	
1.4.1	Sprint 1	160	7	2/12	2/19	
1.4.1.1	Sprint Retrospective	10	7	2/12	2/19	KN, JL, VN, NP, SA
1.4.1.2	Drafting Project DepOps	30	7	2/12	2/19	KN, SA, VN, NP
1.4.1.3	Drafting Programmer's Guide	40	7	2/12	2/19	KN, JL, SA, FM, JW, VN, NP, SK
1.4.1.4	Building Individual Features	70	7	2/12	2/19	All Grey Below

1.4.1.4.1	List Tours	10	7	2/12	2/19	NP
1.4.1.4.2	Search Tour	10	7	2/12	2/19	VN, SK
1.4.1.4.3	Enable VR	0	7	2/12	2/19	FM
1.4.1.4.4	Create Tour	10	7	2/12	2/19	CD, JW
1.4.1.4.5	Edit Tour	10	7	2/12	2/19	NP, JW
1.4.1.4.6	Hamburger menu	0	7	2/12	2/19	NP, FM
1.4.1.4.7	Wheel menu	10	7	2/12	2/19	SA
1.4.1.4.8	Glow effect	10	7	2/12	2/19	TA
1.4.1.4.9	View Tour	10	7	2/12	2/19	NP, VN
1.4.1.4.10	Test API/Explore Testing/Building Test Environment	0	7	2/12	2/19	NP, VN
1.4.1.5	Team Meeting	10	7	2/12	2/19	KN, JL, TA, VN, CD, FM, JI, NP, SA, JW
1.4.2	Sprint 2	120	7	2/20	2/27	
1.4.2.1	Sprint Retrospective (cont.)	5	7	2/20	2/27	KN
1.4.2.2	Building Individual Features	85	7	2/20	2/27	All Grey Below
1.4.2.2.1	List Tours	10	7	2/20	2/27	NP

1.4.2.2.2	Search Tour	10	7	2/20	2/27	VN, SK
1.4.2.2.3	Enable VR	0	7	2/20	2/27	FM
1.4.2.2.4	Create Tour	10	7	2/20	2/27	CD, JW
1.4.2.2.5	Edit Tour	10	7	2/20	2/27	NP, JW
1.4.2.2.6	Hamburger menu	10	7	2/20	2/27	NP, FM
1.4.2.2.7	Wheel menu	10	7	2/20	2/27	SA
1.4.2.2.8	Glow effect	5	7	2/20	2/27	TA
1.4.2.2.9	View Tour	10	7	2/20	2/27	NP, VN
1.4.2.2.10	Test API/Explore Testing/Building Test Environment	10	7	2/20	2/27	NP, VN
1.4.2.3	Drafting Project DepOps	10	7	2/20	2/27	KN, SA
1.4.2.4	Drafting Programmer's Guide	10	7	2/20	2/27	KN, SA
1.4.2.5	Team Meeting	10	7	2/20	2/27	KN, JL, TA, VN, CD, FM, JI, NP, SA, JW
1.4.3	Sprint 3	145	7	2/28	3/7	
1.4.3.1	Sprint Retrospective (cont.)	5	7	2/28	3/7	KN
1.4.3.2	Building Individual Features	90	7	2/28	3/7	All Grey Below

1.4.3.2.1	List Tours	10	7	2/28	3/7	NP
1.4.3.2.2	Search Tour	10	7	2/28	3/7	VN
1.4.3.2.3	Enable VR	0	7	2/28	3/7	FM
1.4.3.2.4	Create Tour	10	7	2/28	3/7	CD, JW
1.4.3.2.5	Edit Tour	10	7	2/28	3/7	NP, JW
1.4.3.2.6	Hamburger menu	10	7	2/28	3/7	NP, FM
1.4.3.2.7	Wheel menu	10	7	2/28	3/7	SA
1.4.3.2.8	Glow effect	10	7	2/28	3/7	TA
1.4.3.2.9	View Tour	10	7	2/28	3/7	NP, VN
1.4.3.2.10	Test API/Explore Testing/Building Test Environment	10	7	2/28	3/7	NP, VN
1.4.3.3	Code Review	30	7	2/20	2/27	NP, VN
1.4.3.4	Drafting Project DepOps	5	7	2/28	3/7	KN, SA
1.4.3.5	Drafting Programmer's Guide	5	7	2/28	3/7	KN, SA
1.4.3.6	Team Meeting	10	7	2/28	3/7	KN, JL, TA, VN, CD, FM, JI, NP, SA, JW
1.4.4	Sprint 4	125	7	3/8	3/15	

1.4.4.1	Sprint Retrospective (cont.)	5	7	3/8	3/15	KN
1.4.4.2	Building Features in Modules	70	7	3/8	3/15	All Grey Below
1.4.2.2.1	List Tours	10	7	3/8	3/15	NP
1.4.2.2.2	Search Tour	10	7	3/8	3/15	VN
1.4.2.2.3	Enable VR	20	7	3/8	3/15	FM
1.4.2.2.4	Create Tour	0	7	3/8	3/15	CD, JW
1.4.2.2.5	Edit Tour	10	7	3/8	3/15	NP, JW
1.4.2.2.6	Hamburger menu	0	7	3/8	3/15	NP, FM
1.4.2.2.7	Wheel menu	5	7	3/8	3/15	SA
1.4.2.2.8	Glow effect	5	7	3/8	3/15	TA
1.4.2.2.9	View Tour	10	7	3/8	3/15	NP, VN
1.4.2.2.10	Test API/Explore Testing/Building Test Environment	0	7	3/8	3/15	NP, VN
1.4.4.3	Code Review	30	7	3/8	3/15	NP, VN
1.4.4.4	Drafting Project DepOps	5	7	3/8	3/15	KN, SA

