

---

# **Project Management Plan**

**for**

# **Theia**

**Version 1.0 draft 1**

**Prepared by Ross Kugler, Huy (Harry) Ky, Benjamin Bordon, Dylan Gyori, Camille Orego, Nick Sturgeon**

**LagZilla**

**September 3<sup>rd</sup>, 2025**

# Table of Contents

<b>Table of Contents .....</b>	<b>ii</b>
<b>Revision History .....</b>	<b>iii</b>
<b>1. Overview .....</b>	<b>1</b>
1.1. Project Purpose, Objectives, and Success Criteria .....	1
1.2. Project Deliverables.....	2
1.3. Assumptions, Dependencies, and Constraints.....	2
1.4. References.....	3
1.5. Definitions and Acronyms.....	3
<b>2. Project Organization.....</b>	<b>4</b>
2.1. Process Model .....	4
2.2. Organizational Structure.....	4
2.3. Roles and Responsibilities .....	5
<b>3. Managerial Process Plans .....</b>	<b>7</b>
3.1. Management objectives and priorities .....	7
3.2. Assumptions, dependencies, and constraints .....	7
3.3. Risk management.....	8
<b>4. Technical Process Plans.....</b>	<b>10</b>
4.1. Methods, tools, and techniques .....	10
4.2. Software documentation.....	10

## Revision History

Name	Date	Reason for Changes	Version
Preliminary Plan Draft 1	9/3/2025	Initial draft	1.0 draft 1
Preliminary Plan Draft 2	9/13/2025	Revised draft	1.1 draft 2

# 1. Overview

## 1.1. Project Purpose, Objectives, and Success Criteria

### *Our Purpose and Business Needs*

The primary purpose of the Theia app is to provide indoor navigation for visually impaired individuals. It addresses the business need for a reliable, smartphone-based solution that allows them to move independently from one location to another within a building.

The app will assist users with critical navigation tasks like:

- Determining their current location.
- Identifying the correct direction to walk in.
- Knowing when and where to turn.
- Detecting obstacles to avoid collisions.

### *Quantitative and Measurable Business Objectives*

- **Safety:** The app should detect and provide instructions to avoid **95%** of obstacles in a controlled testing environment.
- **Efficiency:** The app should guide a user to their destination along the fastest route with a navigation time within **5%** of the ideal travel time.
- **Usability:** The app should be easily usable for blind people, meaning all key functions, such as setting a destination, should be achievable through voice commands or tactile input with a **99%** success rate.

### *Success Criteria for Stakeholders*

- **Primary Stakeholder (Blind User):** The user can consistently navigate to their desired location safely and independently, feeling comfortable with the guidance provided.
- **Secondary Stakeholder (Caretaker):** The caretaker can easily configure the app for the user and receive emergency notifications if the user gets lost or injured.
- **Secondary Stakeholder (Staff/Police):** Emergency services can quickly and accurately locate a user in distress.
- **Secondary Stakeholder (Admin):** Staff with sufficient privileges able to update building layout on the phone due to changes in facility.

### *Project Deliverables*

The project will be delivered in phases, with **Phase I** focused on requirements gathering and prototyping. Key deliverables for this phase include:

- **Requirements Documentation:** A clear and comprehensive specification of the app's functional and non-functional requirements, using the provided WRS template.
- **Prototype and User Manual:** A simple mockup of the smartphone app and a preliminary user manual to demonstrate the core functionality and user experience.
- **Project Plan:** A detailed plan for Phase I, including team organization, roles, a schedule of deliverables, and regular meeting records.

- **Presentation:** A final presentation summarizing the project's purpose, a review of requirements, and the team's planned approach.

### **Relationship to Other Projects and Integration**

The Theia app is a standalone project. There are no direct dependencies on other ongoing projects. However, its development and future iterations may rely on:

- **Shared resources:** The project will use the team's GitHub repository for document and code management.

**Shared designs and components:** The app will need to integrate with a smartphone's existing hardware and software, leveraging sensors like GPS, accelerometers, and gyroscopes. It will also rely on the phone's voice recognition and text-to-speech capabilities.

