
Project Management Plan

for

THEIA

Version 1.0 draft 1

**Prepared by Sam Gibson, Ahmed Mohamed, Julie Rodriguez,
Caden Oslund, Brendan Crebs**

Ice Bucket - WSU SE

12/10/2023

Table of Contents

| | |
|--|------------|
| Table of Contents | ii |
| Revision History | iii |
| 1. Overview | 1 |
| 1.1. Project Purpose, Objectives, and Success Criteria | 1 |
| 1.2. Project Deliverables | 1 |
| 1.3. Assumptions, Dependencies, and Constraints | 1 |
| 1.4. References | 2 |
| 1.5. Definitions and Acronyms | 2 |
| 2. Project Organization | 3 |
| 2.1. Process Model | 3 |
| 2.2. Organizational Structure | 3 |
| 2.3. Roles and Responsibilities | 3 |
| 3. Managerial Process Plans | 4 |
| 3.1. Management objectives and priorities | 4 |
| 3.2. Assumptions, dependencies, and constraints | 4 |
| 3.3. Risk management | |
| 3.4. Monitoring and controlling mechanisms | |
| 4. Technical Process Plans | 5 |
| 4.1. Methods, tools, and techniques | 5 |
| 4.2. Software documentation | 5 |

Revision History

| Name | Date | Reason for Changes | Version |
|-------------|-------------|------------------------------|----------------|
| All | 9/17/23 | Initial draft | 1.0 draft 1 |
| All | 10/15/23 | Revision | 1.0 |
| Sam | 11/30/23 | Update of Managerial section | 1.1 draft 1 |

1. Overview

The motivation behind this project is to allow visually impaired or blind individuals to navigate indoor spaces safely and efficiently. This will provide these individuals with the capability to move from one location to another within buildings and enhance their mobility of navigating within indoor spaces.

The objective in this project is to develop an application that will allow for indoor navigation tailored to fit the needs of blind people which is set to include the following: Enabling the user to set their current location within a building, a GPS that provides step-by-step walking instruction, object detection to avoid collision, and optimized routes within the building provided by GPS.

The success criteria will be based on the following: accuracy of indoor location, effectiveness of object detection, optimized routes, and the successful navigations achieved by the user.

Major deliverables/milestones will include the documentation on functional and nonfunctional project requirements, and a detailed technical specification on the application. Another major deliverable can include a possible prototype of the app that includes a modern UI, safety features with collision detection, and route optimization. Rigorous testing will take place along with user documentation and instruction on how to use the app and finally a comprehensive report detailing the results of the project and the findings of the application.

The constraints of the project is to include accessibility - this may include compatibility with screen readers and must work on smartphones. The application should also ensure user privacy and security and adhere to the predefined budget as well as follow the timeline set for the software development life cycle.

Required resources will include the development team consisting of software engineers, UX/UI designers, and accessibility experts. The other required resources will be a smartphone for testing, software dev tools and frameworks including indoor mapping technologies and user testers or accessibility consultants.

The schedule for this application is to be completed and deployed within 6 months.

1.1. Project Purpose, Objectives, and Success Criteria

The purpose of this project is to create an application to aid visually impaired and blind people navigate indoors environments. The application allows users to navigate from one location to another within one building, or between multiple buildings. To achieve the overlaying task of the application, systems for determining user location, the destination location and the safest quickest route between both points. In addition, a system to notify the user of any real-time obstacle in route. Moreover, the application will include accessibility systems such as sound queues, vibrations, and vocal assistance to insure a customizable usability to users.

Overall, the success of the application will depend on several main criterias. First, the safety of the application, this can be achieved by extensive testing and safety first approach to development. Second, is the accessibility, the application will need to include multiple features to allow user control over their experience such as sound queues, vibrations and volume control.

Finally, the ability to track objects in real time, this will allow users to avoid collisions with unexpected objects.

1.2. Project Deliverables

| Deliverable | Delivery Date | Delivery Method | Comments |
|--------------------------------|----------------------|------------------------|-----------------|
| Preliminary Plan Document | Dec 10 | GitLab | |
| Vision and Scope Document | Dec 10 | GitLab | |
| Process Specification Document | Dec 10 | GitLab | |
| Powerpoint Presentation | Dec 7 | GitLab | |
| Prototype Application | Dec 7 | GitLab | |

1.3. Assumptions, Dependencies, and Constraints

We are assuming that the system only operates in indoor buildings. The application only works on smartphones with cameras and internet access. The user will be able to use sound and vibration as a means of interaction with the application. In addition, the application needs to provide real-time support as the user navigates using the application to avoid obstacles

1.4. References

This section is not filled because there are no references yet.

1.5. Definitions and Acronyms

We have not developed any project-specific words or acronyms as of the writing of the rough draft of this document.

