Winfra

Project Management Plan

Version 2.0

Document Number: SPMP-002

Project Team Number: B13

Project Team Members: Jenesis Blancaflor (jb7801), Ruthvik Mukkamala (rcm8412), Sanjida Orpi (so2189), Elijah Wilson (esw9386)

REVIEW AND APPROVALS

Team Members	Function	Date	Signature
Jenesis Blancaflor	Author	2/19/25	JB
Sanjida Orpi	Author	2/19/25	SO
Elijah Wilson	Author	2/19/25	EW
Ruthvik Mukkamala	Author	2/19/25	RM

REVISION LEVEL

Date	Revision Number	Purpose
11/26/2024	Version 1.0	Initial Release
2/19/2025	Version 2.0	Final Release

TABLE OF CONTENTS

1. OVERVIEW

1.1. PROJECT SUMMARY

The motivation of the Winfra application is to address the challenges faced by New Yorkers due to a lack of infrastructure maintenance by developing a system that streamlines infrastructure service reporting to meet repair needs. The project aims to improve New York City citizens' satisfaction with infrastructure and improve accessibility.

The purpose of this document is to formalize a Software Project Management Plan (SPMP) which describes the process that will be used for the development of the Winfra system. This document aims to ensure that all stakeholders, including but not limited to management, developers, the software quality assurance group, and clients, have a clear understanding of the structural organization, processes, and plans required for the development and maintenance of the software application.

1.2. PURPOSE, SCOPE, AND OBJECTIVES

- **Purpose**: The Winfra system's purpose is to build an interactive network for citizens to report damaged or ineffective infrastructure in order to facilitate the repair of infrastructure in New York City.
- **Scope**: The system will include the following functions:
    - Infrastructural suggestion
    - Community connection
    - Accessibility promotion
    - Interactive map interface
    - Location authentication
    - CTA

    The platform will allow users to post reports of issues in real time through a simple location pin-drop feature on an interactive map interface. Each post will provide a detailed report of an issue along with the citizen's recommendations and exigencies for improvement.

    Users will be able to view reported problems within their area and engage in discussions to build community awareness. The system will enable direct outreach to local government representatives, which will streamline efforts to address infrastructure needs in underrepresented areas.

Mobile access through applications on smartphones and similar devices will be supported, to permit updated and accurate reports. The system will centralize infrastructural reports onto a single, accessible platform that enables users to track issues in their communities.

- **Business Needs**: To address the need for a centralized platform that streamlines infrastructure reporting and community engagement to promote improvement and repairs, the Winfra system requires the team to develop a web application. The release will provide features including the interactive map for location-based reporting, user report posts, user commenting, and providing mobile connection to local representatives. The website will incorporate a variety of scalable technologies to provide an interactive user interface, maintain secure data management, and comply with legal regulations to satisfy the features outlined for this release. In order to maintain an active user base, we will petition with government officials to promote the use of this website and potentially engage students around the New York City Metropolitan area. A market consideration for a future release is to incorporate banner ads and opportunities to gain revenue to meet the economic needs for the maintenance of the project as the bandwidth of users increases.

1.3. ASSUMPTIONS AND CONSTRAINTS

(Winfra Project Proposal Version 1.0 September 17, 2024)

- **Cost and Funding**: We hope to gain funding from key stakeholders within the NYC community such as government officials, investors, and large organizations in the area who are incentivized to improve the local community. The factors that contribute to the cost of the project include domain maintenance, server hosting, web development and maintenance, and user testing.
- **Delivery Dates**: We would like the full cycle of the development to be complete by the end of the year, May 2025. Key deliverables will be discussed with the technical manager for explicit time measurements.
- **Priority**: We would like the website to be the highest priority item for the developers, and we would like the website to consist of an intuitive user interface, with scalable architecture for tens of thousands of NYC residents. Furthermore, we would like the domain for the website to be properly managed to ensure that users are correctly directed to the site as well as provided with key information about our services. Some low order priority items can include “designer” features for flags with higher/lower user interaction or custom designs for the website’s frontend.

- **Mandatory vs. Optional**: A mandatory requirement to meet our business and market needs will be a fully functional map interface for users to flag important infrastructure development needs for the community. An optional requirement is the development of a message board to local officials to heighten awareness of civil issues within the city. For functional testing, we would like the developers to write unit tests, with proper CI/CD pipelines and documentation of defects.

