Project Management Plan

for

Theia

Prepared by:
Pallavi Sharma
Prasanth Athaluri
Garhgaj Singh
Afoke Abogidi

Submitted for:
Software Requirements
CPT_S 584 Phase 1
Instructor: Bolong Zeng, PhD.
Washington State University
December 11th, 2022.

Revision History

Name	Date	Reason for Changes	Version
Preliminary Plan	09-18-2022	initial draft	1.0
Updated Preliminary Plan	10-15-2022	Added new deliverables	1.1
Final Plan	12-11-2022	Added new deliverables	1.2

1. Introduction

1.1. Project Overview

This project aims to create a navigation assistant as a mobile application to assist blind and visually impaired individuals with indoor navigation. The application will support the individuals to navigate within the same building as well as to different buildings that are connected to each other. The functionalities that should be present in the application can be broadly classified into 4 distinct parts: Destination Selection, Route-finding, Guided Navigation and Emergency Services.

Destination Selection focuses on selecting the destination where the individual wants to arrive at.

Route-finding is primarily concerned towards providing the best route for navigation. The guided navigation ensures a smooth and comfortable journey to the destination. Emergency Services includes the features that are available for the individual in case of injury or health-concerns.

1.2. Project Deliverables

Deliverable	Delivery Date	Delivery Method	Comments
Project Phase I: Preliminary Project Plan	Sept 18th, 2022	Gitlab & Canvas	Assigned to Pallavi Sharma
Project Phase I: Final Submission (Work Requirements Specification Document)	Oct 16th, 2022	Gitlab & Canvas	Assigned to Afoke Abogidi
Updated Preliminary Plan	Oct 16th, 2022	Gitlab & Canvas	Assigned to Pallavi Sharma
Meeting Records	Oct 16th, 2022	Gitlab & Canvas	Assigned to Afoke Abogidi
Presentation Slides	Oct 16th, 2022	Gitlab & Canvas	Assigned to Garhgaj Singh

WRS Basic User Manual	Oct 16th, 2022	Gitlab & Canvas	Assigned to Garhgaj Singh
Mock Prototype	Oct 16th, 2022	Gitlab & Canvas	Assigned to Prasanth Athaluri
Project Phase II: Final Submission	Dec 11th, 2022	Gitlab & Canvas	Assigned to Pallavi Sharma
Final Plan Document	Dec 11th, 2022	Gitlab & Canvas	Assigned to Pallavi Sharma
Process Specification Document	Dec 11th, 2022	Gitlab & Canvas	Assigned to Afoke Abogidi
Presentation Slides for Deliverable II	Dec 11th, 2022	Gitlab & Canvas	Assigned to Prasanth Athaluri
Final WRS Document	Dec 11th, 2022	Gitlab & Canvas	Assigned to Garhgaj Singh

1.3. Evolution of SPMP

This document is a preliminary project documentation based on the initial research and business requirements. It lays out our expectations and initial plan of action. Revision of this document due to any scheduled or unscheduled changes are subject to approval from the Project Manager. The final version of this document as well as the revision history will be uploaded to the team’s Gitlab repository.

1.4. Reference Materials

BlindSquare. (n.d.). *Pioneering accessible navigation – indoors and outdoors*. Retrieved from Blindsquare.com: https://www.blindsquare.com/

National Federation Of The Blind. (n.d.). *KNFB Reader*. Retrieved from National Federation Of The Blind: https://nfb.org/programs-services/knfb-reader

Karl E Wiegers and Joy Beatty. 2013. Software Requirements 3. Microsoft Press, USA.

360Logica. (2022). *Agile Development – Advantages, Disadvantages and when to use it?* Retrieved from 360Logica: https://www.360logica.com/blog/agile-development-advantages-disadvantages-and-when-to-use-it/

Jin, Z. (2018). Requirements Engineering Methodologies. In *Environment Modeling-Based Requirements Engineering for Software Intensive Systems* (pp. 13-27). Elsevier Inc. Retrieved from https://www.sciencedirect.com/science/book/9780128019542

Daisuke Sato, Uran Oh, João Guerreiro, Dragan Ahmetovic, Kakuya Naito, Hironobu Takagi, Kris M. Kitani, and Chieko Asakawa. 2019. NavCog3 in the Wild: Large-scale Blind Indoor Navigation Assistant with Semantic Features. ACM Trans. Access. Comput. 12, 3, Article 14 (September 2019), 30 pages. https://doi.org/10.1145/3340319

Meliones A, Sampson D. Blind MuseumTourer: A System for Self-Guided Tours in Museums and Blind Indoor Navigation. Technologies. 2018; 6(1):4. https://doi.org/10.3390/technologies6010004

1.5. Definitions and Acronyms

Agile: Method is really flexible to use where we can incorporate any changes during any point while working on a project. Using this method one is able to experiment with new plans and directions.

