Theia Team

Revised Project Management Plan

Phase I

Chandler Guthrie, Erik Winiski, Julian Hutchins, Logan Cribbs, Matthew Hill, Yaru Gao

10/12/25

Revision History

Responsible Party	Date	Reason For Changes	Version
Chandler, Erik, Julian, Logan, Matthew, Yaru	09/14/25	Initial draft	0.0.1d-14SEP25
Chandler	09/24/25	- Project Organization Section Revision - Responsibilities Revised	0.0.2d-24SEP25
Matthew	10/6/25	Project Overview Revision	0.0.3d-6OCT25
Matthew	10/12/25	Final Phase I submission editing	0.0.4d-12OCT25

TABLE OF CONTENTS

I. Introduction

I.i. Project Overview

The goal of this course project is to illustrate the development process by creating a hypothetical mobile application to allow blind and visually impaired users to safely navigate indoor public areas. Through the theoretical progress of this system, the team will demonstrate their abilities to create a well-structured project management plan using techniques to professionally elicit requirements from clients and stakeholders. Our focus is to develop skills related to requirements elicitation, specification, analysis, and validation to gain an understanding of the requirements engineering process.

As this project is meant for academic purposes, there will be no budget provided; however, our team will instead make use of student-provided resources, licenses, and open-source tools. This also means that any clients and stakeholders involved in this project will be roles assumed by our team.

Major work activities include the following:

- Development of AS-IS/TO-BE scenarios for Project Phase I presentation.
- Create mockup for development plan for Project Phase II.
- Creation of mockup mobile application and user manual.
- WRS document for Project Phase I, and updated WRS document for Project Phase II after development of a mockup application.

Major milestones include the following:

- Project Phase I: Preliminary Project Plan for the initial requirements summary and project planning.
- Project Phase I: Final Submission/Presentation with updated documentation and presentation of initial project plan of action.
- Project Phase II: Final Submission/Presentation with final mockup and short demonstration of working code.

Required resources include the following:

- Shared GitHub repository for collaboration and version control.
- Discord for file sharing, discussing deadlines, planning meetings, etc.
- Software/UI development tools for creation of mobile application prototype.

The official statement of product requirements for this project is provided here.

I.ii. Project Deliverables

Deliverable	Delivery Date	Delivery Location	Quantity
Project Phase I: Preliminary	9/14/25	Canvas and GitHub	Preliminary plan

Project Plan			
Project Phase I: Final Submission/Presentation	10/12/25	Canvas and GitHub	Phase I documents, meeting records, PowerPoint slides
Project Phase I: Peer Review	10/12/25	Canvas	Form
Prototype and User Manual	11/9/25	GitHub	Prototype, user manual
Refined Prototype w/ Expanded Features and Updated Documentation	11/23/25	GitHub	Refined prototype, updated documentation
Final Integrated Prototype, Project Phase II Submission	12/7/25	Canvas and GitHub	Final prototype, Phase II documents
Final Project Submission/Presentation	12/7/25	Canvas and GitHub	Final project documents

I.iii. Evolution of the PPMP

This PPMP will be drafted at the start of Project Phase I and collaboratively maintained via Google Docs. Its contents consist of the team's initial plans and agreements for this project. Scheduled updates will occur after team review and approval is given, while unscheduled updates will be proposed during Discord communications (messaging or voice channel) and adopted after team review and approval.

The completion of this project will happen incrementally and will be shared with all members of the team. Project updates are to be communicated during weekly team meetings and proper documentation in GitHub.

Updates and changes will be traced via the team's recorded version control and GitHub uploads. A finalized version of this document will later be added to the team's GitHub repository and submitted to Canvas for the end of Phase I.

I.iiii. Reference Material

Karl Wiegers and Seilvel, "Good practices for requirements engineering," in Software Requirements, 3rd ed. Redmond, WA, USA: Microsoft Press, 2013, ch. 3, pp. 44–56.

Karl Wiegers and Seilvel, "Requirements Management Practices," in Software Requirements, 3rd ed. Redmond, WA, USA: Microsoft Press, 2013, ch. 27, pp. 468-469.

Karl Wiegers and Seilvel, "Glossary," in Software Requirements, 3rd ed. Redmond, WA, USA: Microsoft Press, 2013, pp. 597, 600, 602-603.