1.4. PROJECT DELIVERABLES

- Project Proposal: February 5, 2025
- Software Requirements Specification (SRS): February 10, 2025
- Software Project Management Plan (SPMP): February 19, 2025
- Project Description: February 24, 2025
- Design Description (SDD) – Initial (w/ initial code): March 3, 2025
- Design Document Final (w/ final code): April 21, 2025 – May 5, 2025
- Formal Oral Presentation / Demonstration: April 21, 2025

1.5. SCHEDULE AND BUDGET SUMMARY

- **Schedule:**

  The project schedule outlines the major milestones and activities required for completion by May 2025. The system design phase, which involves finalizing the architecture and technical specifications, is set to be completed by early February 2025. Following this, the interactive map interface will be developed and integrated by the end of February 2025. The posting and commenting features will be built and tested by March 2025, ensuring user engagement functionalities are operational. The integration of call-to-action (CTA) elements and accessibility information will enhance usability and inclusivity, with this phase concluding in April 2025. Quality assurance testing, beta testing, and feedback collection will occur throughout the final months, leading to the project's final deployment and launch by May 2025. Budget allocations will correspond to these phases, ensuring resources are available for each critical activity.

- **Budget** *(To be completed in a future delivery)*

1.6. EVOLUTION OF THE PLAN

Under any instance where a change is made to the system which impacts the schedule and management of the project, changes will be documented on this Software Project Management Plan. The plan will be updated as needed and the plan will then be inspected and posted.

The SPMP will evolve through a structured process to ensure all changes are carefully managed. Issues or feedback will be identified during stakeholder discussions, team reviews, or milestone meetings and logged for analysis. Each proposed change will be assessed on its effects on timelines, resources, and deliverables. Approved changes will then be documented in the SPMP, with updated schedules, strategies, or deliverables communicated to the team and stakeholders.

The evolutions will be placed under configuration management using tools like GitHub to track changes. This process ensures all changes are traceable, and the most current version of the plan is accessible to the team and stakeholders.

2. REFERENCES

Blancaflor, Mukkamala, Orpi, Wilson. “Winfra Project Proposal.” Version 2.0. 5 Feb. 2025.

Blancaflor, Mukkamala, Orpi, Wilson. “Winfra System Requirements Specification.” Version 4.0. [SRS-004] 10 Feb. 2025.

3. DEFINITIONS

Term	Definition
CTA	Call to Action
Scalable software	Development languages that are popular or easy to use.
SQL	Structured Query Language: Used for database management
API	Application Programming Interface
UI	User Interface
Front-end Technologies	Tools and frameworks used to build the user interface the web applications
Back-end Technologies	Tools and frameworks used to build the service side portion of the user interface the web applications
Geolocation	The mapping of an IP address, used to identify the geographic location of a device
Target location	The location at which a user aims to review local infrastructure
PK	Primary Key
FK	Foreign Key
(S)QA	(Software) Quality Assurance
MVP	Minimum Viable Product

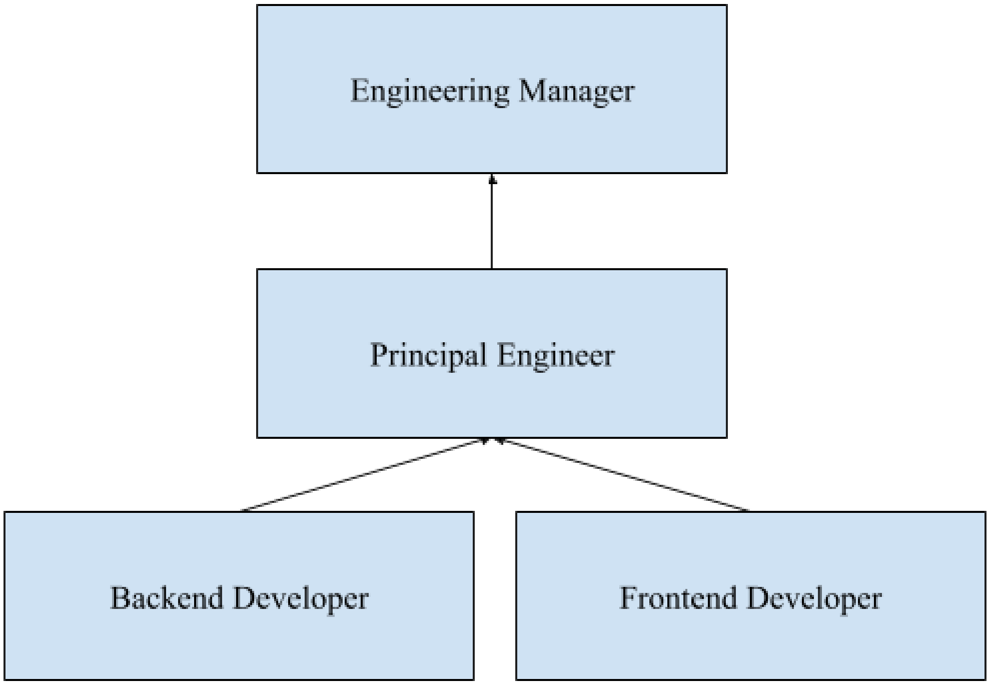
4. PROJECT ORGANIZATION

4.1. EXTERNAL INTERFACES

The organizational boundaries between the Winfra project and external entities are clearly defined to ensure clear responsibilities for the successful implementation of this project. The team must interface with the software quality group, testing group, developers, and operations. The software quality group will review products and deliverables to ensure that all products meet quality standards and requirements. The development team will be responsible for system integration and development of features. The operations team will be responsible for monitoring the system and maintaining the reliability of the system's performance. The team will additionally need to consult and interface with end users and authorities to ensure accessibility and features meet standards.