1.4.4.5	Drafting Programmer's Guide	5	7	3/8	3/15	KN, SA
1.4.4.6	Team Meeting	10	7	3/8	3/15	KN, JL, TA, VN, CD, FM, JI, NP, SA, JW
1.4.5	Sprint 5	150	9	3/16	3/25	
1.4.5.1	Sprint Retrospective (cont.)	5	9	3/16	3/25	KN
1.4.5.2	Building Features in Modules	50	9	3/16	3/25	All Grey Below
1.4.5.2.1	List Tours	10	9	3/16	3/25	NP
1.4.5.2.2	Search Tour	10	9	3/16	3/25	VN
1.4.5.2.3	Enable VR	0	9	3/16	3/25	FM
1.4.5.2.4	Create Tour	10	9	3/16	3/25	CD, JW
1.4.5.2.5	Edit Tour	10	9	3/16	3/25	NP, JW
1.4.5.2.6	Hamburger menu	10	9	3/16	3/25	NP, FM
1.4.5.2.7	Wheel menu	5	9	3/16	3/25	SA
1.4.5.2.8	Glow effect	0	9	3/16	3/25	TA
1.4.5.2.9	View Tour	10	9	3/16	3/25	NP, VN
1.4.5.2.10	Test API/Explore Testing/Building Test Environment	0	9	3/16	3/25	NP, VN

1.4.5.3	Code Review	30	9	3/16	3/25	NP, VN
1.4.5.4	Drafting Project DepOps	10	9	3/16	3/25	KN, SA
1.4.5.5	Drafting Programmer's Guide	10	9	3/16	3/25	KN, SA
1.4.5.6	Comments and Styling	10	9	3/16	3/25	KN, SA
1.4.5.7	Proofread	20	9	3/16	3/25	SA, KN, CD, NP, JL
1.4.5.8	Milestone 3 Deliverables (DevOps, Programmer's Guide, Presentation)	10	9	3/16	3/25	KN, JL, TA, VN, CD, FM, NP, SA, JW
1.4.5.9	Team Meeting	5	9	3/16	3/25	KN, JL, TA, VN, CD, FM, JI, NP, SA, JW
1.5	Testing and Documentation	75	6	3/26	4/1	
1.5.1	Final Testing	20	6	3/26	4/1	NP, JL, JI, SK, VN, KN, TA, CD, FM, JW, SA
1.5.2	Draft User Guide	5	6	3/26	4/1	SA, JL, JI, SK
1.5.3	Draft Test Report	5	6	3/26	4/1	SA, JL, JI, SK, NP, KN
1.5.4	Review User Guide	5	6	3/26	4/1	NP, JL, JI, SK, VN, KN, TA, CD, FM, JW, SA
1.5.5	Review Test Report	5	6	3/26	4/1	NP, JL, JI, SK, VN, KN, TA, CD, FM, JW, SA
1.5.6	Test and Documentation Review with Client	5	6	3/26	4/1	NP, JL, JI, SK, VN, KN, TA, CD, FM, JW, SA
1.5.7	Incorporate feedback into User Guide and Test Report	5	6	3/26	4/1	NP, JL, JI, SK, VN, KN, TA, CD, FM, JW, SA

1.5.8	Incorporate Feedback into Specifications, Schedule, and Budget	5	6	3/26	4/1	NP, JL, JI, SK, VN, KN, TA, CD, FM, JW, SA
1.5.9	Milestone 4 Deliverables Due (User Guide & Test Report)	5	6	3/26	4/1	NP, JL, JI, SK, VN, KN, TA, CD, FM, JW, SA
1.5.10	Teams Meeting	10	6	3/26	4/1	NP, JL, JI, VN, KN, TA, CD, FM, JW, SA
1.7	Deployment	75	6	3/26	4/1	
1.7.1	Certify Release and Archive	5	6	3/26	4/1	NP, JL, JI, SK, VN, KN, TA, CD, FM, JW, SA
1.7.3	Milestone 4 Deliverables (User's Guide, Test Report, Presentation)	60	6	3/26	4/1	NP, JL, JI, SK, VN, KN, TA, CD, FM, JW, SA
1.7.4	Upload Project Code and Deliverables to GitHub	5	6	3/26	4/1	NP, JL, JI, SK, VN, KN, TA, CD, FM, JW, SA
1.7.5	Team Meeting	5	6	3/26	4/1	NP, JL, JI, SK, VN, KN, TA, CD, FM, JW, SA

Table 2.2 – Work Breakdown Structure

2.3.Deployment Plan

The deployment plan will be established during Milestone 2. We will provide details about how to build and deploy the project through mobile and web platforms. A few of the steps we will need to take are outlined on Flutter's online documentation ("Build and release," n.d.):

1. Signing the application with a digital signature.
2. Creating an upload keystore.
3. Shrinking the code.
4. Reviewing the app's manifest.
5. Building the app for release.

2.4.Change Management

Change management will be handled by Project Managers. Any updated requirements will be brought to their attention or through the project's communication board Microsoft Teams. By outlining requirements early in the project lifecycle, change is minimized. Minimizing changes decreases risk since new requirements later on the in the project's lifecycle increases scope and therefore requires additional unplanned work. Internal Changes Requests

Change requests that derive from the project team will be collected and discussed internally and evaluated for viability. If the change request is deemed necessary and directly affects the client, the Project Manager and Product Owner will gain approval for the change from the client before prioritizing and executing the change.

2.5.Change Control Board

Team A utilized the Work Distribution Excel File to view the work details and the hierarchy of review for the documentation. After the junior engineers work on the assigned tasks, the status changes to 'Ready to Review'. The Document In-charge will perform the first round of review and leave comments if any. The Project Manager, Product Owner, and Lead Developer subsequentially performs a high level and detail technical review. The junior developers then go through the comments and make adjustments if applicable.

Role	Responsibilities
Project Manager/ Product Owner/ Lead Developer	• Leads application design and document structure • High level review • Detail review
Document In-charge served as a back-up for the PM/PO/Lead Dev	• Manages and leads the completion of the document
Junior Developers	• Self-assign tasks and work within subgroup • Asks questions by adding comments within the document

Table 2.5 - Change request form process flow requirement

2.6.External Changes Requests

If new top-level requirements are required by course management during project execution, the teams will communicate and vote to determine which Team makes sense to take it on. Teams anticipate that they will receive the responsibility if the request is like what a team is already doing. However, if many change requests occur, the workload between the teams will be considered and distributed in a way that the Project Manager and Product Owners see fit to keep the product on schedule.

As changes are understood and accepted by the Team, Issues will be created in GitHub to make the required updates to any affected documentation or application component and integrated into a current or future sprint.