Frequency Response and Bias, NERC Reliability Standard BAL-003-0.1b, May 2009. [Online]. Available: http://www.nerc.com/files/BAL-003-0_1b.pd

I.v. Definitions and Acronyms

- ***Agile;*** A term used for software development methods characterized by continuous collaboration between developers and customers, limited documentation of requirements in the form of user stories and corresponding acceptance tests, and rapid and frequent delivery of small increments of useful functionality. Agile developments include Extreme Programming, Scrum, Feature-Driven Development, Lean Software Development, and Kanban.
- ***Backlog(Kanban);*** Prioritized list of work remaining for the project. It can contain user stories, business processes, change requests, infrastructure development, and defect stories. Work items from the backlog are allocated to upcoming iterations based on their priority.
- ***GPS;*** Global Positioning System.
- ***PPMP;*** Preliminary Project Management Plan.
- ***Sprint;*** An uninterrupted development period, typically one to four weeks in duration, during which a development team implements a defined set of functionality selected from the product backlog or baselined requirements for the product.
- ***SRS;*** A collection of the functional and non-functional requirements for a software product. (should be written for understanding by both developers and stakeholders).
- **Story Point;** Total is proportional to the amount of effort the team must expend to implement the requirements.
- ***User Story;*** A format to capture user requirements on agile projects in the form of one or two sentences that articulate a user need or describe a unit of desired functionality, as well as stating the benefit of the functionality.
- ***Use Case;*** A description of a set of logically related possible interactions between an actor and a system that results in an outcome that provides value to the actor. It can encompass multiple scenarios.

II. Project Organization

II.i. Process Model

To complete the documentation and partial parts of the programming side, the life cycle we have chosen is agile as it specializes in flexibility and collaboration within our team, especially as we must act as different stakeholders along with the business analyst. Everything will be done in sprints as an incremental and iterative process for elicitation, analysis, specification, validation,

requirements management, and project management. However, with the scope of this project, we will only be going through one sprint. To accomplish such a process from start to end, our sprint will go through a rotation of planning, design, development, deployment, and review. Some of these steps may be revisited during other parts of the sprint.

II.ii. Organizational Structure

Our team responsible for all steps of the agile life cycle is called the Theia Team. The members are as follows: Chandler Guthrie, Erik Winiski, Julian Hutchins, Logan Cribbs, Matthew Hill, and Yaru Gao. Matthew Hill will be the liaison of Theia Team. All members will be involved in every part of the project, such as the documentation, which includes the WRS, this document, and the AS-IS and TO-BE slides, as well as the design and development of the application. All reviews will be taken by our team, Theia Team, with ourselves acting as the client/stakeholders of the application.

II.iii. Organizational Boundaries and Interfaces

All parties of the Theia Team and our mentor will be communicating through the chosen liaison. The main form of contact between them will be by email, with the possibility of meetings on Zoom for clarification. Each member of Theia Team will be using platforms such as Discord and email for communication and constant collaboration, and other tools which are discussed in depth in *Methods, Tools, and Techniques*.

II.iiii. Project Responsibilities

Due to the nature of this project and the scope, many of the responsibilities may be shared; however, it is completely possible to begin with claims of ownership of said responsibilities. These people will be responsible for the completion of the task, including giving out possible responsibilities.

Role	Description	Person
Liaison	Communicates with all parties outside Theia, keeping everyone in the know.	Matthew Hill
Project Manger	In charge of communicating responsibilities and keeping track of progress.	Chandler Guthrie

We, the Theia Team, have decided to break down each responsibility in accordance with the agile methodology and good engineering practices for a requirements development process framework.

Responsibility	Reference To Course Work	Owner(s)	Number of Responsible Requirement
Introduction to the project, including the scope, the objectives, and	WRS [1] ,1, 2, 3, 5	Chandler G.	1

success criteria, and an overview.			
Definitions, Acronyms, and Abbreviations Adding references to the Writing Requirements Specification (WRS)	WRS[1] 4, AND WRS [7]	ALL	INF
(Provided the use cases, if any) Use the business requirements to list the preliminary domain of the project and list all the preliminary functional and non-functional requirements.	WRS [2] 1, 2, 3	Julian H.	1
Using the preliminary definitions, find issues and point them out, then create some options to solve the issue and choose the best option for each preliminary domain, functional, and non-functional requirements	WRS [3] 1, 2, 3	Julian H. Yaru G Chandler G.	3
Using all the known knowledge from previous information, write down the problems that we are trying to solve with the app, and then the goals to solve those problems.	WRS [4.1] 1,2	Logan C.	1
Using new knowledge of Domain, Stakeholders, Functional, and Non-Functional create improved objectives for Domain, Functional, and Non-Functional requirements. Will also need to list the known Stakeholders for the entire project.	WRS [4.1.3] 1,2,3,4	Yaru G. Erik W Chandler G.	3
Finally, using all previous documentation, create system functional, non-functional, and specifications (this is user interaction and DB schema)	WRS [4.2] 1,2,3	Matthew H. Julian H Logan C	3
Create documented prototypes that explain the functional flow(maybe with some UML) of some of the features and how to use them in a user manual	WRS [5]	Yaru G. Chandler G. Matthew H.	3
Create a mock-up design of the user interface and how the user will interact with the system. (Possibly using Figma)	WRS[6]	Logan C. Erik W. Julian H.	3
Create **2** As-Is To-Be scenarios and then do further analysis on one of them using the practices presented in the lecture for lesson 6 in the course	AS-IS TO-BE	Matthew H.	1
Create a table chart comparing the advantages and disadvantages of the Theia app compared to a can or a dog.	AS-IS TO-BE	Logan C.	1
Collection of meetings as a record	MEETING RECORDS PHASE I	Matthew H.	1
Adding a backlog(Kanban) of tasks (OPTIONAL: including priority and story points)	None (However Important)	Erik W.	1
Front implementation of the Theia Application	Programming how the Users are going to interact with the device.	Logan C. Julian H.-0.5x Erik W. Yaru G.-0.5x	3
Backend implementation of the Theia Application	Programming the functionality that needs to be built in order for the Users to do what they need to	Yaru G. Chandler G. Matthew H.	3