4.2. INTERNAL STRUCTURE

Within the team, there is a frontend developer, backend developer, product manager, and a project manager however by running a lean team an individual may need to take on additional responsibilities in order for the team to meet the milestones and requirements. Everyone on the team is capable of developing the software, however there are individuals who specialize in particular areas. The engineering manager helps influence project management decisions, while the principal engineer and the backend/frontend developers will help with sprint planning and meeting milestones.



4.3. ROLES AND RESPONSIBILITIES

Hereafter, “Team” refers to Jenesis Blancaflor, Ruthvik Mukkamala, Sanjida Orpi, and Elijah Wilson.

Roles	Members
Authors	Team
Reviewers	Team

SQA Group	Team
Poster	Elijah Wilson
Engineering Manager	Jenesis Blancaflor
Principal Engineer	Ruthvik Mukkamala
Design Lead	Elijah Wilson
Frontend Developer	Sanjida Orpi

5. MANAGEMENT PROCESSES

5.1. STARTUP PLAN

5.1.1. Estimation Plan (To be completed in a future delivery)

5.1.2. Staffing Plan (To be completed in a future delivery)

5.1.3. Resource Acquisition Plan (To be completed in a future delivery)

5.1.4. Training Plan

For software engineers and developers, the goal of training is to adapt new employees to the company's culture, project mission, and tools, ensure that there is proficiency with technical tools and programming languages, familiarity with the company tech stack, frameworks, and workflows, and to cultivate collaboration and initiation between the new member and the team. The training schedule for software engineers and developers is four weeks long for 10 hours a week. Four developers for the project are needed and they will need to be trained on all aspects mentioned. Week 1 of the training is an orientation that focuses on Winfra's mission values and team introductions so that new engineers and developers are familiar with the team standards and people. Week 2 of the training focuses on company policies and the work environment, getting developers ready and set up to work in their environment. Week 3 of the training focuses on technical training and mentorship from the software leads to ensure that the developers are well acquainted with the languages, frameworks, and current projects that are ongoing. Week 4 of the training focuses on team integration and involvement in a project, this will be how the new developer will continue working from this point on. The exit criterion is to demonstrate a strong ability to produce programming projects and complete tasks with the required programming languages and frameworks, using the company tech stack and environment, as well as the success in working with a smaller team or another member to complete a task collaboratively.

For managers, the goal of training is to ensure that new managers have the appropriate leadership skills to manage teams, are able to further collaboration and performance effectively, and are familiarized with the company goals, missions, members, team structure, and organization. The training schedule for managerial positions is five weeks long for 12 hours a week. Week 1 of the training is an orientation that focuses on Winfra's mission values and team introductions so that the new manager is confident about the team's goals, values, and their leadership expectations. Week 2 of the training focuses on policies involved with their position and introductions to all members on the team to gain familiarity with the staff and teams that they will be managing. Week 3 and 4 of the training focuses on management skills such as leadership, decision making, conflict resolution, and adaptability. Week 5 of the training focuses on team integration and starting to practice the management skills learned in the prior weeks. The exit criteria is to demonstrate proficiency with company tools, organization, and policies, ability to lead meetings and effectively communicate with other managers and all employees, and show strength in decision making through a final assessment which will assess these qualities. The methods used to provide this training in an effective manner for all new employees are workshops, meetings, mentorship, shadowing, and pair programming for developers only.

5.2. WORK PLAN

5.2.1. Work Activities

The following work activities and tasks are to be completed as the *Winfra* project progresses. A task's status is marked as complete (C), in progress (P), or to be completed (T). Projected dates are italicized.

WBS	Activity	Task	Assigned	Start Date	End Date	Duration	Status
0.0	Preliminary						
0.1		Requirements Specification (SRS)	Team	9/25/24	11/14/24	20 days	C
0.2		Project Management Plan (SPMP)	Team	11/7/24	11/27/24	20 days	C
1.0	Software Design						
1.1	Software Design Description (SDD)		Team	*1/21/25*	*2/5/25*	*16 days*	T
1.1.1		Write SDD	Team	*1/21/25*	*2/3/25*	*13 days*	T
1.1.2		Review SDD	Team	*2/4/25*	*2/5/25*	*2 days*	T
2.0	Construction						
2.1	Map Construction						
2.1.1.		Map programming	SO, EW	*2/6/25*	*2/13/25*	*7 days*	T
2.1.2		Testing Map Features	SO, EW	*2/14/25*	*2/20/25*	*6 days*	T
2.2	Post/ Comment Construction						
2.2.1		Post/Comment Programming	RM	*2/21/25*	*2/28/25*	*7 days*	T