2.7.Change Request Process

Step	Description
Create Task in the Work Distribution Excel File	A team member determine what needs to be change and create a task
Complete Task and Create Code Review	Team member completes a CR Form and sends the completed form to the Change Manager
Authorize	Approval to move forward with incorporating the suggested change into the project/product
Implement	If approved, make the necessary adjustments to carry out the requested change and communicate CR status to the submitter and other stakeholders

Table 2.7 - Change request form process flow requirement

2.8.Change request form and change management log

Element	Description
Date	The date the CR was created
Code Review	Assigned by the Change Manager
Title	A brief description of the change request
Description	Description of the desired change, the impact, or benefits of a change should also be described
Submitter	Name of the person completing the CR Form and who can answer questions regarding the suggested change
Product	The product that the suggested change is for
Version	The product version that the suggested change is for
Priority	A code that provides a recommended categorization of the urgency of the requested change (High, Medium, Low)

Table 2.8 - Change request form and change management log

2.9.Evaluating and authorizing change request

Priority	Description
High	The change is needed as soon as possible because of potentially damaging service impact.
Medium	The change will solve irritating problems or repair missing functionality. This change can be scheduled.
Low	The change will lead to improvements, changes in workflow, or configuration. This change can be scheduled.

Table 2.9 – Evaluating and authorizing change request flow

2.10. Change requests are evaluated and assigned one or more of the following change types

Type	Description
Scope	Change affecting scope
Time	Change affecting time
Cost	Change affecting cost
Resources	Change affecting resources
Deliverables	Change affecting deliverables
Product	Change affecting product
Quality	Change affecting quality

Table 2.10 – Change request change types

2.11. Change request are evaluated and assigned on of the following status types

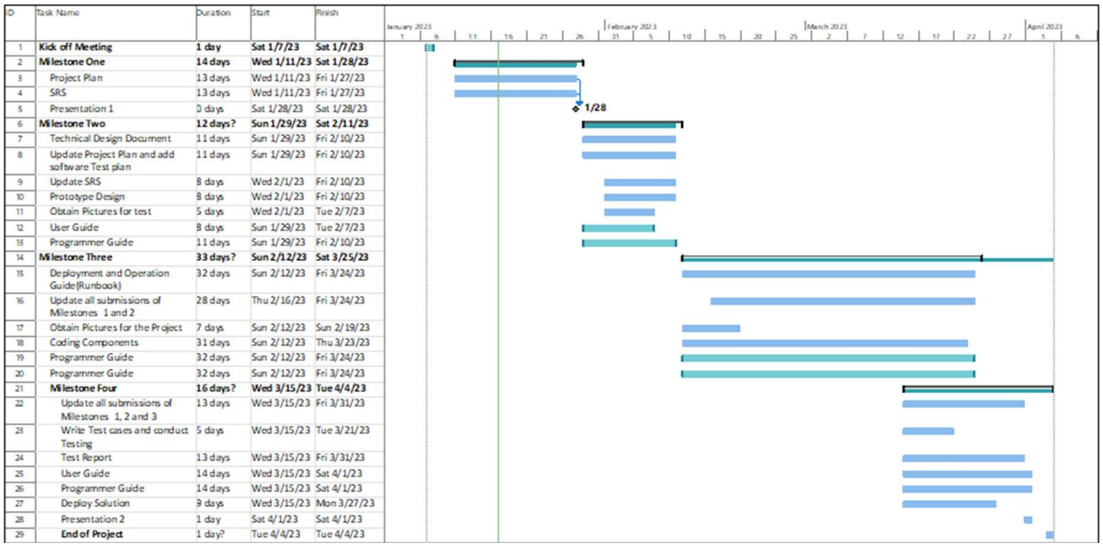
Status	Description
Open	Entered/Open but not yet approved or assigned
Work in Progress	CR approved, assigned, and work is progressing
In Review	CR work is completed and in final review prior to testing
Testing	CR work has been reviewed and is being tested
Close	CR work is complete, has passed all tests, and updates have been released.

Table 2.11 – Change request status types

3. Time Management

3.1.Schedule

The project schedule for ViroTour application contains four major milestones containing various tasks under them. The Gantt chart below shows information about each task in the project from start to finish.



ID	Task Name	Duration	Start	Finish
1	**Kick off Meeting**	**1 day**	**Sat 1/7/23**	**Sat 1/7/23**
2	**Milestone One**	**14 days**	**Wed 1/11/23**	**Sat 1/28/23**
3	Project Plan	13 days	Wed 1/11/23	Fri 1/27/23
4	SRS	13 days	Wed 1/11/23	Fri 1/27/23
5	Presentation 1	0 days	Sat 1/28/23	Sat 1/28/23
6	**Milestone Two**	**12 days?**	**Sun 1/29/23**	**Sat 2/11/23**
7	Technical Design Document	11 days	Sun 1/29/23	Fri 2/10/23
8	Update Project Plan and add software Test plan	11 days	Sun 1/29/23	Fri 2/10/23
9	Update SRS	8 days	Wed 2/1/23	Fri 2/10/23
10	Prototype Design	8 days	Wed 2/1/23	Fri 2/10/23
11	Obtain Pictures for test	5 days	Wed 2/1/23	Tue 2/7/23
12	User Guide	8 days	Sun 1/29/23	Tue 2/7/23
13	Programmer Guide	11 days	Sun 1/29/23	Fri 2/10/23
14	**Milestone Three**	**33 days?**	**Sun 2/12/23**	**Sat 3/25/23**
15	Deployment and Operation Guide(Runbook)	32 days	Sun 2/12/23	Fri 3/24/23
16	Update all submissions of Milestones 1 and 2	28 days	Thu 2/16/23	Fri 3/24/23
17	Obtain Pictures for the Project	7 days	Sun 2/12/23	Sun 2/19/23
18	Coding Components	31 days	Sun 2/12/23	Thu 3/23/23
19	Programmer Guide	32 days	Sun 2/12/23	Fri 3/24/23
20	Programmer Guide	32 days	Sun 2/12/23	Fri 3/24/23
21	**Milestone Four**	**16 days?**	**Wed 3/15/23**	**Tue 4/4/23**
22	Update all submissions of Milestones 1, 2 and 3	13 days	Wed 3/15/23	Fri 3/31/23
23	Write Test cases and conduct Testing	5 days	Wed 3/15/23	Tue 3/21/23
24	Test Report	13 days	Wed 3/15/23	Fri 3/31/23
25	User Guide	14 days	Wed 3/15/23	Sat 4/1/23
26	Programmer Guide	14 days	Wed 3/15/23	Sat 4/1/23
27	Deploy Solution	9 days	Wed 3/15/23	Mon 3/27/23
28	Presentation 2	1 day	Sat 4/1/23	Sat 4/1/23
29	**End of Project**	1 day?	Tue 4/4/23	Tue 4/4/23