Application: A program designed to perform a certain task or a function for the end user.

Android: Open source operating system designed primarily for mobile devices.

Sprint: A short period of time slated to complete a set of tasks.

2. Project Organization

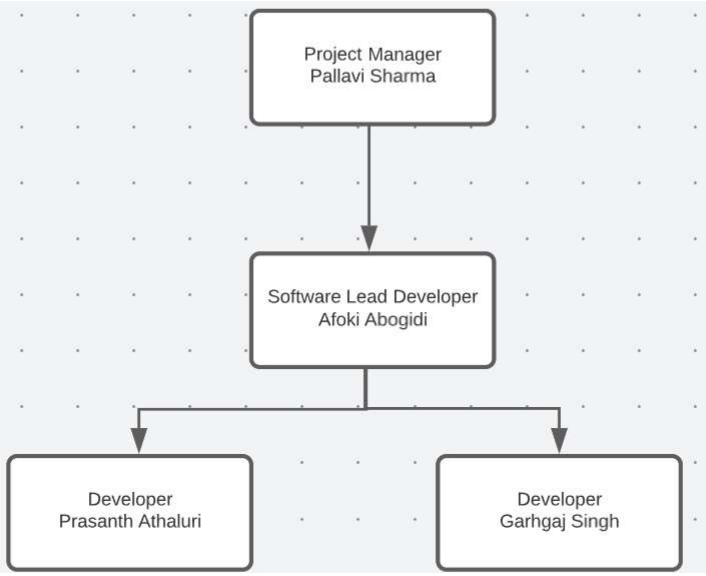
2.1. Process Model

In order to attain frequent feedback from the customer, we will be utilizing the Agile Software Model to facilitate the development of the mobile application. The development would be conducted in shorter cycles called sprints. Each sprint would consist of Development, Testing, Deployment and Review phases. Each sprint would typically comprise of 14 days (i.e. 2 weeks). To keep track of the sprints and deliverables, we will be using the Project Management Tool- JIRA.

2.2. Organizational structure

The members involved in this project are: Pallavi Sharma, Prasanth Athaluri, Garhgaj Singh and Afoke Abogidi. Our Project Manager will be Pallavi Sharma. The rest of the members will be developers with Afoke Abogidi being the Software Lead Developer. Pallavi will ensure that the team is on track and each developer on the team will have to complete specific milestones and tasks.

The image below shows the organizational hierarchy:



2.3. Organizational Interfaces

Team will conduct a meeting once a week to communicate and collaborate any new thoughts and ideas toward the project. To communicate daily regarding any updates, the team will be using discord. Team will also be using Google Drive and Gitlab to share files and documents. Communication between mentors, instructors and the team would occur through the Project Manager (Pallavi Sharma), primarily through emails and occasionally via zoom video calls.

2.4. Project Responsibilities

The team members would be equally involved in all the phases of the project development lifecycle. Please see table below for a breakdown of the project's roles and responsibilities.

Phase	Description	Tasks	Assignee
Phase 1	Preliminary Plan	Information Gathering and Analysis.	All team members
		Communication with the stakeholders	Project Manager
		Requirements Elicitation	All team members
		Technical Resource	Developers

		Requirements	
		Workload Scheduling	Software Lead Developer
Phase 2	Software Requirements Specification (SRS)	Communication with the stakeholders	Project Manager
		Preparing Requirements Specification	Software Lead Developer
		Creating Technical Design	Developers
		Developing Prototype	All team members
Phase 3	Final Submission	Communication with the stakeholders	Project Manager
		Project Presentation	All team members
		Revised preliminary plan	All team members
		Meeting records collection	Project Manager
		Final Requirement Specification	Software Lead Developer

3. Managerial Process

3.1. Management objectives and Priorities

The management objectives of this project are classified into four key groups:

3.1.1. Requirement Management objective

This objective focuses on managing the scope of the project to ensure that the business requirements as well as the requirements of all stakeholders are properly captured, evaluated and incorporated into the project plan. The development team would have a requirement elicitation meeting to clearly analyze user stories and define clear functional and non-functional requirements. These requirements would be broken into phases/milestones for proper structuring. Tasks would be created using the Jira tool for easy tracking and visibility of the project workloads.

