

Final Project Plan

Caleb Valdez-Copeland

Keegan Milsten

Luke Thorvilson

Darrel Nitereka

Gerald Hoff

Version	Release	Responsible Party	Major Changes	Date
---------	---------	-------------------	---------------	------

0.1	—	September 17, 2023
-----	---	--------------------

1.0	—	December 10, 2023
-----	---	-------------------

Software Project Management Plan (SPMP)

0.1 Initial Document Release for Comment

1.0 Final Document Release for Final Plan

Table of Contents

Table of Contents	2
1. Introduction	3
1.1 Project Overview	3
1.2 Project Deliverables	4
1.3 Evolution of the SPMP	4
1.4 Reference Materials	5
1.5 Definitions and Acronyms	5
2. Project Organization	6
2.1 Process Model	6
2.2 Organizational Structure	6
2.3 Organizational Interfaces	7
2.4 Project Responsibilities	8
3. Managerial Process	9
3.1 Management Objectives and Priorities	9
3.2 Assumptions, Dependencies, and Constraints	9
3.3 Risk Management	9
3.4 Monitoring and Controlling Mechanisms	10
4. Technical Process	11
4.1 Methods, Tools, and Techniques	11
4.1.1 Computing Systems	11
4.1.2 Development Method	11
4.1.3 Standards and Policies	11
4.1.4 Team Structures	11
4.1.5 Languages	11
4.2 Software Documentation	12
4.2.1 Software Requirements Specification (SRS)	12
4.2.1.1 Location Acceptance	12
4.2.1.2 Real-Time Route Planning and Decision Making	12
4.2.1.3 Audible Directions	12
4.2.1.4 Detect Path Obstructions	12
4.2.1.5 Detect Falls, Rapid Movement, and safety hazards	12
4.2.1.6 System Prediction of Habits	12
4.2.2 Software Design Description (SDD)	13

4.2.3 Software Test Plan	13
4.3 User Documentation	13
4.4 Project Support Functions	13
4.4.1 Configuration Management	13
4.4.2 Quality Assurance	13
4.4.3 Verification and Validation Plan	13
5. Work Packages, Schedule, and Budget	14
5.1 Work Packages	14
5.2 Dependencies	14
5.3 Resource Requirements	15
5.4 Budget and Resource Allocation	15
5.5 Schedule	18

1. Introduction

This section of the SPMP provides an overview of the project.

1.1 Project Overview

The project's primary objective is to develop Theia- a smartphone application tailored to aid blind individuals in safely navigating indoor spaces, including but not limited to: offices, schools, and hospitals. Theia is built for the blind — with their needs in mind, key features encompass voice-controlled navigation, real-time obstacle detection, and customizable settings for user preferences. The primary stakeholders are visually impaired users, with caretakers and family members serving as secondary users who can configure the app and offer assistance when required. Additionally, building managers or employers can update layouts in real time to account for unexpected changes.

Theia app employs smartphone sensors and artificial intelligence to pinpoint the user's location, identify obstacles, and recommend the most efficient routes within the building. Notable features include voice command input and audio guidance, mapping algorithms for route planning, obstacle detection via sensor integration, and safety measures like emergency contacts and monitoring. Users can also personalize settings, adjusting volume, speech rate, and other preferences.

The primary focus of this project is to address the needs of blind individuals seeking independent navigation within unfamiliar indoor environments. Secondary stakeholders, such as caretakers and family members, can configure Theia and provide remote assistance as necessary. Project deliverables encompass a requirements model, specifications document, and the creation of a prototype app available on both iOS and Android platforms.

1.2 Project Deliverables

Deliverable	Delivery Date	Created By	Delivery Method	Quantity	Comments
Preliminary Plan	09/17/2023		Canvas	1	Completed
Theia App Documentation				1	
Project Phase I: Final Submission	10/15/2023		Canvas Gitlab	1	Completed
Project Phase II: Final Submission	12/10/2023		Canvas, Github	1	Work in Progress
Theia Prototype	12/7/2023		Canvas, Github	1	Completed
Theia App	01/10/2024		Installable app for both iOS and Android	1	Work in Progress

1.3 Evolution of the SPMP

The SPMP will undergo incremental development throughout the project's duration, managed by the team. Sections will be added as necessary to comprehensively define project parameters, scope, deliverables, and processes. The initial draft will be shared with team members for review and input. Planned revisions will occur at project phase milestones and will be disseminated for approval. Unscheduled updates may emerge in response to changes in project scope or requirements. All revisions will be systematically tracked and managed through version control.