Figure 3.1 - Gantt Chart showing Project Schedule by Milestone

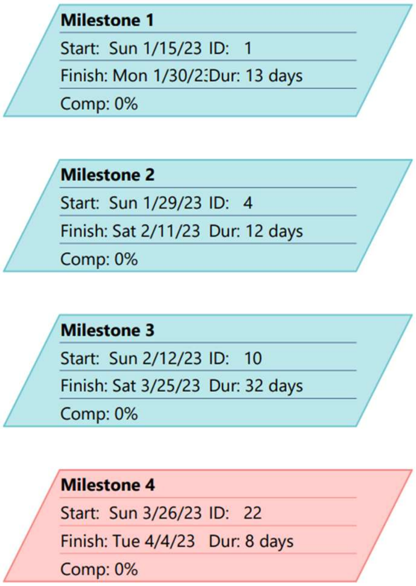
Note:

Refer to the Appendix for a project schedule.

Testing will take place in sprints starting 2/26/2023 to end of the project.

3.2.Milestones

Milestones are significant events within the project schedule and depend on the statement of work (SOW) and project deliverables. The ViroTour application project has four milestones each containing multiple tasks to be completed. The completion of every task in brings about the completion of a milestone which indicates progress in the project. In the Appendix section of this document a Microsoft Project Gantt chart is included to show detail information about the milestones and tasks under them.



Milestone 1
Start: Sun 1/15/23 ID: 1
Finish: Mon 1/30/2:Dur: 13 days
Comp: 0%

Milestone 2
Start: Sun 1/29/23 ID: 4
Finish: Sat 2/11/23 Dur: 12 days
Comp: 0%

Milestone 3
Start: Sun 2/12/23 ID: 10
Finish: Sat 3/25/23 Dur: 32 days
Comp: 0%

Milestone 4
Start: Sun 3/26/23 ID: 22
Finish: Tue 4/4/23 Dur: 8 days
Comp: 0%

Description	**Delivery Date**
Milestone 1	
Project Plan	Week 3 (1/28)
Software Requirements Specification (SRS)	Week 3 (1/28)
Presentation 1	Week 3 (1/28)
Milestone 2	
Technical Design Document	Week 5 (2/11)
User guide	Week 5 (2/11)
Programmer Guide	Week 5 (2/11)
Deploy Prototype	Week 5 (2/11)
Presentation 2	Week 5 (2/11)
Milestone 3	
Deployment and Operations Guide (Runbook)	Week 10 (3/25)
Add Software Test Plan to the Project Plan	Week 10 (3/25)
Deploy version 1	Week 12 (3/25)
Milestone 4	
Test Report	Week 12 (4/4)
User Guide	Week 12 (4/4)
Programmer Guide	Week 12 (4/4)
Deploy version 2	Week 12 (4/4)

Presentation 2	Week 12 (4/4)

Table 3.2 - Project Milestones

3.3.Dependencies

Activity	Depends on	Dependency
Project Plan, Task Planning, Risk Identification	Business Case, Feasibility Study, Project Charter, Stakeholder Analysis, Team Assembly	Start-to-start
Manage SDLC process, Project Deliverables, Change Requests, Performance Data, Issue Log, Documentation Request, Monitor Control	Project Plan, Task Planning, Risk Identification	Finish-to start
Review deliverables and ensure all the work has been completed, obtain acceptance of project deliverables, hand off operations and support responsibilities, document lesson learned	Manage SDLC process, Project Deliverables, Change Requests, Performance Data, Issue Log, Risk Register, Documentation Request, Monitor Control	Finish-to-finish

Table 3.3 - Project Activity Dependencies

3.4.Assumptions

- There are limited changes to the scope and requirements during the initial stage of the project.
- The input images to build the panoramic image can be collected successfully using any phone camera mounted on the Matterport Axis device.
- The input images can be processed using Matterport platform.
- The Team can acclimatize the coding requirement of the Flutter framework and Dart programming language.
- The requirements are mainly addressed within the project life span.

3.5.Constraints

- The project is a second job to most team members, and the project life span from start to finish is short for the number of requirements expected.
- Project members' skills working with Flutter framework and Dart programming language may be limited.
- The project being run from remote locations and in different time zones hinders the progress of the project.

4. Cost Management

The ViroTour Project has no realized cost to the Client throughout the project. For reference, the cost analysis of salaries was retrieved in January 2023 from salary aggregation website levels for employees of Amazon. Team A members will work on the project for eleven consecutive weeks, an average of 1-2 hours on each of the five weekdays.

The cost analysis/predictions are for the total life span of the project, which is 11 weeks. The table covers weekly work hours, hourly pay per professional, and a weekly pay analysis. Depending on the interest of the project stakeholders, the team can present a cost analysis report on every milestone completion by the project manager or assigned member.

Role	Hourly Rate	Weekly Cost	Total Cost
Product Owner (L5)	$95.00	$950.00	$10,450.00
Product Manager (L5)	$90.00	$900.00	$9,900.00
Senior Software Lead (L4)	$75.00	$750.00	$8,250.00
Test Manager (L4)	$75.00	$750.00	$8,250.00
Junior Software Developer (L3)	$65.00	$650.00	$7,150.00
Junior Software Developer (L3)	$65.00	$650.00	$7,150.00
Junior Software Developer (L3)	$65.00	$650.00	$7,150.00
Junior Software Developer (L3)	$65.00	$650.00	$7,150.00
Junior Software Developer (L3)	$65.00	$650.00	$7,150.00
Junior Software Developer (L3)	$65.00	$650.00	$7,150.00
Business Analyst (L3)	$65.00	$650.00	$7,150.00
Total	**$790.00**	**$7,900.00**	**$86,900.00**

Table 4 – Project Cost

5. Quality Management

- Product Quality:

The project team will use this to measure whether the ViroTour system meets up with the requirements and specifications set. The deliverables will contain the usability and reliability of the system which all stakeholders are made aware of, and all newly identified quality standards are documented in the Project Management Plan.