2.2.2		Map and Post/Comment Integration	RM	*3/1/25*	*3/3/25*	*2 days*	T
2.2.3		Testing Post/Comment Construction	RM	*3/4/25*	*3/7/25*	*3 days*	T
2.2.4		Post/Comment Deployment	Team	*3/8/25*	*3/9/25*	*1 day*	T
2.3	CTA and Accessibility Integration						
2.3.1		CTA and Accessibility Programming	JB	*3/10/25*	*3/17/25*	*7 days*	T
2.3.2		CTA and Accessibility Integration with Posts/Commen ts	JB	*3/18/25*	*3/20/25*	*2 days*	T
3.0	Review / Quality Assurance						
3.1	Application meets requirements	Compare application to SRS	EW	*3/23/25*	*3/24/25*	*1 day*	T
3.2	Features Functioning as Intended	Test features to ensure working	SO, RM	*3/25/25*	*3/31/25*	*6 days*	T
3.3	Useability Assessment	Ensure features are intuitive and accessible	JB	*4/1/25*	*4/2/25*	*1 day*	T
4.0	Testing						

4.1	Local Server Testing	Run server on localhost and ensure features work as planned	JB	*4/3/25*	*4/7/25*	*4 days*	T
4.2	Cloud Server Testing	Run on server and ensure features work as planned	RM	*4/8/25*	*4/12/25*	*4 days*	T
4.3	User Testing with Control Variables	Test application features while application runs on server	Team	*4/13/25*	*4/17/25*	*4 days*	T
4..4	*Winfra* Application Testing	Verify applications run as intended and meet quality standards	SO, EW	*4/18/25*	*4/22/25*	*4 days*	T
5.0	Deployment						
5.1		*Winfra* Application Deployment	Team	*4/23/25*	*5/1/25*	*8 days*	T

5.2.2. Schedule Allocation

The scheduling of the *Winfra* project as detailed below will be monitored throughout the life cycle of the project to assess progress and efficiency. Time is allocated to milestones with the knowledge that work is to be completed on a part-time basis.

Project Milestones	**Start Date**	**End Date**	**Dependencies**
System Design Completion	January 21st, 2025	February 5th, 2025	Requirements finalized
Interactive Map Implementation	February 5th, 2025	February 19th, 2025	System design finalized
Posting/Commenting Features Integration	February 19th, 2025	March 12th, 2025	Interactive map successfully integrated
CTA and Accessibility Information Integration	March 12th, 2025	March 26th, 2025	Posting/Commenting features successfully integrated
Quality Assurance Testing	March 26th, 2025	April 9nd, 2025	SQA group, All features implemented
Beta Testing and Feedback Collection	April 9nd, 2025	April 23th, 2025	Successful QA, Potential end users acquired
Final Deployment and Launch	April 23th, 2025	May 7th, 2025	Successful testing

5.2.3. Resource Allocation

For resource allocation, the team has to allocate personnel to various work tasks in order to meet our project goals. On the team, there are four engineers, where each team member focuses on frontend and backend programming - therefore in order for the development team to stay on track tasks designated for frontend vs. backend development must be noted and given to the appropriate team member. Based on the team, the experience level ranges from levels 1-5 from lowest to highest on specific skills that span the development cycle, where in order to maximize the development speed of the projects the team hope to maximize the most qualified for the tasks while limiting each developer to a fixed number of tickets based on a given week. Furthermore, the team has access to software tools such as Jira, GitHub, SQL Server, coding frameworks, testing tools such as CircleCI, and productivity tools such as Notion.

Work Tasks	Team Member	Cloud/Compute Resources	Software Tools
System Design Completion	All team members	N/A	Draw.io
Interactive Map Implementation	Sanjida Orpi, Elijah Wilson	GitHub	IDE, SQL Server, ReactJS, JavaScript
Posting/Commenting Features Integration	Ruthvik Mukkamala	GitHub	IDE, SQL Server, ReactJS, JavaScript
CTA and Accessibility Information Integration	Jenesis Blancaflor	GitHub	IDE, SQL Server, ReactJS, JavaScript
Quality Assurance Testing	Ruthvik Mukkamala	GitHub	Jest, Selenium, IDE
Beta Testing and Feedback Collection	Sanjida Orpi, Elijah Wilson	GitHub	Notion

Final Deployment	All team members	GitHub Deployment/Docker	Jira, Notion

5.2.4. Budget Allocation (To be completed in a future delivery)

5.3. CONTROL PLAN

5.3.1. Requirement Control and Traceability

Throughout the agile software development process, requirements are bound to change. Therefore in order to ensure the software development process continues to build for and accomplish the requirements, we will manage the software development on an agile software development toolkit such as Atlassian's Jira to denote changes to tasks and milestones throughout the process. As the requirements change, the changes must reflect with the scheduling and budget planning which can be accomplished within Jira's Budget Tracker tool. For prototyping and modeling new requirements, there will be some time to ideate with the engineering team and develop a simple MVP. Traceability and impact analysis are also provided within the Jira software tool.