For changes to the SPMP, an approval process will be employed. Proposed modifications will be evaluated by the project manager and relevant team members to assess their impact on project scope, schedule, and budget. Upon approval, these changes will be integrated into a new version of the SPMP. Major changes may necessitate approval from the project sponsor.

This change management process will ensure that all project team members remain well-informed

of pertinent SPMP alterations, which may pertain to processes, deliverables, schedules, risks, or other factors that could affect their work. Scheduled and unscheduled updates will be managed with diligence to maintain project alignment with evolving requirements and conditions.

1.4 Reference Materials

GeeksforGeeks. (2021, January 5). How to convert speech into text using JavaScript.

<https://www.geeksforgeeks.org/how-to-convert-speech-into-text-using-javascript/>

Chinda, G. (2021, June 8). How To Build a Text-to-Speech App with Web Speech API.

DigitalOcean.

<https://www.digitalocean.com/community/tutorials/how-to-build-a-text-to-speech-app-with-web-speech-api>

Codepo. (n.d.). GitHub - codepo8/github-page-pwa: A template to offer GitHub pages as a PWA with offline functionality. GitHub. <https://github.com/codepo8/github-page-pwa>

1.5 Definitions and Acronyms

SPMP: Project Management Plan

Agile: A software process in which a project is broken up into bite-sized portions and tasks are grouped together to be completed in relatively short time periods

AI: Artificial Intelligence

iOS: Apple's mobile operating system

Android: Google's mobile operating system

GPS: Global Positioning System

Stakeholder: Individuals or entities with an interest or involvement in the project

Obstacle Detection: The capability to identify physical obstacles in the user's path

Collision Avoidance: The ability to guide the user around obstacles to prevent collisions

Navigation Instructions: Directions and guidance provided to the user for safe navigation

Prototype: A preliminary version of the Theia app used for testing and evaluation

Playwright: Testing framework that provides end-to-end testing for modern web apps.

Emergency Contacts: Predefined contacts for immediate assistance in case of emergencies

User Preferences: Customizable settings tailored to individual user needs

PWA: Stands for Progressive Web App, a term which refers to applications available via a web browser in the form of a website, built using web technologies but available with many of the features of a usual application.

2. Project Organization

2.1 Process Model

Being that Theia will be a smartphone app, the best approach will be the agile development life cycle model. The agile development model is an adaptive and iterative approach that allows for flexibility, reduced risks, and high product quality which are beneficial attributes for the specifications asked in a project like Theia. The roles will include the Project Manager, Technical Team Leaders, Developers, and Quality assurance. The project will be broken down into key phases with each phase holding activities. The **Initiation phase** will hold the project charter creation which will lead to the Preliminary plan deliverable. The **Product development Phase** will hold the User Story Creation and Sprint planning which contributes to the Project phase: Final Submission deliverable, and the coding and development which continues to the Theia App deliverable. The **Product Release Phase** will hold Beta testing and accessibility testing that contribute to the Theia app, as well as documentation and training for the Theia app documentation. The **Project Termination Phase** will hold the Project evaluation which may consist of reviewing the deliverables for the Preliminary Plan and Project Phase 1: Final Submission progress. It will also hold documentation that will involve finalizing and documenting the progress made and a potential Handover process that will be included in Theia App Documentation. The celebration will then mark the successful completion of the project and its deliverables.