- Process Quality:

This would be established by the project team. This will typically be used to assess the data analysis of the process performance over time leading to outputs that will correlate with the deliverables that the ViroTour system will produce, or other outputs expected from this activity.

5.1.Quality Assurance

The ViroTour Project's quality assurance basically ensures that the system meets all the quality standards. Based on the existing standards, this process will be a continuous process throughout the project as it would be assessed based on the objectives of this project. The deliverables will continue to be assessed by the project team given that the project's lifecycle uses an iterative process to ensure the products high standards.

Role	Quality Assurance Responsibility
Project Sponsor	Approve all quality standards for the USPS Informed Delivery Project. Review and approve all project tasks and deliverables.
Project Manager/Quality Manager	Assure availability of essential project resources for identified quality activities. Ensure resolution of quality issues escalated. Facilitate resolution of quality issues, escalating as needed.
Quality Group	Work with the Project Sponsor and Project Manager to develop and implement the Quality Management Plan. Recommend tracking techniques and methods to establish acceptable quality levels. Create and maintain project quality logs.
Project team and stakeholders	Assist the Project Manager, Project Sponsor, and Quality Specialists in establishing acceptable quality standards.

Table 5.1 Quality Assurance

6. Staffing Management

The ViroTour Project will consist of a matrix structure with support from internal resources that will be allocated to this project. All work will be performed internally, and no outside third-party contractual assistance will be required.

Staffing requirements for the ViroTour Project include the following:

Project Manager / Scrum Master – responsible for all ViroTour Project management. The Project Manager / Scrum Master is responsible for providing a framework for project methodology & plan, directing the team towards a common goal, monitoring progress, ensuring stakeholder satisfaction, managing time & money, and resolving roadblocks within the team.

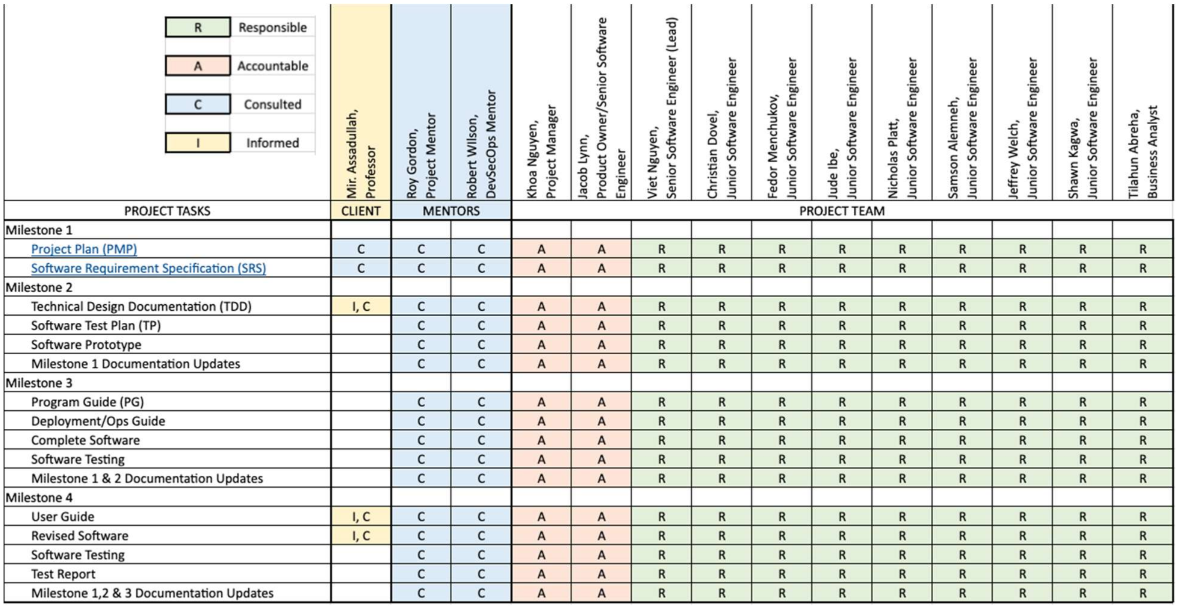
Product Owner (1 position) – responsible to act as the internal representative/interface for the needs of the customer, creating and managing the team backlog so it meets the needs of the customer, and providing quality assurance by overseeing and reviewing all completed work by the Team to ensure it passes customer acceptance.

Senior Software Engineer (2 positions) – responsible for coding and programming for the ViroTour Project. Responsibilities include issue assignment based on preferences, completing project documentation, software development, quality assurance testing, presentation of functionality during milestone reviews, and aligning responsibilities to high-priority items in the product backlog. Additionally, this position will be responsible for mentoring and reviewing code prepared by the Software Engineer I and II positions.

Junior Software Engineer (6 positions) – responsible for coding and programming for the ViroTour Project. Responsibilities include issue assignment based on preferences, completing project documentation, software development, quality assurance testing, presentation of functionality during milestone reviews, and aligning responsibilities to high-priority items in the product backlog.

DevSecOps Engineer (shared) – responsible for managing the code repository policies and managing the application build/release pipelines. As time allows, this position is also responsible for issue assignments based on preferences, completing project documentation, software development, quality assurance testing, presentation of functionality during milestone reviews, and aligning responsibilities to high-priority items in the product backlog.