	do.		

III. Managerial Process

III.i. Management Objectives and Priorities

We will break the project into small, trackable tasks and use Discord as our primary channel for posting assignments, updates, and quick questions. Tasks will be assigned in posts or during meetings, and we will rebalance workloads collaboratively if someone needs to switch or trade work. When trade-offs arise, we will put the deadline and required core features first, and we will put reliable, working code ahead of minor visual improvements.

Specific roles other than team liaison and focus areas are not finalized; as the project progresses and we better understand each member's skills and interests, we will clarify individual responsibilities and goals.

Team meetings will be held online in the Discord voice channel every Tuesday at 2:00 pm for 1 to 2 hours, with timing adjusted if conflicts arise. Communication with sponsors and mentors will go through the designated liaison, who will serve as the single point of contact and share summaries, decisions, and action items with the entire team to keep everyone aligned.

All deliverables will be stored in the shared GitHub repository; each person will work on a feature branch. When a task is complete, the deliverable will be submitted via pull request. The liaison will then evaluate it against the criteria in the course materials and this preliminary plan, and the team will review it in meetings to gather feedback and make revisions. Once the work meets the criteria, it will be merged into the main branch.

III.ii. Assumptions, Dependencies, and Constraints

This section makes our working conditions explicit so that planning and risk management are grounded in reality. Assumptions clarify what we expect to be true inside the team, dependencies identify what we rely on outside the team, and constraints define the non-negotiable limits we must work within. When any of these change, we can quickly decide whether to adjust scope, timeline, or approach.

Assumptions

AS-1: Team members attend scheduled meetings and respond within 24 hours on the primary channel (Discord).
AS-2: Before each weekly meeting, every member has at least some deliverables or a clear progress update ready to share.

AS-3: The team has baseline proficiency with the required tools (programming languages, frameworks, and workflows) to complete the project.

Dependencies

DE-1: All team members have read/write access to the GitHub repository.
DE-2: University and course infrastructure (Canvas and Perusall) remains available and functioning.
DE-3: Mentors and sponsors are reachable for questions and provide timely guidance.

Constraints

CO-1: Schedule is fixed, with phase and sprint deadlines set by the syllabus.
CO-2: Staffing is fixed at team size 6, with no additional members.
CO-3: No paid services are permitted; we will use open-source or academic-tier tools only.
CO-4: The technology stack must be suitable for a smartphone app and approved by the course, using frameworks and languages that provide the required device capabilities.

III.iii. Risk Management

We will be keeping a log to track potential issues and how likely they are. Additionally, the log will contain our plan to mitigate each specified risk. Risks will be reviewed in weekly team meetings and updated as the project progresses.

No.	Risk	Type	Likelihood	Impact	Mitigation/Response
1	Breakdown of team communication	Managerial	Low	Medium	Schedule regular meetings and check-ins. Use Discord for consistent communication. Document team decisions.
2	Missed or unclear deliverables	Managerial	Medium	High	Set clear deadlines and review all work before submission. Confirm what is expected early.
3	Loss of project files or version control	Technical	Low	High	Use Git for version control, proper branching, and commits. Store all project documents in the shared Google Drive.
4	Uneven distribution of work	Managerial	Low	Medium	Reassign work as needed, and keep a visible tracker of assigned tasks.
5	Loss of	Managerial	Medium	Medium	Have open discussions

	engagement from team member(s)				regarding any issues or worries about the project. Redistribute work if needed and keep the team accountable.
6	Schedule slippage from underestimating tasks	Managerial	Medium	High	Break tasks down into smaller blocks. Track weekly progress, make schedule adjustments early if needed.