2.2 Organizational Structure

The internal management structure will be in the form of:

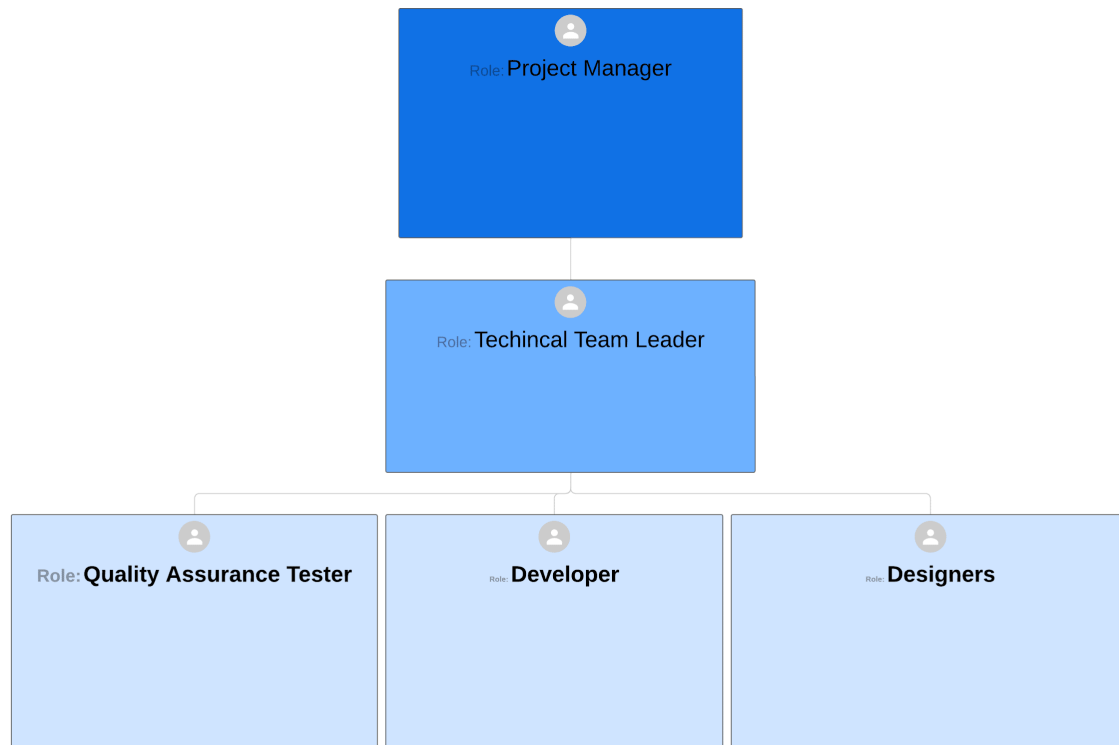
Project Manager: Responsible for coordination of the project, communication with stakeholders, and overall team leadership

Technical Team Leader: Technical aspects of project and coordination of developmental efforts led by this role.

Developers: Coding and app feature implementation

Quality Assurance Tester: Ensures the quality of the app through testing

Designers: Responsible for user experience design and user interface



2.3 Organizational Interfaces

Organization Liaison	Contact Information
Customer	[Customer Name]
	Phone: [Customer Phone Number]
	Email: [Customer Email Address]
Subcontractor	[Subcontractor Name, if applicable]
Software Quality Assurance	[QA Liaison Name]
	Phone: [QA Liaison Phone Number]
	Email: [QA Liaison Email Address]
Software Configuration Management	[Configuration Management Liaison Name]

	Phone: [Configuration Management Phone Number]
	Email: [Configuration Management Email Address]
[Add any other relevant interfaces]	[Contact Information for Other Entities]

2.4 Project Responsibilities

Role	Description	Person
Project Manager	Responsible for coordination of the project, communication with stakeholders, and overall team leadership	[Name]
Technical Team Leader(s)	Technical aspects of the project and coordination of developmental efforts led by this role.	[Name]
Developers	Coding and app feature implementation	[Name]
Quality Assurance Tester	Ensures the quality of the app through testing	[Name]
Designers	Responsible for user experience design and user interface	[Name]

3. Managerial Process

3.1 Management Objectives and Priorities

The management philosophy for this project is to divide the project into clear achievable deliverables and to create thoroughly tested code that ensures the safety of the user. Tasks, including testing, were divided among team members with discord being used to communicate as a team. Communication between the team and the client will be facilitated through email. The completion of the project as a whole is dependent on the combined contributions of individual team members. Therefore there is an understanding of mutual accountability in completing these tasks. The exact specifications of deliverables will be flexible, as we will likely gain new insights to the functional requirements that will best satisfy the high importance non-functional requirements.