2. Project Organization - Brendan Crebs

2.1. Process Model

The process model we will use is the agile framework. Specifically we will use the Extreme Programming(XP) model. Here are the development phases we will follow:

Planning: The first phase for the XP model would be the planning phase in which we will determine the scope and goals of the release. A part of this phase would be breaking down the release plan into smaller iterative cycles. This is part of the iterative nature of the XP model. The customer would be included in this step to assure our team's goals align with their requirements.

Design: The next phase would be design where we would determine implementation details to achieve the goals set out in the planning phase. According to the XP model principles we would focus here on pursuing the most simple design to avoid unnecessary complexity.

Coding: Following the completion of a satisfactory design we would begin the coding phase. Work during this phase would be highly collaborative and the results held to a high standard. Some core tasks to be included during this stage to maintain quality would include:

- a. Following a predetermined set of coding standards that will be enforced by linters and extensive code review.
- b. Including a continuous integration build server so code is constantly being built and tested.
- c. 100% code coverage from unit tests
- d. Pursuing simplistic implementation to avoid bloated breakable code.

Testing: After an implementation has been produced we would enter the testing phase. While unit tests will be completed in the coding phase, This phase would be for functional testing and regression testing. Regression tests will be conducted if we are releasing a new version of a previous implementation. Functional tests would validate that the implementation meets the requirements and standards of the customer.

Feedback: A significant part of the XP model is customer involvement in the development process. During this stage we will present our current implementation to the customer in order to receive feedback. Using this feedback we will determine the path forward for the project. This could include starting a new iteration to address the feedback with a new version or to move on with deployment if the customer is satisfied.

Deployment: At this stage we would make the changes necessary so that the application can be packaged and distributed to customers.

Maintenance: In this stage we would be maintaining the deployed application by listening to any customer complaints and also frequently reviewing the code for bugs, vulnerabilities, depreciated dependencies, etc.

Each of these phases can be, and are likely to be revisited frequently throughout the development lifecycle. Customer communication will also be consistent throughout each phase so that there is less of a chance that issues over requirements will arise.

2.2. Organizational Structure

The team members that will be working on this project will be: Brendan Crebs, Sam Gibson, Ahmed Mohamed, Caden Oslund and Julie Rodriguez. Our communication liaison will be Sam Gibson. All members of the team will be developers that will be given roughly equal responsibilities. These responsibilities will overlap if necessary.

2.3. Roles and Responsibilities

Project Roles:

Project Manager, Communication liaison: Sam Gibson

Technical Lead: Ahmed Mohamed

Software Engineer: Julie Rodriguez

Test Engineer: Brendan Crebs

Documentation: Caden Oslund

3. Managerial Process Plans

3.1. Management objectives and priorities

To achieve our philosophy of efficiency and feasibility, the project will be separated into individual tasks assigned to each member of the team. Tasks will be assigned depending on both preference and proficiencies. Tasks will avoid being tightly coupled to each other in order to prevent delays from waiting on a task completion. Communications will be handled over Discord, involving meetings, resource posting, and progress reports.

Our team will prioritize implementation of core features to be as successful as possible, reevaluating remaining tasks upon each completion. Specifically, writing and documentation tasks will be done prior to coding tasks. This will include meetings and other forms of communication involving all team members.

Communications with the client will be handled by a liaison, who will share information with the best interest of the team in mind. Client communications will be presented to the team whenever possible.

3.2. Assumptions, dependencies, and constraints

Initial assumptions include the platform of the application being limited to the Android mobile platform, and will depend on camera and speaker hardware. Additionally, available libraries will be limited to the development platform and version chosen. A secondary user who serves as the primary user's aid is assumed to also interact with the application, therefore proper GUI elements should be present. The project will also depend on GitHub for source control and collaboration, therefore the availability of the GitHub service will be crucial.

When beginning to develop the application, all other deliverable activities must be completed prior. The programming of the application highly depends on specification activities and WRS. This is to ensure the final prototype meets the requirements.

Since the majority of the project's development will not be done in person, frequent online meetings will be necessary. These are planned to be conducted weekly during development, and consist of discussions on progress and future developments. During development of the application, physical testing may be necessary and will either be done individually or in person considering the online nature of the project. Meetings will also be planned in regards to team members availability.

3.3. Risk management

| No. | Risk | Type | Likelihood | Description |
|-----|---|------------|------------|---|
| 1 | Failure to meet deadlines for deliverables | Managerial | Unlikely | Deliverable is not ready by the deadline |
| 2 | Team member unavailability | Managerial | Unlikely | Team members are not available to complete project or deliverable on time |
| 3 | Resource / tool unavailability | Technical | Unlikely | Project dependent resources or tools become unavailable temporarily or permanently |
| 4 | Accidental loss of progress | Technical | Unlikely | A portion of the project is lost due to source control errors, crashes, hardware destruction, ect |
| 5 | Requirement change | Technical | Unlikely | Project requirements change during development |
| 6 | Deliverable holding up project development | Managerial | Unlikely | Certain deliverable requires excessive time commitment, delaying proceeding deliverables |
| 7 | Deliverable does not meet client's requirements | Managerial | Unlikely | Project deliverable does not satisfy client and must be redeveloped |
| 8 | Poor code quality | Technical | Unlikely | Code does not meet standards, is inefficient, or contains bugs |