## **1.2. Project Deliverables**

Deliverable	Delivery Date	Delivery Method	Comments
Project Phase 1: Preliminary Plan Submission	9/14/2025	Canvas/GitHub Repo	Preliminary Project Plan Document
Project Phase 1: Final Submission	10/12/2025	Canvas/GitHub Repo	WRS Revised Phase 1 Plan Meeting Records PowerPoint Slides Peer Review

## **1.3. Assumptions, Dependencies, and Constraints**

### **Assumptions (AS):**

- AS-1: Users will have smartphones with sensors (camera, accelerometer, gyroscope, microphone) and accessibility features enabled.
- AS-2: Buildings targeted for navigation will have or can provide floor plans or digital blueprint.
- AS-3: Users are familiar with built-in accessibility features (VoiceOver for iOS, TalkBack for Android), as the app will use integration with these systems.
- AS-4: The primary users are individuals who fall within the spectrum of visual impairment but do not have additional sensory disabilities.
- AS-5: Some users have a caretaker who can also access and interact with the phone on their behalf.

### **Dependencies (DE):**

- DE-1: Access to accurate building maps and/or positioning aids (Bluetooth beacons, or WiFi mapping), setup up by building staff rather than users.
- DE-2: Integration with smartphone accessibility APIs and voice assistant features.
- DE-3: User testing and feedback from visually impaired individuals to verify safety and usability.

### **Constraints (CO):**

- CO-1: Indoor GPS is inaccurate; alternative positioning methods must be used.
- CO-2: Smartphone battery and processing power limit continuous use of multiple sensors.
- CO-3: Safety and usability requirements are non-negotiable given the target user group.
- CO-4: Schedule is constrained by course deadlines for Phase I and Phase II.
- CO-5: The project must be completed without dedicated funding or external financial support.

### **Open Issues / To Be Determined (TBD) Items:**

- TBD-1: How to measure navigation distance (steps, minutes, or precise units).
- TBD-2: Whether destinations are defined at entrances or at specific points inside rooms.
- TBD-3: Extent of user customization (which features should always remain fixed for safety).
- TBD-4: How to balance safety vs. speed in route selection.
- TBD-5: Translation support for non-English users.
- TBD-6: Risk handling for specific indoor hazards (stairs, crowds, furniture).
- TBD-7: Emergency protocol design (automatic fall detection vs. user-initiated “panic button”).
- TBD-8: Define measurable criteria for vague requirements such as “easily extensible,” “ubiquitous,” and “comfortable route.”
- TBD-9: Clarify what makes the system “usable for blind people” in testable terms

## **1.4. References**

- *Project Management Plan Template* (Wiegers, 2013).
- *Requirements Review Checklist* (Wiegers & Seilevel, 2013).
- *GitHub Repository – Theia Project* - <https://github.com/rk3026/Theia>.

## **1.5. Definitions and Acronyms**

- Theia – Working name for the indoor navigation app.
- AS / DE / CO – Identifiers for Assumption, Dependency, Constraint.
- TBD – To Be Determined; flags open issues that require clarification or decision in later project phases.
- GPS – Global Positioning System (not reliable indoors).
- Bluetooth Beacons (iBeacon/Eddystone) – Small wireless transmitters installed in buildings that broadcast signals to nearby smartphones to assist with indoor positioning.
- WiFi Mapping – Use of WiFi signal strength or access point locations for indoor positioning when GPS is unavailable.
- Accessibility API – Smartphone accessibility features such as VoiceOver for iPhone and TalkBack for Android.

## 2. Project Organization

### 2.1. Process Model

This project will utilize the **Spiral Model** for the development process. The Spiral Model emphasizes risk management throughout all phases of development, which is particularly beneficial for Theia, given its reliance on smartphone sensors, accessibility features, and safety-critical usability. Each cycle of the spiral will incrementally refine prototypes and deliverables.

#### *Phases of the Spiral Model and Corresponding Tasks*

##### 1. Identify Objectives

- Define project scope and goals for the iteration.
- Gather and refine requirements (functional and non-functional).
- Engage stakeholders (users, caretakers, accessibility experts) for input.
- Prioritize requirements based on user safety, usability, and feasibility.
- Establish success criteria and measurable outcomes for the iteration.

##### 2. Risk Analysis

- Identify potential technical, usability, and safety risks.
- Evaluate alternative approaches (e.g., different indoor positioning methods such as Bluetooth beacons vs. WiFi mapping).
- Perform feasibility studies and proof-of-concept experiments.
- Rank risks by probability and impact (e.g., risk of inaccurate navigation vs. battery drain).
- Define mitigation strategies (e.g., fallback navigation methods, additional validation tests).

##### 3. Product Development

- Design system components (UI/UX mockups, navigation algorithms, data storage).
- Implement prioritized features using Android Studio and Flutter SDK.
- Conduct unit and integration testing of features.
- Create/update prototypes to demonstrate progress.
- Prepare associated documentation (updated WRS, diagrams, user manual drafts).