6.1.RACI Matrix



R	Responsible
A	Accountable
C	Consulted
I	Informed

	Mir. Assadullah, Professor	Roy Gordon, Project Mentor	Robert Wilson, DevSecOps Mentor	Khoa Nguyen, Project Manager	Jacob Lynn, Product Owner/Senior Software Engineer	Viet Nguyen, Senior Software Engineer (Lead)	Christian Dovel, Junior Software Engineer	Fedor Menchukov, Junior Software Engineer	Jude Ibe, Junior Software Engineer	Nicholas Platt, Junior Software Engineer	Samson Alemneh, Junior Software Engineer	Jeffrey Welch, Junior Software Engineer	Shawn Kagwa, Junior Software Engineer	Tilahun Abreha, Business Analyst
PROJECT TASKS	CLIENT	MENTORS		PROJECT TEAM										
Milestone 1														
Project Plan (PMP)	C	C	C	A	A	R	R	R	R	R	R	R	R	R
Software Requirement Specification (SRS)	C	C	C	A	A	R	R	R	R	R	R	R	R	R
Milestone 2														
Technical Design Documentation (TDD)	I, C	C	C	A	A	R	R	R	R	R	R	R	R	R
Software Test Plan (TP)		C	C	A	A	R	R	R	R	R	R	R	R	R
Software Prototype		C	C	A	A	R	R	R	R	R	R	R	R	R
Milestone 1 Documentation Updates		C	C	A	A	R	R	R	R	R	R	R	R	R
Milestone 3														
Program Guide (PG)		C	C	A	A	R	R	R	R	R	R	R	R	R
Deployment/Ops Guide		C	C	A	A	R	R	R	R	R	R	R	R	R
Complete Software		C	C	A	A	R	R	R	R	R	R	R	R	R
Software Testing		C	C	A	A	R	R	R	R	R	R	R	R	R
Milestone 1 & 2 Documentation Updates		C	C	A	A	R	R	R	R	R	R	R	R	R
Milestone 4														
User Guide	I, C	C	C	A	A	R	R	R	R	R	R	R	R	R
Revised Software	I, C	C	C	A	A	R	R	R	R	R	R	R	R	R
Software Testing		C	C	A	A	R	R	R	R	R	R	R	R	R
Test Report		C	C	A	A	R	R	R	R	R	R	R	R	R
Milestone 1,2 & 3 Documentation Updates		C	C	A	A	R	R	R	R	R	R	R	R	R

Table 6.1 – RACI Matrix

6.2.Roles and Responsibilities

Resource	Role	Responsibilities
Dr. Mir Assadullah	Client	• Provides business context, expertise, and guidance to the project manager and the team. • Acts as the link between the project, the business, and strategic level decision-making groups.
Roy Gordon Robert Wilson	Project Mentor	• Provide guidance and advisory for any impediments, issues, and improvements.
Khoa Nguyen	Project Manager Scrum Master	• Provide project guidance for the project team. • Foster communication and collaboration within the team. • Ensure project remains on schedule and on budget and bring to attention any risks/issues that may impact the team. • Contributes to documentation and implementation (project plan, SRS, etc.).

Jacob Lynn	Product Owner Test Manager	• Manage and prioritize the project backlog. • Serve as a liaison between product and development. • Evaluate project progress. • Contributes to documentation and implementation (project plan, SRS, etc.). • Managed customer deliverables.
Viet Nguyen	Software Engineer Senior	• Leads development of coding and deployment. • Researches and proposes architecture. • Provides technical leadership. Supports Junior Engineers. • Contributes to documentation and implementation (project plan, SRS, etc.).
Nicholas Platt	Test Manager Software Engineer	• Develop test strategy and test implementation plan. • Similar responsibilities as the Software Engineer Senior.
Tilahun Abreha	Business Analyst	• Manages requirements. • Serves as a liaison to the Product Owner. • Contributes to documentation and implementation (project plan, SRS, etc.).
Christian Dovel Fedor Menchukov Jude Ibe Samson Alemneh Jeffrey Welch Shawn Kagwa	Software Engineer Junior	• Contributes to coding, deployment, and testing. • Contributes to documentation and implementation (project plan, SRS, etc.).

Table 6.2 – Roles and Responsibilities

7. Communications Management

Microsoft Teams will be the primary communication methods. The communication management plan for Team A anticipates frequent communication among all stakeholders. The plan also identifies specific communication channels to ensure effective project coordination, with the goal of maintaining a clear and consistent flow of information throughout the duration of the project. Additionally, the plan establishes protocols for addressing and resolving any communication-related issues that may arise, to minimize disruptions and maintain momentum on the project.

7.1.Microsoft Teams

This platform was built around and has extended traditional video conferencing technology to focus on increasing a user's ability to collaborate quickly with individuals or groups, which is crucial to mitigating risks associated with working with a large group over a distance. Communication on this platform will be facilitated by using Teams and Channels. A Team is a grouping mechanism for individuals with a common goal.

Team	Distribution
@Team A	Includes all members of Team A
@Team B	Includes all members of Team B
@Customer	Dr. Mir Assadullah
@Project Manager	Khoa Nguyen
@Product Owners	Jacob Lynn

Table 7.1 - MS Teams Tags

MS Teams Channels:

- Joint Collaboration: For all communication related to the project between members of Team A and Team B.
- Team A: Acted as the communication and working directory for Team A.

7.2.GitHub

GitHub will serve as the main tool for the team to coordinate their work on issue tracking and application development. Team members will keep track of backlogs, assign tasks to themselves based on the priority of current milestones, and update the status of tasks to communicate progress. This approach will enhance the team's ability to coordinate their work and give each member a clearer picture of the project's progress.

8. Risk Management

8.1.Risk identification: early in the process in other to arrange it and prioritize.

- Risk assessment: to located and find the appropriate solution.
- Risk control: develop and appropriate response to minimize his effect on the project.
- Risk review control: to make sure that the solution is efficient and some case if there is no way to eradicate at least mitigate the risk and make sure that the control is effective.

- Risk register: tool usedto identify, assess, and prioritize potential risks that could affect the success of a project. It is a key component of risk management and provides a structured way to manage potential risks throughout the project lifecycle. The risk register includes a list of potential risks, their likelihood and impact, as well as any mitigation or contingency plans that have been developed to manage them. It is used to monitor and track the progress of risk mitigation actions and provides a record of how risks have been identified and addressed throughout the project. See appendix D: Risk Register.