3.2 Assumptions, Dependencies, and Constraints

The assumptions for this project are that it will be completed over the course of several months by our team. And that it will be an application that can safely help visually impaired users navigate from one location to another. Functionality is therefore the most important constraint as it potentially impacts the physical safety of the users. Scheduling is the second most important constraint as we have a limited amount of time to complete the project.

The software itself will be dependent on the application platforms we use. Essentially depending on whether we want the app to be supported on Android and/or IOS

3.3 Risk Management

No.	Risk	Type	Likelihood	Description
1	Failure to fully consider use cases	Managerial	Likely High potential impact	The use cases for the application are specific to visually impaired users. Given this, we might not understand all the needs of the users.
2	Failure to meet deadlines for deliverables	Managerial	Unlikely High potential impact	The unavailability of team members, or other situations that lead to deadlines not being met.

3	Requirements change	Technical	Unlikely High potential impact	Requirements are unlikely to change, unless suggested by the mentor, or unless new revelations about use cases emerge.
4	Accidental loss of valuable information	Technical	Unlikely High potential impact	Loss of information due to improper git usage, or file storage.

3.4 Monitoring and Controlling Mechanisms

No.	Risk	Monitoring and Controlling
1	Failure to fully consider use cases	<ul style="list-style-type: none"> • Understanding the requirement specification • Considering the full scope of user use cases
2	Failure to meet deadlines for deliverables	<ul style="list-style-type: none"> • Set up milestones to meet deadlines on time. • If milestones are not met, meet with mentor
3	Requirements change	<ul style="list-style-type: none"> • Understanding the requirement specification to prevent changes to use cases. • Assess the amount of work to be changed. • Assign more resources to meet additional requirements
4	Accidental loss of valuable information	<ul style="list-style-type: none"> • Maintain back-ups for every piece of project work. • Use version control and shared documents

		to prevent data loss.
--	--	-----------------------

4. Technical Process

4.1 Methods, Tools, and Techniques

4.1.1 Computing Systems

Since the navigation app needs to be handheld and lightweight so the user can navigate through buildings, the computing system would be a mobile device. This being devices that are IOS and Android. The app will function on both operating systems using the devices built in GPS and other related tools.

4.1.2 Development Method

Agile Development will be used in the processes going forward. Team members will each come together, receive tasks and complete these tasks for the current sprint. The members will come together and provide the progress they've made by the end of the sprint. Then the process will continue with new sprint tasks.

4.1.3 Standards and Policies

The standards of Web Content Accessibility Guidelines(WCAG) will be followed to ensure that disabled users, such as blind users are able to use the app. These guidelines are a good standard to follow so that our app is usable to its fullest potential.

Data security is of most importance to our team. User data will be very strict considering the app uses locational information. The app is also expected to provide quality assurance, this being that it remains accurate for the user, given all circumstances.

<https://www.w3.org/WAI/standards-guidelines/wcag/wcag3-intro/>

4.1.4 Team Structures

The team is made up of mobile app, backend, and UI developers. This being the major makeup for the main app. There are also accessibility developers; these developers take into account the ideals that must be followed when developing a website for a blind user. There are testers who will test all frontend/backend functionalities.

4.1.5 Languages

The languages the app will consist of are CSS, HTML, Javascript, and SQL for database design. These will use the Playwright framework for testing which supports all modern

rendering engines such as the one Chrome utilizes. Playwright is a testing framework that provides end-to-end testing for modern web apps, such as Theia. Unit tests will also be used to test backend and Playwright will be used for frontend testing.

4.2 Software Documentation

4.2.1 Software Requirements Specification (SRS)

4.2.1.1 Location Acceptance

The system must be able to store user data such as destinations and mapping data for the navigational system. The system should be able to suggest destinations too for blind users so they don't have to search the system's data for old destinations. The suggestions will come based on locations when they were used.

4.2.1.2 Real-Time Route Planning and Decision Making

The system should be able to decide the fastest routes based on past data. This being time spent on a certain route so it can determine which way may be the more efficient route.

4.2.1.3 Audible Directions

The system should announce when a turn is coming and how long it may take to get to the turn. The system should use GPS coordinates to determine how far the user may have to go to make these decisions. Accuracy of directions are stored for future uses. The system will also tell the user when to stop exactly and turn using prior knowledge of the location and layout of the building.