3.1.2. Configuration management objective

This objective oversees the process of changes to the software deliverable product. It defines, analyzes, controls, tracks and audits a change to the deliverable in the project. A change request initiates the process of a change to the product deliverable. This is followed by a team discussion to analyze and evaluate the impact of the change before it can become an official request. The new requirement is added to the project tasks in Jira and after implementation, an official pull request is required before a modification to the existing deliverable is fully made. Using the gitlab source control tool, changes go through a pull request and approvals from all developers are required before a merge.

3.1.3. Communication management objective

This objective highlights the processes, tools and types of communication anticipated throughout all phases of the project. Tools for communication as mentioned earlier include, email, zoom video calls, and discord chat. Please see table below for specific communication types and tools between all organizational interfaces for this project.

Organizational boundary/interface	**Tools**	**Type of communication**
Between Instructor/mentor and development team	Email, Zoom calls	Approvals, questions, clarifications.
Between Customer and development team	Email, Zoom calls	Clarifications, questions
Within the development team	Email, zoom calls. Discord chat.	Clarifications, elicitations, comprehension, questions, collaboration.

3.1.4. Resource management objective

This objective documents the use of resources throughout the project. These resources include human, tools, workspaces and how these resources would be allocated, managed and released. Human resources for this project is made up of the entire development team defined in section 2.2.

Management tools for this project include Jira for workload prioritization and visualization, Gitlab and Git for source control and Google docs for document sharing.

3.2. Assumptions, Dependencies and Constraints

3.2.1. Assumptions

- It is assumed that the user will be using this application from a smart phone having all the basic utilities and functionalities.
- It is assumed that the user will be providing access to their location when using the application.
- It is assumed that the user will possess a stable internet connection while utilizing the application.
- It is assumed that the user will use a registered account to log in and use the services.

- It is assumed that the user will be using either an IOS based or an Android-based phone for accessing the application.

3.2.2. Dependencies

- Users must have moderate/strong internet connections to use the app.
- Well documented user manuals are needed to ensure the safety of users
- Users must have a phone either IOS or Android
- Development team should have enough time and members to complete all the tasks assigned to them.

3.2.3. Constraints

- Operating System: The mobile application has limitations on the operating systems it can work on.
- Language: The Application will be communicating with users in English only.
- Time: Prototype for this project has to be done by the end of semester and all document submissions should be done before the deadlines.
- Budget: Budget for this application is limited.

3.3. Risk Management

No	Risk	Type	Likelihood	Description
1	Tools and technical components are unable to meet specific version requirements	Technical	Likely – High potential impact	As newer OS versions keep being deployed they do not have compatibility with existing framework and libraries.
2	Inadequate planning	Managerial	Unlikely – High potential impact	Failure to account for all the parameters that will affect deliverable date.
3	Unable to estimate resource requirements	Managerial	Unlikely – High potential impact	At a later stage of project the resources required for the project are not available or not enough
4	Additional requirements	Technical	Unlikely – High potential impact	For a feedback based application there will be changes that need to be

				added or made on existing core features
5	Loss of information due to external factors	Technical	Unlikely – High potential impact	The artifacts related to the project are lost due to factors such as damage to storage space or device.
6	Inefficient Taskforce	Managerial	Likely – High potential impact	Not being able to meet project milestones due to inactivity and negligence on the part of the team members

3.4. Monitoring and Controlling Mechanisms

No	Risk	Monitoring and Controlling
1	Tools and technical components are unable to meet specific version requirements	● Each version requirements should be clearly stated along with a list of possible changes. ● Suggestions on places for tools and libraries must be added
2	Inadequate planning	● Achievable goals must be set when scheduling dates for deliverables and milestones ● If a milestone is not complete on the target dates, an extension or an already prepared backup plan must be ready for use.
3	Unable to estimate resource requirements	● Efficient planning of the resources before the commencement of the project to ensure that the deliverables are not hampered due to the shortage of resources. ● Regular tracking of the deliverables and milestones to understand if there are any issues with the resource allocation.
4	Additional requirements	● Optimized code should be written in the areas where functional requirements are expected to change. ● Pre-allocating resources to such expected requirements.