8.2.Purpose of Risk Management

A risk is a condition that, if it occurs, could have a positive or negative impact on one or more project objectives. A Risk Management strategy provides a framework for processing these events. The following tasks identified in the risk management plan define how risks connected to the ViroTour Project will be managed using one of the five risk mitigation strategies outlined below. Additionally, the plan outlines how risk management tasks will be accomplished, documented, and monitored throughout the project's lifecycle and provides templates and practices for recording and prioritizing risks.

The risk management plan includes the following tasks:

- **Identify** – develop a comprehensive list of all known uncertainties with the potential to impact the project and record them in the risk register (Appendix D). Techniques such as brainstorming, cause and effect (Ishikawa) diagrams, and expert judgment can be used to develop a thorough understanding of this project's risks.
- **Classify –** use qualitative techniques to prioritize identified risks and quantitative techniques to determine the potential impact of those risks.
- **Plan –** integrate risk response into project plans by using the following techniques: creativity, decision-support, and implementation. Creativity techniques help identify potential responses, decision-support techniques are used to find the optimal response, and implementation techniques are used to identify a specific action based on a risk response.
- **Monitor –** risk monitoring activities will be integrated into project status tracking activities to ensure that stakeholders remain informed, risk plans should be periodically updated when major changes are incorporated into the project schedule, risks should be reprioritized periodically as needed to eliminate those with the lowest priority, and the risk register should be updated as new risks are identified.
- **Mitigating Actions –** based on the risk classification, the following mitigating options are available:

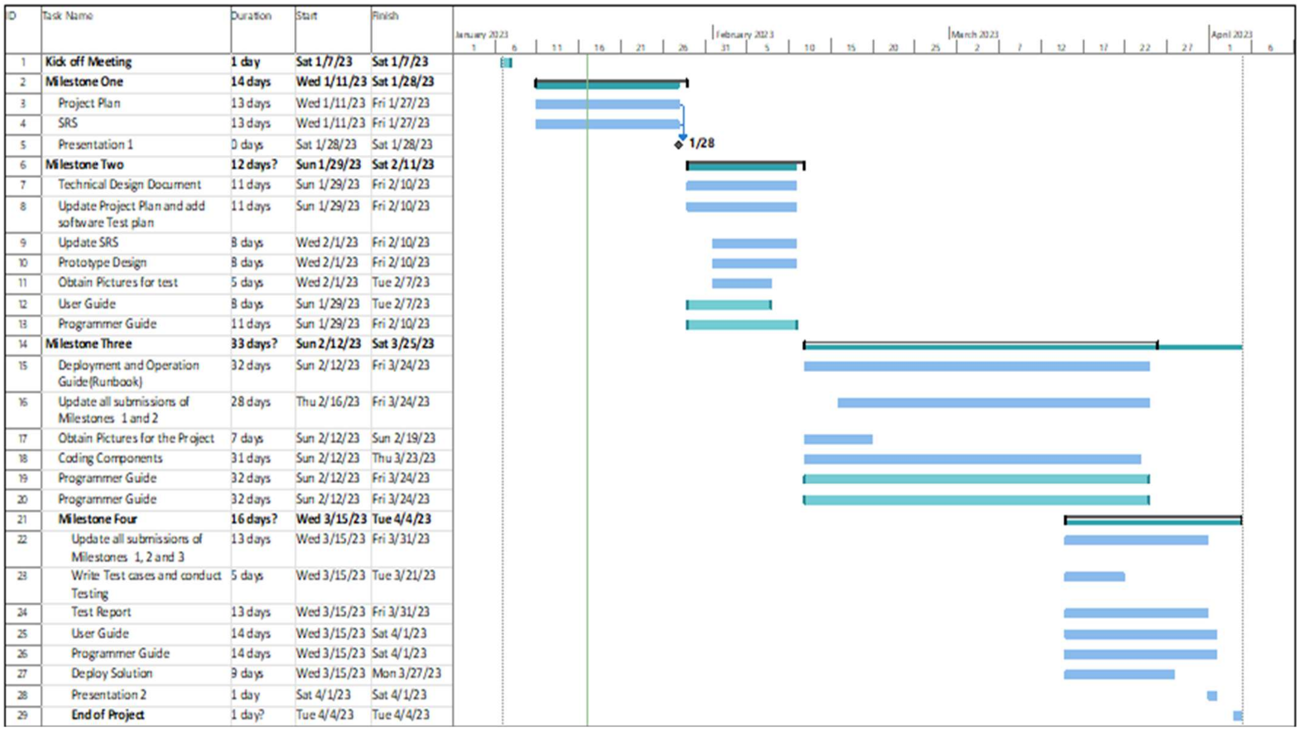
    - **Acceptance** – acknowledge that a risk exists and is impacting the project but implement no mitigating action.
    - **Avoid** – mitigate the risk by adjusting the scope, schedule, or budget.
    - **Control** – implement an action limiting the risk's impact on the project.
    - **Transfer** – transfer the risk out of the project to another stakeholder.
    - **Continue monitoring –** in the case of low-impact risk, continue to monitor to ensure the risk's possible impact does not increase.
- **Communicate –** open and transparent communication about the impact and status of project risks with stakeholders throughout the execution of a project is critical to a successful outcome.

Following the closeout process, the Project Manager will perform a post-mortem on the risk management activities for the ViroTour Project to identify lessons learned that can be documented for future projects.

Appendix A - Acronyms and Abbreviations

1. 3D – three-dimension/three dimensional
2. API - Application Programming Interface
3. CCB - Change Control Board
4. CD - Continuous Delivery
5. CI - Continuous Integration
6. CR – Code Review
7. EVM – Earned Value Management
8. FR - Functional Requirement
9. GUI – Graphical User Interface
10. iOS – iPhone Operating System
11. NFR – Non-functional Requirement
12. QA – Quality Assurance
13. PMP – Project Management Plan
14. RACI – Responsible, Accountable, Consulted, Informed
15. RMF – Risk Management Framework.
16. SDLC - Software Development Life Cycle
17. SOW – Statement of Work
18. SWE - Software Engineer
19. UI – User Interface
20. UMGC – University of Maryland Global Campus
21. VC – Version Control
22. WBS – Work Breakdown Structure
23. YAML - Yet Another Markup Language