4.2.1.4 Detect Path Obstructions

The system uses sensors on the phone to detect objects obstructing the path. The audible system then will tell the user to halt and reroute them around the object safely. If unable the user will be rerouted down a different path to avoid the obstruction.

4.2.1.5 Detect Falls, Rapid Movement, and safety hazards

The system will be able to sense motion in response to a fall or collision. The system will first prompt for an audible from the user, if the user is unresponsive, the system will place an emergency call to paramedics and authorities to send over help. The system also will report to any close contacts that there has been an emergency.

4.2.1.6 System Prediction of Habits

The system should be able to use historical data from the user based off of schedule, time, and previous records to prompt suggestions. There will also be GPS notable areas which are triggered when entered.

4.2.2 Software Design Description (SDD)

The database used will be a Postgres SQL database which will use user information as well as store locational and graphical information that is needed for the app to function properly. For the UI, bootstrap will be a major contributor to the app's user friendly feel. It will also have voice command using Google Web Speech API, for users that do not have access to the UI.

4.2.3 Software Test Plan

Unit testing will be conducted through Test Driven Development. Each feature such as the ones specified in SRS above will be created based off of black-box tests. All functionality will be tested with Playwright. Playwright supports PWA unit tests and will be fully utilized throughout the whole application for proper and full functionality.

4.3 User Documentation

User documentation will be created for blind and non-blind users. Thus braille will be used in correspondence to descriptions and images. Images will be described in braille so that users are still able to develop proper understanding of the software functionalities. Most if not all the documentations developed will be translated to braille. Proper information will be gathered from reliable sources for validity and correctness.

<https://www.brailletranslator.org/>

4.4 Project Support Functions

4.4.1 Configuration Management

The app will be configured and managed through GitHub. Every release will be pushed to a main branch for download. Each change the version will be described in a ReadMe.md file that describes changes made after each software update. Each version will be tracked through GitHub's commit history and provides great detail for each change pushed to the repository containing the full-stack application.

4.4.2 Quality Assurance

Quality will be maintained through every version that is released. This being that all functionality will be tested to make sure that it provides full user-friendly, as well as blind-friendly features that work properly. All design strategies that are necessary will be followed for each feature as well as code based reviews to make sure that code follows all standards and maintains a clean and proper structure in the apps frontend and backend.

4.4.3 Verification and Validation Plan

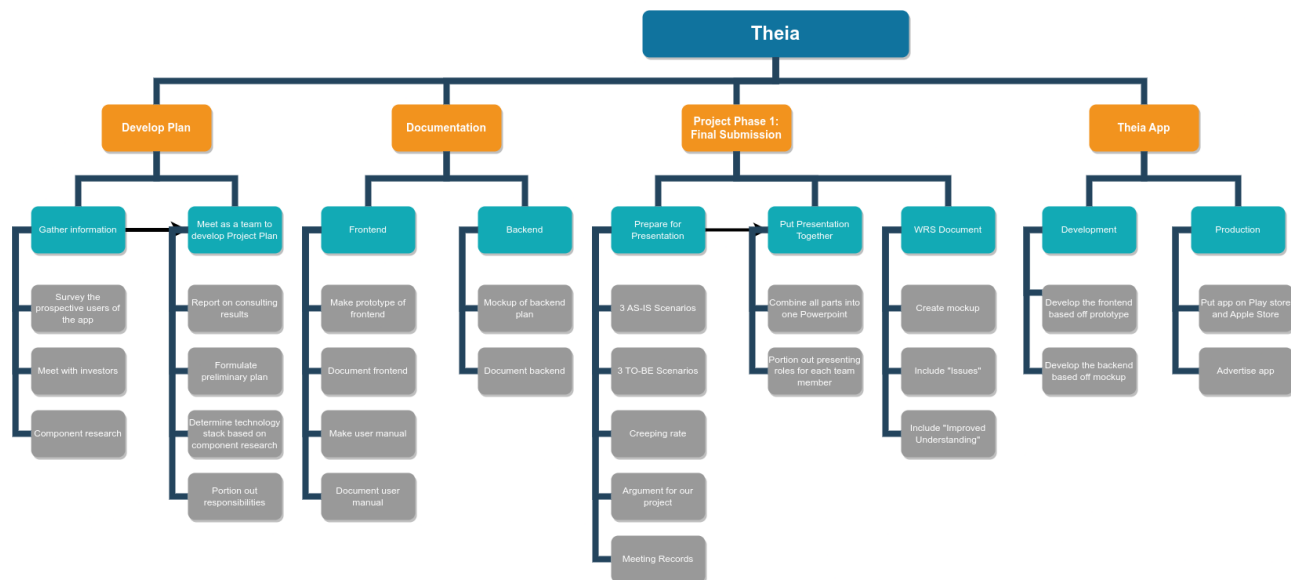
For verification and validation, all features will be unit, integration, and system tested. This