Effort Estimate: We estimate that risk management will take ~5% of our project effort (roughly 1 hour per week for check-ins and updates).

III.iiii. Monitoring and Controlling Mechanisms

Progress will be monitored and tracked through our team's weekly check-ins, shared documents, and reports.

Info Communicated	From	To	Frequency
Status Update	Team Members	Project Manager	Weekly
Status Report	Project Manager	Instructor	Weekly
Project Review	Team	Instructor	Bi-weekly
Risk Log Update	Project Manager	Team	Weekly
Urgent Issues	Any Team Member	Team	As Needed

Monitoring Tools:

- **Weekly Meetings** - Check task progress, update risk log, and adjust schedule or task assignment if needed.
- **Shared Tools**: Git and GitHub for version control, Google Drive for documentation, and Discord as the main channel of communication.
- **Peer Review** - Review all project work before submission and make changes if needed.

IIII. Technical Process

IIII.i. Methods, Tools, and Techniques

Methods:
Firstly, we will be using the Agile development process. This means that we will work in 2-3 week sprints that include their own phases of planning, development, testing, deliverables, etc. This will allow, for example, prototypes and early versions to be previewed. As for the design method, we

will follow a user-centered design approach, where the requirements are derived from AS-IS and TO-BE scenarios. For example, features that assist visually impaired users.

Tools:
We will be using Discord for general communication and Google Docs for commenting on any peer-reviewed documents, such as this one. GitHub will be used as the code hub for version control and more. The code will be written in an IDE such as Visual Studio Code and tested in an environment like Android Studio. Figma will be used for early UI design brainstorming.

Techniques:
For general task distribution, we will be using a Kanban board on GitHub. Additionally, to gather further requirements, we will utilize user stories and scenarios.

IIII.ii. Software Documentation

This project's documentation will be produced and updated regularly as the team moves through each sprint. We will begin by creating a Vision and Scope document that explains the project's broad goals, boundaries, and objectives. Requirements will be captured using AS-IS and TO-BE scenarios, which describe how visually impaired users navigate and how the project aims to improve that process. These scenarios will then be expanded into a Software Requirements Specification (SRS) that outlines both functional and non-functional requirements clearly for both project stakeholders and developers. As work progresses, we will build an initial user manual with clear instructions for using the application's capabilities. We will also create notes and summaries for each meeting. In addition, version control records will be kept on GitHub to document code changes and project history. All documentation will be published via GitHub and Google Docs, with revisions at the end of each sprint and reviews at weekly meetings. Finalized versions will be submitted to Canvas at the end of each step.

V. Work Elements and Schedule

The **Visual Impairment App** project will be developed in several phases, with each phase building toward the final integrated prototype and presentation in **December 2025**.

Phase 1: Planning and Requirements (Aug. 18 – Oct. 12)

- Identify stakeholders (blind users, caretakers, accessibility staff).
- Develop AS-IS and TO-BE scenarios to capture real user needs.
- Build initial requirements models and clarify objectives.
- Draft the Vision & Scope and initial Software Requirements Specification (SRS).
- Document any meeting notes

- **Key Deliverables:** Preliminary Project Plan (Sept. 14) and Phase I Final Submission & Presentation (Oct. 12).

Phase 2: Prototype Development (Oct. 13 – Nov. 9)

- Create a low-fidelity prototype (basic navigation interface, voice command system, audio/tactile cues).
- Conduct walkthroughs and scenario testing to evaluate usability for blind users.
- Write a preliminary user manual to explain basic features.
- Document any meeting notes
- **Key Deliverables:** Prototype and User Manual (Nov. 9).

Phase 3: Refinement and Feature Expansion (Nov. 10 – Nov. 23)

- Improve voice guidance and route planning logic.
- Begin integration of obstacle detection using smartphone sensors.
- Check requirements and test the prototype with simulated blind-user scenarios.
- Document any meeting notes
- **Key Deliverables:** Refined prototype with expanded features and updated documentation (Nov. 23).

Phase 4: Integration and Demo Preparation (Dec. 1 – Dec. 7)

- Test full navigation flow, including route planning and accessibility functions.
- Refine user interface for clarity and comfort (audio prompts, gesture controls).
- Prepare the app for the live demonstration.
- **Key Deliverables:** Final integrated prototype, Phase II Submission, and Demo (Dec. 7).

Phase 5: Final Submission (Dec. 8 – Dec. 7)

- Complete final documentation: requirements, scenarios, prototype description, and user manual.
- Submit the final project package and present findings.
- **Key Deliverables:** Final Project Submission & Presentation (Dec. 7).