5.3.2. Schedule Tracking and Adjustment

The team will use tools provided by Jira to help with schedule tracking such as sprint planning meetings to define workflows once milestones have been decided by project manager(s). Furthermore, following the agile software methodology, the daily standup meeting will help the team touch base on challenges and changes required to stay on track to finish tasks for the sprint. When the schedule is adjusted, the tickets are moved to a proper sprint or shifted to different personnel within the team to ensure the team is on track to accomplish the milestones.

5.3.3. Budget Tracking and Adjustment (To be completed in a future delivery)

5.3.4. Quality Control

In order to ensure that the software follows specific quality control standards, the team has an individual on the team that generates testing frameworks for the software product—since the team is building a web application, there are many accessible software/application testing frameworks such as Jest or Puppeteer that ensure the intended functionality is accomplished. Within the CI/CD pipeline on GitHub there are code linter tools that are included to make sure the code is clean. Furthermore, once the software is deployed, engineers on the team will complete testing user flows as well as alternative flows. Once the flows are verified to work, then they can be moved into production, otherwise the engineers will need to create new tickets to fix the flow.

5.3.5. Reporting Mechanisms (To be completed in a future delivery)

5.3.6. Metrics Collection Plan

To collect the metrics of the project, for example items such as the defect density or code coverage—tools such as GitHub and Jira collect detailed information about the project metrics that are shared with the development team. Specifically in Jira, there are project dashboards that enforce data privacy and integrity since we are able to verify the code coverage based on function coverage, statement coverage, branches, condition, and line coverage all within the Jira platform. Furthermore, in order to ensure that the software requirements are completed on time, the team should frequently look at the metrics collection every sprint.

5.4. RISK MANAGEMENT PLAN

The risk management plan will provide a framework for stakeholders to identify and assess the project risk factors. The contingency planning procedures begin after risks are identified, and tasks are prioritized. The contingency planning methods used to track and evaluate the risk factors include testing and reassessing the plan, sharing the contingency plans with members of the software quality assurance group and all stakeholders. Our plan for assessing initial risk factors and the ongoing identification include identifying risk hazards, indicating the potential audience impacted by the vulnerabilities, assessing the likelihood of the risk occurring, the severity of its impact, and scoring its risk level. Throughout the project lifecycle, review updates will be conducted to monitor risks as the product changes and features are added, as the risk conditions will change, and new mitigation strategies will need to be evaluated.

5.5. POST-IMPLEMENTATION PLAN ***(To be completed in a future delivery)***

6. TECHNICAL PROCESSES

6.1. PROCESS MODEL

The *Winfra* project will use an object-oriented process model with an iterative life cycle. To ensure iterative development and that all aspects of the product are met, an agile technical method will be implemented for this project. This will include frequent collaboration with stakeholders that will allow the team to quickly respond to any changes or developments that are to be modified or added. Changes in requirements will necessitate revision of the requirements specification. Additionally, to ensure that the site is appealing and intuitive for users, we will also implement user-centered design and ensure their feedback is used for further iteration and development.

The flow of work products in the project focus on system design that will guide the development of features like the interactive map, posting/commenting functions, and CTA integrations. Each phase will include key work products, such as wireframes, prototypes, and test cases, delivered frequently to align with the project's agile model. Reviews will be conducted throughout the project, including stakeholder reviews for requirements and designs, peer reviews for code quality, and technical reviews for system functionality. Major milestones such as System Design Completion, Interactive Map Implementation, Posting and Commenting Features Integration, CTA and Accessibility Information Integration and Quality Assurance Testing will serve as checkpoints to measure progress. Baselines will be established for requirements, design specifications, and the website's core functionality to ensure consistency throughout development. Deliverables, including user manuals, and testing scripts, will require stakeholder approvals at each phase, ensuring alignment with project goals and readiness for deployment.

6.2. METHODS, TOOLS, AND TECHNIQUES

The development of the system will follow an agile methodology to ensure an iterative and flexible approach. The prioritization of agility will ensure incremental delivery of features, regular feedback from stakeholders, and quick adaptation to changes, so that the status of the project remains aligned with its goals.

For development, we’ll use JavaScript, CSS, and HTML for the front end, to design a responsive and interactive user interface. The back end will be built with Python, utilizing server-side logic and API integrations. PostGreSQL will be used to manage the system’s databases, ensuring secure and efficient storage of user reports, comments, and other data. The map feature will utilize a map API, such as Google Maps or Mapbox, to allow users to pinpoint and report infrastructure issues directly on an interactive map interface.

The tools and techniques will include GitHub for version control, ensuring effective collaboration and tracking of changes for example within the CI/CD pipeline there will be tools such as Selenium to help with test automation and deployment.