##### 4. Evaluation

- Conduct usability testing with visually impaired users where possible.
- Collect and analyze feedback from stakeholders.
- Validate results against success criteria (safety, efficiency, usability).
- Review project deliverables for completeness and quality.
- Decide on scope and focus of the next iteration.

### 2.2. Organizational Structure

The Theia project team is organized in a collaborative structure that emphasizes clear role assignments, effective communication, and accountability for deliverables. While the project

does not have a formal senior management layer due to its academic context, the team functions with a designated Communication Liaison for submitting deliverables.

### **Communication and Organizational Tools**

Team Member Communication: Discord  
 File Sharing: GitHub, Microsoft OneDrive  
 Deliverable Submission: Canvas

Optional tools considered for future development include Microsoft Teams for professional team communication, and Jira or Azure DevOps for project management.

## **2.3. Roles and Responsibilities**

The Theia team consists of Software Engineers who assume specialized roles to ensure that all technical and organizational aspects of the project are covered. Responsibilities are distributed to team members according to individual strengths. Each member contributes not only to their primary role but also to peer reviews, testing, and documentation. Table 1 outlines the roles and responsibilities of the project.

Role	Team Members	Responsibilities
Communication Liaison	Camille Orego	Submission of all deliverables, communication with professor, scheduling meetings
Hardware Research & Development	Dylan Gyori, Huy Ky	Researching and implementing phone hardware capabilities (sensors, GPS, accessibility features), ensuring proper integration with the app
UI/UX Developer	Ross Kugler, Camille Orego	Designing and implementing the mobile user interface, ensuring usability for visually impaired users, integrating accessibility standards
Navigation System Developer	Nick Sturgeon	Research, design, and implementation of the indoor navigation and routing algorithm
Floor Plan System Developer	Ben Bordon	Development of the floor plan creation and management system within the app



Database Specialist	Ross Kugler	Database schema design, setup, management, and integration into app features
Data Security Specialist	Ben Bordon	Designing and implementing secure data storage and access, protecting sensitive user data, ensuring compliance with accessibility and safety constraints
Cross-Platform Developer	Dylan Gyori	Ensuring app compatibility across devices and operating systems (Android and potential future iOS support)

Table 1. Roles and Responsibilities

In addition to individual roles, we assign specific project deliverables to an owner to ensure accountability and timely completion. Deliverables may involve collaboration between multiple team members, but each has a designated owner responsible for overseeing its completion.

Deliverable	Owner
Preliminary Project Plan Report	Ross Kugler
WRS	Dylan Gyori
Revised Phase 1 Plan	Ben Bordon
Meeting Records	Camille Orego
PowerPoint Slides	Camille Orego, Nick Sturgeon
Peer Reviews	All

Table 2. Deliverables and Owners

## 3. Managerial Process Plans

### 3.1. Management objectives and priorities

The project will be divided into multiple milestones and deliverables to ensure steady progress and effective collaboration among team members. This structured approach allows the team to adapt quickly to any necessary changes. Establishing multiple deadlines and checkpoints is essential to maintaining project momentum and ensuring timely completion.

#### *Communication Tools*

- Primary communication through Discord for tasks, deadlines, and updates.
- Tasks may also be communicated through a project management tool (e.g., Jira, Asana, Slack)
- Reinforced through weekly in-person/hybrid meetings.

#### *Communication Liaison Responsibilities*

- Communicate with stakeholders and schedule meetings
- Record and distribute all meeting information, including schedules, agendas, and notes
- Ensure all team members remain aligned and informed throughout the project lifecycle
- Organize and submit project deliverables

#### *Meeting Structure*

- **Frequency:** Weekly on Wednesdays (hybrid format).
- **Task Cycle:** Tasks assigned during one meeting are due by the following meeting.
- **Standard Agenda:**
  1. Validate completed tasks
  2. Discuss problems and issues
  3. Update backlogs in the project management tool
  4. Assign new tasks

### 3.2. Assumptions, dependencies, and constraints

#### *Constraints*

- **Time Management:** Each deliverable must be completed within the specified timeline; delays could hinder overall project progression and affect subsequent phases such as testing and deployment.
- **Academic Scheduling:** Milestones must align with course requirements, review periods, and deadlines set by instructors.
- **Limited Resources:** Access to testing devices and tools for accessibility validation is restricted. Additionally, team members must balance workload across other courses, which may limit availability.