is to make sure that no failures will occur that can cause unwanted incidents. The system will have a failsafe system in case of a failure. This being that the app will restart and reload where it left off in-case of any glitches or crashes. In the case of a crash, an incident report will be sent to the team so the fault can be identified and squashed for future users and functionality.

5. Work Packages, Schedule, and Budget

5.1 Work Packages

This project is scheduled to be completed by January 10th, 2024, satisfying the project agreement. Depicted below is a work breakdown structure detailing what must be done to reach our goal by January 10th, 2024. Currently our team has accomplished all but the last stage: the final orange box on the right.



5.2 Dependencies

Project development takes place from left to right of the orange boxes. First the plan is developed, and finally the app is delivered. As for the layer of blue boxes beneath them, arrows specify to move on to the box it is pointing to only after the source of the arrow is complete. If no arrows exist, work can be done in tandem. One good example of this is documenting the backend and frontend. The documentation of each can be done simultaneously, while under the orange box of "Develop Plan", all steps of "Gathering Information" must be done prior to the development of a project plan.

5.3 Resource Requirements

Estimate of Total Resources		Time Estimate
Project Manager	1 person	All 5 months
Technical Team Leaders	5 people	All 5 months
Developers	20 people	All 5 months
Quality Assurance Tester	1 person	All 5 months
Designers	5 designers	All 5 months
WBS software		1 month
25 copies of an IDE license + software (for developers and technical team leaders)		All 5 months
1 computer for each person (32 computers)		All 5 months
1 office unit (cubicle) for each person (32 office spaces)		All 5 months
Gas Money + airline tickets (for meeting with investors)		First week

5.4 Budget and Resource Allocation

Budget and Resource Allocation (by personnel)	
Project Manager	\$60,000
Technical Team Leaders	\$50,000/each
Developers	\$40,000/each
Quality Assurance Tester	\$30,000
Designers	\$30,000/each

Budget and Resource Allocation (by items)	
WBS Software	\$500
25 copies of an IDE license + software	\$500/each
Computers	\$1,000/each
Office Unit	#1000/each
Gas Money (for necessary travel)	\$500
Airline tickets (for necessary travel to meet investors)	\$1000/each

Budget and Resource Allocation (by task)		
Gather Information		\$300
↳ <i>Survey the prospective users of the app</i>	<i>1 designer</i>	<i>\$100</i>
↳ <i>Meet with investors</i>	<i>Project Manager</i>	<i>\$100</i>
↳ <i>Component research</i>	<i>Technical Team Leaders + Developers</i>	<i>\$100</i>
Meet as a team to develop Project Plan		\$0
↳ <i>Report on consulting results</i>	<i>Project Manager</i>	<i>\$0</i>
↳ <i>Formulate preliminary plan</i>	<i>Project Manager + Technical Team Leaders</i>	<i>\$0</i>
↳ <i>Determine technology stack based on component research</i>	<i>Project Manager</i>	<i>\$0</i>
↳ <i>Portion out responsibilities</i>	<i>Project Manager + Technical Team Leaders</i>	<i>\$0</i>
Frontend		\$1300
↳ <i>Make prototype of frontend</i>	<i>Designers + Developers</i>	<i>\$1000</i>