Documentation of all aspects of the project will ensure clarity and ease of maintenance for future updates. We will comply with security policies to protect user data, such as encrypted data storage and secure API communications. The system will also follow accessibility standards, providing features like keyboard navigation, high-contrast modes, and descriptive alt text to make the platform inclusive for all users. Additionally, data privacy policies will ensure compliance with regulations to safeguard user information and provide transparency about data usage.

The product acceptance plans will begin with smoke testing to evaluate the core features and fix any bugs we found in the code. After that, there will be user and performance testing among a small focus group to get their feedback on how intuitive the website is to use and how well it performs. At every major milestone, the team will perform regression testing and work with stakeholders to make sure the product is on track and aligned with the project's goals. Each feature, like the interactive map and reporting tools, will need to meet clear standards for performance, security, and accessibility. Sticking to this plan, will ensure the final product is reliable, easy to use, and ready to make a real impact.

6.3. INFRASTRUCTURE PLAN

To accomplish cross-platform functionality which requires substantial infrastructure planning prior to the execution of the software project. The website will have a responsive design to support devices with varying screen sizes, including desktops and mobile devices. Therefore, in order for the project requirements to be accomplished, the engineering team will need to have at least

8GB of RAM to run the testing environments and integrated development environment (IDEs) for the coding process.

The most essential network resource for the project is a stable internet connection to run and host tools on cloud platforms such as GitHub as well as calling APIs for the development process. For the project to be coherently completed, the team will use Git for version control. Security policies will be utilized to ensure user data is securely saved during the development of the application. These include the use of secure network connections, maintenance of encryption protocols, and denial of service to vulnerable environments and adversarial users. Data management policies will govern how user data is handled.

The team expects to use HTML, CSS and Javascript for UI/UX design, automated testing tools such as Jest and Selenium, and device testing (BrowserStack) - and finally Jira to be a project management tool.

6.4. PRODUCT ACCEPTANCE AND MIGRATION PLAN *(To be completed in a future delivery)*

7. SUPPORTING PROCESSES PLANS

7.1. CONFIGURATION MANAGEMENT PLAN

Since the inception of the *Winfra* project, the team has kept an archive of relevant documents and files in addition to the project code of the web application. As demonstrated by previous releases, consecutive releases of documentation will be demarcated by the incrementation of the version number and the document number. The documentation of the *Winfra* project has been developed in environments that track changes and make version history accessible.

The *Winfra* project code is to be housed in a GitHub repository, and standard GitHub conventions will be used for the configuration management of the project, including the frequent use of branches and pull requests. Versions of the project will be demarcated by the features that they aim to integrate or functional requirements that they aim to meet. Status accounting will be performed through the description of proposed changes in the repository.

The team will meet after the completion of deliverables and work products to establish baselines. The following work products are currently planned to serve as baselines:

- Software Requirements Specification (SRS)
- Software Project Management Plan (SPMP)

- Design Description (SDD)
- Project Source Code
- Project Tests
- Incremental releases of the *Winfra* web application

7.2. QUALIFICATION (VERIFICATION AND VALIDATION) PLAN

The scope of this qualification plan is to ensure the Winfra system meets all requirements. We will conduct verification and validation through the use of tools such as Jest for automated testing, to ensure that features like the interactive map and reporting tools are functional as outlined. We will also use prototyping techniques to allow for early testing and refinement of key features and modeling techniques will be used to verify system behaviors. The system's preparation for testing will be handled by the development team, the verification and validation of the system will be supervised by the quality assurance, the review of milestones and feedback approval will be provided by stakeholders. Traceability checks will be completed to link requirements to deliverables. Milestone reviews will be conducted to monitor progress on the project milestones, consulting the task deadlines on the schedule. Walkthroughs and peer reviews will be conducted to ensure code and design quality. Prototyping and simulation testing will be used to improve features like the location-based reporting system throughout its development.

The validation plan focuses on confirming the system’s ability to operate as expected for all end users. This will involve unit, integration, system, and acceptance testing to identify and resolve any issues. Demonstrations will highlight critical features, such as the map interface and CTA workflows, to stakeholders. Detailed analysis and inspections will be conducted to ensure that the final product meets all quality, functionality, and usability standards outlined in the project requirements, ensuring the application effectively addresses infrastructure reporting needs.

7.3. DOCUMENTATION (LIBRARY) PLAN

The requirements for the documentation/library ensure that the code developed by the team can be used for future development, in case the team grows or the code changes hands. Emphasizing the critical role of organized and traceable documentation promotes consistency and readability throughout the life of the project. Therefore, the plan includes generating both internal and deliverable work products, such as architectural descriptions for each unit of software that is developed so that engineers are mapped to where changes need to be made. Furthermore, design specifications, traceability metrics, source code, and manuals, all of which are vital for maintaining clarity throughout the development lifecycle, are to be developed. There will also be focus on defining distribution lists and maintaining up-to-date review for agile software development. By implementing a documentation plan, both internal and external stakeholders and engineers are able to refer to well-defined processes and deliverables.