Appendix B - Figure 1 Detailed Project Schedule (Listing all project phases, activities, and tasks) by Milestone



ID	Task Name	Duration	Start	Finish
1	**Kick off Meeting**	**1 day**	**Sat 1/7/23**	**Sat 1/7/23**
2	**Milestone One**	**14 days**	**Wed 1/11/23**	**Sat 1/28/23**
3	Project Plan	13 days	Wed 1/11/23	Fri 1/27/23
4	SRS	13 days	Wed 1/11/23	Fri 1/27/23
5	Presentation 1	0 days	Sat 1/28/23	Sat 1/28/23
6	**Milestone Two**	**12 days?**	**Sun 1/29/23**	**Sat 2/11/23**
7	Technical Design Document	11 days	Sun 1/29/23	Fri 2/10/23
8	Update Project Plan and add software Test plan	11 days	Sun 1/29/23	Fri 2/10/23
9	Update SRS	8 days	Wed 2/1/23	Fri 2/10/23
10	Prototype Design	8 days	Wed 2/1/23	Fri 2/10/23
11	Obtain Pictures for test	5 days	Wed 2/1/23	Tue 2/7/23
12	User Guide	8 days	Sun 1/29/23	Tue 2/7/23
13	Programmer Guide	11 days	Sun 1/29/23	Fri 2/10/23
14	**Milestone Three**	**33 days?**	**Sun 2/12/23**	**Sat 3/25/23**
15	Deployment and Operation Guide(Runbook)	32 days	Sun 2/12/23	Fri 3/24/23
16	Update all submissions of Milestones 1 and 2	28 days	Thu 2/16/23	Fri 3/24/23
17	Obtain Pictures for the Project	7 days	Sun 2/12/23	Sun 2/19/23
18	Coding Components	31 days	Sun 2/12/23	Thu 3/23/23
19	Programmer Guide	32 days	Sun 2/12/23	Fri 3/24/23
20	Programmer Guide	32 days	Sun 2/12/23	Fri 3/24/23
21	**Milestone Four**	**16 days?**	**Wed 3/15/23**	**Tue 4/4/23**
22	Update all submissions of Milestones 1, 2 and 3	13 days	Wed 3/15/23	Fri 3/31/23
23	Write Test cases and conduct Testing	5 days	Wed 3/15/23	Tue 3/21/23
24	Test Report	13 days	Wed 3/15/23	Fri 3/31/23
25	User Guide	14 days	Wed 3/15/23	Sat 4/1/23
26	Programmer Guide	14 days	Wed 3/15/23	Sat 4/1/23
27	Deploy Solution	9 days	Wed 3/15/23	Mon 3/27/23
28	Presentation 2	1 day	Sat 4/1/23	Sat 4/1/23
29	**End of Project**	1 day?	Tue 4/4/23	Tue 4/4/23

Figure B – Detailed Project Schedule

Appendix C - Condensed Project Schedule

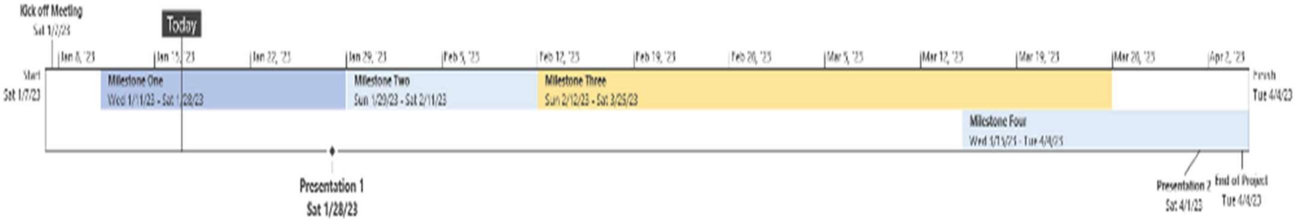


Figure C2- Condensed Project Schedule

Appendix D – Risk Register

Risk ID	Risk Description	Probability	Impact	Severity	Mitigation Strategy	Contingency Plan
R001	Delay in obtaining necessary permits and licenses	Medium	High	High	Start the application process as soon as possible and have regular follow-ups with the regulatory agencies	Delay the project timeline, allocate additional resources to expedite the permit process, and explore alternative locations
R002	Inadequate funding or cost overruns	High	High	Critical	Develop a detailed budget and track expenses closely. Explore alternative sources of funding	Re-evaluate the budget and scope of the project. Seek additional funding sources, reduce project scope or extend the timeline
R003	Technical issues with the software or hardware	Low	High	Medium	Perform thorough testing and quality assurance. Have a backup plan in place in case of failure.	Contact technical support or seek outside expertise. Develop contingency plan for hardware failures
R004	Inadequate user adoption or interest	High	Mediu m	Medium	Conduct user research and testing to understand their needs and preferences. Provide incentives for early adopters.	Re-evaluate the user experience and improve the product accordingly. Explore additional marketing and promotional strategies
R005	Weather conditions impacting the outdoor VR experience	Medium	Mediu m	Medium	Monitor weather forecasts and have contingency plans in place for rescheduling or indoor activities	Explore alternative indoor activities or experiences in case of unfavorable weather
R006	Key team member leaving the project	Low	High	Medium	Develop cross-training plans and have backup team members	Reassign responsibilities or recruit new team

					identified. Offer competitive compensation and incentives to retain key team members.	members. Plan for knowledge transfer
R007	Data privacy and security breaches	Medium	High	High	Implement robust security protocols, perform regular security audits and risk assessments.	Have a breach response plan in place. Notify affected parties immediately and address the breach promptly
R008	Legal disputes with vendors or partners	Medium	High	High	Have detailed contracts and agreements in place. Have legal counsel review all contracts and agreements.	Seek legal counsel and attempt to negotiate a settlement. Have a contingency plan for finding new vendors or partners
R009	Health and safety concerns for participants and staff	Medium	High	High	Develop comprehensive health and safety protocols. Train staff on proper safety procedures. Monitor and enforce compliance.	Have a contingency plan in place for emergencies. Consult with public health officials if necessary

Figure D - Risk Register

Note: This risk register is not appropriate for every situation. It should be reviewed and updated regularly throughout the project lifecycle.