Table 1: Potential risks and their descriptions

3.4 Monitoring and controlling mechanisms

| No. | Risk | Monitoring and Controlling |
|-----|---|--|
| 1 | Failure to meet deadlines for deliverables | <ul style="list-style-type: none"> • Utilize project planning platforms. Ex. Trello, Asana • Set up milestones and report outs for each deliverable • Use frequent communication through discord |
| 2 | Team member unavailability | <ul style="list-style-type: none"> • Reevaluate member availability during each meeting • Reschedule members when necessary • Reschedule deadlines to match member availability |
| 3 | Resource / tool unavailability | <ul style="list-style-type: none"> • Reach out to resource IT or help services • Determine alternatives or backups for resources or tools |
| 4 | Accidental loss of progress | <ul style="list-style-type: none"> • Maintain project backups outside of source control • Ensure hardware stability |
| 5 | Requirement change | <ul style="list-style-type: none"> • Set up team meetings to assess changes and current progress usability • Reschedule deadlines and reassign tasks as needed |
| 6 | Deliverable holding up project development | <ul style="list-style-type: none"> • Reassess deliverable priorities and dependencies • Assign additional members to deliverable • Split deliverable into smaller ones |
| 7 | Deliverable does not meet client's requirements | <ul style="list-style-type: none"> • Obtain feedback on changes • Evaluate deliverable and refactor to meet feedback |
| 8 | Poor code quality | <ul style="list-style-type: none"> • Maintain a common code standard • Ensure all team members agree and abide with coding standard • Utilize testing at every stage of development • Conduct code reviews |

Table 2: Controlling and monitoring methods for risks described in Table 1

4. Technical Process Plans

4.1. Methods, tools, and techniques

Our plan for the application would be to create an app for Android devices that uses a machine learning algorithm, in combination with and several sensors on the device, such as the camera, GPS, gyro sensors, to provide information on the surroundings and make decisions about the data, informing the user of potential obstacles at well as helping them navigate. For the interface, we plan to use speech-to-text so that the user can issue voice commands to the application, and text-to-speech, as well as other auditory cues, to relay information to the user. Because the interface is largely through speech, and not visual information, it should be easy for the application to run on a wide range of devices. We will be using GPS tracking, combined with a digital map and possibly visual data to give the user directions for navigating inside of a building.

The app will be programmed in Java using Android Studio to provide the best compatibility with external software, such as the machine learning algorithm, which could also make it easy to expand on the features of the app later on. To further help with this, as well as our own development, we will make the app as modular and compartmentalized as reasonably possible, so that various portions of it can be easily modified or replaced, and potentially disabled based on the available features of the host device. This will allow us to implement an iterative design process so that we can quickly prototype, and easily refine our project over time. Our machine learning model will be externally trained on a desktop computer, and the resulting network will be ported over to run on mobile devices.

As for the development process, we will be using Github/Gitlab for source control for all of our code and documentation, and making heavy use of branches so that multiple people can work on different features at the same time, independent of others. After a good amount of progress has been made, we will put it through a testing process, running the application on several different devices with different specs, sensors, aspect ratios, etc., as well as testing all of the features individually and looking for bugs. After this, we will then give the devices to members of the blind community to have them test it and give feedback, or if that's not possible, potentially doing something like blindfolding team members and having them try it. After this testing process, all of the code that has been completed and tested can be committed to the main branch and become part of the application.

For keeping track of progress on the application, we will be using Github Projects or similar software to keep a list of items and track their projects, as well as a separate list of major milestones to reach, to give a better general overview of the project. This will allow us to easily stay on the same page as a team, as well as seeing what other team members are working on, and being able to quickly and efficiently update the list of tasks and change the status of items. This will be extremely important, in the middle to late stages of the project where we will have many different goals to track and will need to make certain that we have a very clear understanding of what needs to be done.

We will be using a variation on the Agile design philosophy, and iterating on our project many times through its development to help us adapt to changes and to improve it over time. This is especially important for this project because it makes use of machine learning, which

commonly requires many failed attempts, and subsequent changes, before producing a useful result, and even more to improve upon that result and make it more accurate. It will also help us adapt to the needs of the blind community, and let us change our software based on their feedback to make it easier to use and more useful to them. After the application is released, user reviews and feedback will be a very important part of the continued development over the lifespan of the app.

4.2. Software documentation

| Document | Template or Standard | Created By | Reviewed By | Target Date | Distribution |
|-------------------|-----------------------------|-------------------|--------------------|--------------------|------------------------|
| User Guide | N/A | Caden | Sam | Oct 15 | Text and Speech in App |
| Help Guide | N/A | Caden | Sam | Oct 15 | Text and Speech in App |
| Command List/Info | N/A | Caden | Sam | Oct 15 | Text and Speech in App |