7.4. QUALITY ASSURANCE PLAN

The quality assurance plan is designed to ensure that both the development process and the final product meet the requirements, standards, and guidelines specified in the project documentation. This plan focuses on delivering a reliable, user-friendly system that enables residents to report and address infrastructure issues efficiently while adhering to the highest quality standards.

QA procedures will include analyzing features like the interactive map, location-based reporting, and CTA workflows meet the system requirements and function as intended. Traceability metrics will be used to ensure all requirements are linked to deliverables, confirming that no functionality is overlooked. Regular inspections of the system architecture, design documents, and codebase will help identify potential issues early and maintain alignment with quality standards.

Additionally, assessments will be conducted throughout the project lifecycle to evaluate key deliverables. For example, system functionality will be tested against performance benchmarks during integration and system testing. Usability assessments will ensure that features like the pin-drop reporting mechanism and user dashboards are intuitive and accessible. QA activities will also evaluate the scalability and reliability of the system to handle increased usage over time.

7.5. REVIEWS AND AUDITS

The project will follow a structured plan for reviews and audits to ensure quality, mitigate risks, and maintain alignment with project goals. We will conduct these activities regularly throughout the project lifecycle of our agile process, focusing on key milestones and deliverables. Organizational management reviews will be conducted at major milestones, such as finalizing requirements, approving designs, and deploying the system. Stakeholders will evaluate progress, approve deliverables, and provide feedback for the next project phase.

Developer peer reviews will occur weekly, focusing on reviewing code contributions to ensure they meet coding standards and project requirements. These reviews will help identify issues early, improve code quality, and validate the implementation of critical features like the location-based reporting system and map interface. Technical reviews will be held at the end of each development sprint, where team leads and architects will assess the technical implementation of features such as the map API integration, backend database functionality, and CTA workflows. These reviews will ensure that the system is scalable, efficient, and aligned with technical specifications.

Walkthroughs and inspections will be scheduled bi-weekly to examine design documents, system architecture, and test plans. The team will focus on identifying potential issues in usability, security, and performance, ensuring that the interactive map and reporting tools are reliable and intuitive for users. Detailed inspections will verify compliance with requirements, such as protecting user location data and adhering to accessibility standards.

Formal audits will be conducted at the end of each project phase by the SQA team to evaluate compliance with standards, policies, and security guidelines. These audits will assess the system performance, accessibility, and data security, providing detailed reports with recommendations for improvement. Tools like GitHub (for code reviews), Jira (for tracking issues), Selenium & Jest (for testing) will be used to streamline the review and audit processes.

7.6. PROBLEM RESOLUTION PLANS

Resources for problem resolution will include dedicated time from the development team, configuration management team, and the software quality assurance SQA team. Stakeholders will also provide input during reviews to help prioritize critical issues. Tools such as Jira or another issue-tracking platform will be used to document and manage problem reports, track their resolution status, and assign responsibilities.

Methods and procedures will begin with problem reporting, where team members or SQA personnel document issues identified during testing, reviews, or stakeholder feedback. Each problem will be logged in the issue-tracking system. Problems will then be analyzed by the development team to determine root causes and assess solutions. The team will meet, review and approve solutions for high-impact problems before implementation.

Developers will focus on resolving assigned problems and ensuring fixes are integrated into the system without introducing new issues. The configuration management team will oversee version control and ensure that changes are properly documented and tested before deployment. SQA personnel will certify that resolved problems meet the required quality standards through regression testing and validation.

7.7. ENVIRONMENT MANAGEMENT PLANS *(To be completed in a future delivery)*

7.8. PROCESS IMPROVEMENT PLAN *(To be completed in a future delivery)*

8. ADDITIONAL PLANS *(To be completed in a future delivery)*

9. INDEX

No index at this time.

10. RATIONALE

No rationale at this time.

11. NOTES

No notes at this time.

12. APPENDICES

12.1. SCHEDULE TRACKING

Hours

Artifact or Deliverable	Who (individual or Team)	Estimated	Actual	Difference
SPMP	Jenesis Blancaflor	5 hrs	6.5 hrs	+1.5 hrs
	Ruthvik Mukkamala	4 hrs	4 hrs	+0 hrs
	Sanjida Orpi	5 hrs	7 hrs	+2 hrs
	Elijah Wilson	3 hrs	4 hrs	+1 hr
	Summary for the Entire Team	17 hrs	21.5 hrs	+4.5 hrs