#### *Assumptions*

- **Team Engagement:** All members will remain consistently engaged, attend scheduled meetings, complete tasks on time, and contribute evenly to the workload.
- **Stakeholder Availability:** Instructors, accessibility advisors, and other stakeholders will be available to provide timely feedback.
- **Stable Requirements:** Accessibility compliance requirements (e.g., WCAG standards) will remain unchanged during the project timeline.
- **Technology Reliability:** Existing accessibility APIs and tools (e.g., VoiceOver, TalkBack) will remain supported and functional throughout development.

### Dependencies

- **Internal Reliability:** The project relies on each team member's consistency in completing assigned responsibilities and participating in meetings to ensure clear communication.
- **Testing Resources:** The team depends on the availability of mobile devices and accessibility tools to validate app functionality.
- **End-User Feedback:** Input from blind and visually impaired individuals is critical to ensuring the app meets accessibility goals.
- **Stakeholder Oversight:** Progress depends on timely reviews, feedback, and approvals from stakeholders to maintain alignment with technical and user-focused requirements.

### 3.3. Risk management

We will create a separate document to manage risks in the project. This will be shared in the OneDrive with the team. We can also keep track of risks in project management tools like Azure DevOps.

Performing effective risk management is especially critical for a project of this magnitude, as the primary focus is to serve and support individuals with disabilities, particularly blind individuals, who are especially vulnerable to certain risks. Our team plans to implement a structured risk management process that encompasses multiple steps.

#### Risk Identification

- Risks will be identified continuously throughout all project phases (requirements, design, implementation, testing, deployment, and maintenance).
- All risks will be documented in a separate Risk List, with each entry including:
  - A description of the risk
  - Likelihood of occurrence
  - Potential impact
  - Mitigation strategy

#### Risk Analysis and Prioritization

- Each identified risk will be evaluated for probability and potential impact using a qualitative risk matrix (e.g., low, medium, high).
- Risks will be prioritized based on their potential effect on:
  - User safety
  - Accessibility
  - Data integrity
  - Project deadlines

### ***Risk Control and Contingency Planning***

- High-priority risks will include defined contingency actions such as:
  - Alternate design choices
  - Additional testing rounds
  - Fallback features to ensure critical functionality remains intact

### ***Ongoing Risk Management***

- Risk identification and review will remain an ongoing task throughout the entire project lifecycle, including post-release updates.
- Risk management activities will be fully integrated into the project schedule and budget to ensure they receive appropriate resources and attention

## 4. Technical Process Plans

### 4.1. Methods, tools, and techniques

This project will be an Android application which will be developed using Android studio, the Android SDK, and the Flutter SDK. This would provide a platform that has a low cost to develop and test on and easier access to a diverse set of phones and equipment. The development environment is a Windows system and Git/GitHub is used for version control. The hardware requirements to operate the application will need phone GPS, speakers, a touch screen, and network connectivity.

The development methodologies used to document requirements and features would include use of use case diagrams. Class diagrams and sequence diagrams will be used to plan the implementation of the requirements and features. The programming languages available to use with the Flutter SDK are Dart and C/C++.

Code testing will be done through code reviews utilizing GitHub's code review features and unit testing. The separate features of the app for its requirements will be tested separately in a development environment. Quality assurance will be practiced with real world use of the app with the visually impaired users that will test its use.

### 4.2. Software documentation

The Initial plan of the project will require the documents System Use Case Diagram, Risk Analysis, System Requirements Specification, and the Requirements Checklist. These documents serve to plan out the requirements of the project. The use case diagram describes the features and requirements on a high level while the system Requirements Specification goes into these requirements in more depth. The Risk analysis will help determine how suitable these features are.

The System Class Diagram and System Design Report serve to plan implementation details of the project. The class diagram is the low-level implementation details of each module in the project, and the system design report provides a detailed description and any necessary supporting documentation to further describe the implementation details for the project's requirements like a sequence diagram.

The User Guide is a document describing the functionality and use of the features to a caretaker for the user. This document goes over the requirements for the project that suits the user's needs. The Quality Assurance Plan provides a plan to test the functionality described in the User Guide document.

Document	Template or Standard	Created By	Reviewed By	Target Date	Distribution
WRS	WRS-template.rtf				GitHub, OneDrive

Requirements Checklist	Requirements Review Checklist.docx				GitHub, OneDrive
Risk Analysis					GitHub, OneDrive
System Requirements Specification					GitHub, OneDrive
System Use Case Diagram	UML				GitHub, OneDrive
System Class Diagram	UML				GitHub, OneDrive
System Design Report					GitHub, OneDrive
User Guide					GitHub, OneDrive
Quality Assurance Plan					GitHub, OneDrive