↳ <i>Document frontend</i>	<i>5 Developers</i>	<i>\$100</i>
↳ <i>Make user manual</i>	<i>5 Designers</i>	<i>\$100</i>
↳ <i>Document user manual</i>	<i>5 Developers</i>	<i>\$100</i>
Backend		\$600
↳ <i>Mockup of backend plan</i>	<i>All Designers</i>	<i>\$500</i>
↳ <i>Document backend</i>	<i>5 Designers</i>	<i>\$100</i>
Prepare for Presentation		\$0
↳ <i>3 AS-IS Scenarios</i>	<i>Project Manager + Technical Team Leaders</i>	<i>\$0</i>
↳ <i>3 TO-BE Scenarios</i>	<i>Project Manager + Technical Team Leaders</i>	<i>\$0</i>
↳ <i>Creeping rate</i>	<i>Project Manager</i>	<i>\$0</i>
↳ <i>Argument for our project</i>	<i>Project Manager</i>	<i>\$0</i>
↳ <i>Meeting Records</i>	<i>Technical Team Leaders</i>	<i>\$0</i>
WRS Document		\$100
↳ <i>Create mockup</i>	<i>All Developers</i>	<i>\$100</i>
↳ <i>Include "Issues"</i>	<i>1 Developer</i>	<i>\$0</i>
↳ <i>Include "Improved Understanding"</i>	<i>1 Developer</i>	<i>\$0</i>
Theia app development		\$2000
↳ <i>Develop out the frontend based off prototype</i>	<i>Half the developers + their team leaders, as well as all the designers + quality assurance testers</i>	<i>\$1000</i>
↳ <i>Develop the backend based off mockup</i>	<i>Half the developers + their team leaders</i>	<i>\$1000</i>
App publication/production		\$1200
↳ <i>Put app on Play store and Apple Store</i>	<i>1 Technical Team Leader</i>	<i>\$200</i>
↳ <i>Advertise app</i>	<i>Project Manager +</i>	<i>\$1000</i>

	<i>Designers</i>	
--	------------------	--

5.5 Schedule

Schedule — Totalling 158 days (~5 months)	
Gather Information	15 days
↳ <i>Survey the prospective users of the app</i>	<i>9/01 – 9/08</i>
↳ <i>Meet with investors</i>	<i>9/01 – 9/08</i>
↳ <i>Component research</i>	<i>9/06 – 9/15</i>
Meet as a team to develop Project Plan	3 days
↳ <i>Report on consulting results</i>	<i>9/18</i>
↳ <i>Formulate preliminary plan</i>	<i>9/18 – 9/20</i>
↳ <i>Determine technology stack based on component research</i>	<i>9/20</i>
↳ <i>Portion out responsibilities</i>	<i>9/22</i>
Frontend	21 days
↳ <i>Make prototype of frontend</i>	<i>9/21 – 10/11</i>
↳ <i>Document frontend</i>	<i>10/10 – 10/12</i>
↳ <i>Make user manual</i>	<i>9/21 – 10/11</i>
↳ <i>Document user manual</i>	<i>10/10 – 10/12</i>
Backend	21 days
↳ <i>Mockup of backend plan</i>	<i>9/21 – 10/11</i>
↳ <i>Document backend</i>	<i>10/10 – 10/12</i>
Prepare for Presentation	1 day
↳ <i>3 AS-IS Scenarios</i>	<i>10/13 – 10/13</i>
↳ <i>3 TO-BE Scenarios</i>	<i>10/13 – 10/13</i>
↳ <i>Creeping rate</i>	<i>10/13 – 10/13</i>

↳ <i>Argument for our project</i>	<i>10/13 – 10/13</i>
↳ <i>Meeting Records</i>	<i>10/13 – 10/13</i>
WRS Document	2 days
↳ <i>Create mockup</i>	<i>10/13 – 10/14</i>
↳ <i>Include "Issues"</i>	<i>10/13 – 10/14</i>
↳ <i>Include "Improved Understanding"</i>	<i>10/13 – 10/14</i>
Theia app development	46 days
↳ <i>Develop out the frontend based off prototype</i>	<i>10/23 – 12/22</i>
↳ <i>Develop the backend based off mockup</i>	<i>11/23 – 12/22</i>
App publication/production	14 days
↳ <i>Put app on Play store and Apple Store</i>	<i>12/27 – 01/05</i>
↳ <i>Advertise app</i>	<i>12/27 – 01/10</i>