12.2. DEFECT TRACKING

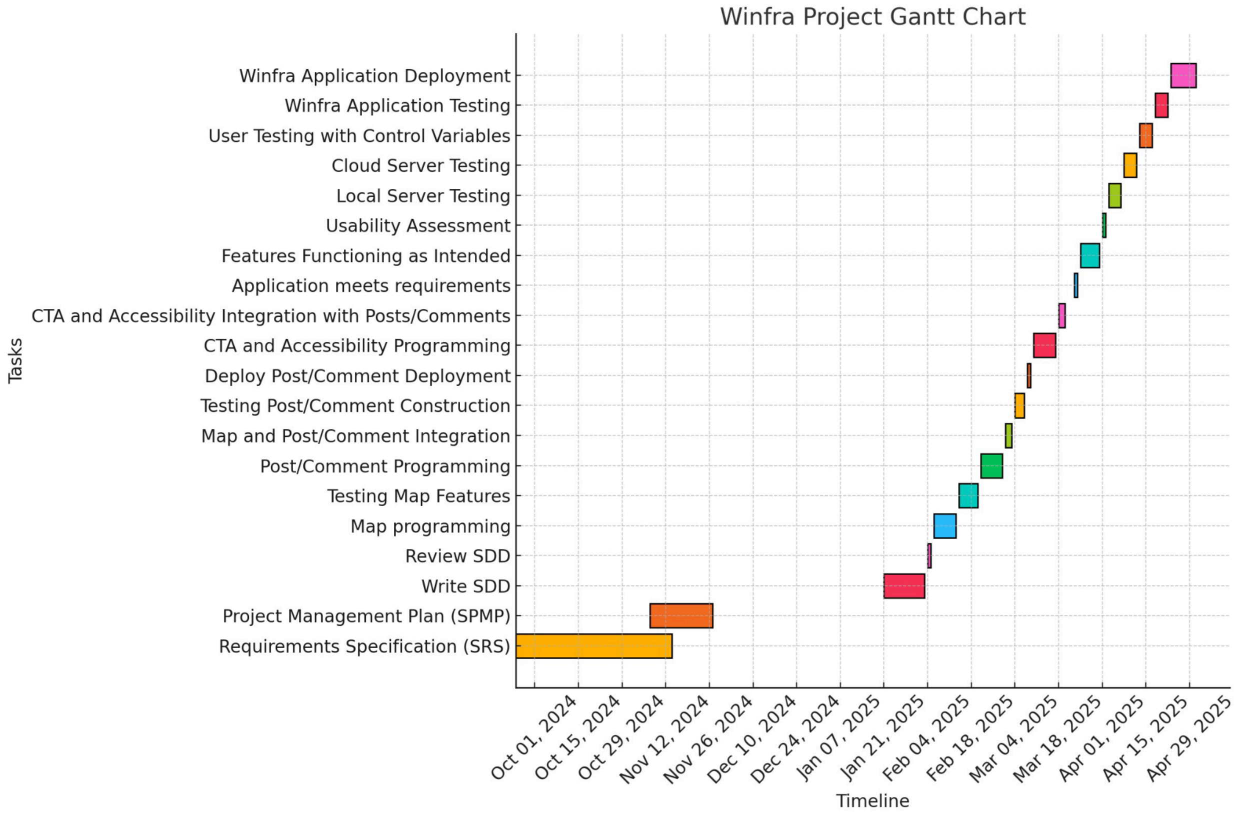
Counts

Artifact or Deliverable	Who (individual or Team)	Estimated	Actual	Difference
SPMP	Jenesis Blancaflor	4 counts	5 counts	+1 counts
	Ruthvik Mukkamala	3 counts	5 counts	+2 counts
	Sanjida Orpi	4 counts	5 counts	+1 counts
	Elijah Wilson	3 counts	5 counts	+2 counts

	Summary for the Entire Team	14 counts	20 counts	+6 counts

12.3. GANTT CHART / MICROSOFT PROJECT SCHEDULE

Below is the Gantt chart for the work tasks of the *Winfra* project. The chart visualizes the project schedule that has been described earlier in the plan (§5.2.1).


Winfra Project Gantt Chart
Tasks
Winfra Application Deployment
Winfra Application Testing
User Testing with Control Variables
Cloud Server Testing
Local Server Testing
Usability Assessment
Features Functioning as Intended
Application meets requirements
CTA and Accessibility Integration with Posts/Comments
CTA and Accessibility Programming
Deploy Post/Comment Deployment
Testing Post/Comment Construction
Map and Post/Comment Integration
Post/Comment Programming
Testing Map Features
Map programming
Review SDD
Write SDD
Project Management Plan (SPMP)
Requirements Specification (SRS)
Oct 01, 2024
Oct 15, 2024
Oct 29, 2024
Nov 12, 2024
Nov 26, 2024
Dec 10, 2024
Dec 24, 2024
Jan 07, 2025
Jan 21, 2025
Feb 04, 2025
Feb 18, 2025
Mar 04, 2025
Mar 18, 2025
Apr 01, 2025
Apr 15, 2025
Apr 29, 2025
Timeline