5	Loss of information due to external factors	● A separate and secure back-up must be placed and should be accessed minimally and updated regularly . ● Finer management of storage devices
6	Inefficient Taskforce	● Expectations must be set before starting the project. ● Professional standards are to be strictly maintained.

4. Technical Process Plans

4.1. Methods, Tools and Techniques

For this project our team will be using Java as our main programming language. Java will be helpful in this project because it supports mobile applications using large amounts of data. Java is a server side language so will be useful to use for mobile application development projects such as this one.

For version control our team will be using GitLab and Git to make collaboration easier allowing members to make changes and merge into a single branch. With Git we can track changes and keep track of what's happening with the project.

We will also be using Google Docs in order to share files and documents. Using Google Docs, members are easily able to collaborate and make appropriate changes to keep the documents updated.

Jira is another powerful tool that we will be using for project management. With Jira we are easily able to monitor productivity as well as keep track of time in order to successfully finish upcoming tasks. We can also add team members with specific tasks and see if they are able to finish it on time and come up with a real time performance report.

4.2. Software Documentation

The software documentation created during the course of the project will be stored on Google Drive as well as Gitlab. The storage on Gitlab helps to access all the versions of the document. All the documents should be uploaded in pdf format. The table below provides greater insight into the documents created during the project.

Document	Template or Standard	Created By	Reviewed By	Target Date	Storage
Software Project	Template	Project Manager and	Stakeholders	Sept 18th, 2022	Google Drive & Gitlab

Management Plan (SPMP)		Requirement Analyst			
Software Requirements Specification (SRS)	Template	Project Manager	Stakeholders	Oct 8th, 2022	Google Drive & Gitlab
Software Design Description (SDD)	Template	Software Lead Developer	Project Manager	Oct 18th, 2022	Google Drive & Gitlab
Software Test Plan	Template.	Developers	Software Lead Developer	Oct 8th, 2022	Google Drive & Gitlab
User Documentation	Standard	Developers	Stakeholders	Dec 6th, 2022	Google Drive & Gitlab

The user documentation will be available in English and Spanish for the current version as the product is scheduled to launch in the United States followed by its global launch in 2024. Since our primary user is visually challenged, The user documentation will be available in Braille and audio format in addition to text notation.

5. Work Element and Schedule

Project should be completed by Dec 11, 2022 for the final submission. The following is the roadmap for the project:

5.1. Planning Phase (3 weeks)

- Requirement Gathering & Analysis (9/17 - 9/20)
- Preliminary Plan Document (9/17 - 9/20)
- Prototype Development (9/20 - 10/7)
- Creation of Prototype (9/26 - 10/4)
- SRS Document (10/4 - 10/7)
- Prototype Demo (10/7-10/8)
- Revision of SRS (10/7-10/8)

5.2. Design (2 weeks)

- Software Design Description (SDD) (10/14 - 10/18)
- Software Test Plan Documentation (10/14 - 10/18)
- Architectural Design model (10/19 – 10/21)
- UI User Flow Maps (10/22 – 10/26)

- Wireframe mockups (10/22 – 10/26)
- Requirements Validation (10/27 – 10/28)

5.3. Development (5 weeks)

- Infrastructure Development (1 week)
    - Database design and creation (10/29 - 11/03)
    - Data Preprocessing (11/04 - 11/05)
    - Server Connection setup (11/04 - 11/05)
- Visualization (1 week)
    - UI/UX Elements (11/06 - 11/09)
    - User Controls (11/06 - 11/09)
    - Validity Walkthrough/Demo (11/09 - 11/11)
    - UI Refinement (11/09 - 11/11)
    -
- Backend Development (3 weeks)
    - Request Validation (11/12 - 11/16)
    - User Authentication (11/12 - 11/16)
    - Data Processing (11/16 - 11/21)
    - Navigation Model (11/16 - 11/21)
    - Emergency Services (11/21 - 11/26)
    - Optimization (11/21 - 11/26)

5.4. Component Testing (1 week)

- Unit testing (11/27- 12/02)
- Requirement Validation (11/27- 12/02)
- Refinement (11/27-12/02)

5.5. Integration, System Testing and Deployment (1week)

- Functional Testing (11/29-12/02)
- Access management (12/02- 12/04)
- Final Refinement (12/02- 12/06)
- User Documentation (12/02- 12/06)

5.6. Final Demo (12/11)