# Project Management Plan

**Version 2.1**

**Prepared by: Julia Lee, Shaya Arya, and Andrew Neal**

**CPTS 484_THEIA**

**12/01/25**

# 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 2
1.4. References 3
1.5. Definitions and Acronyms 3
1.6. Safety Considerations 4
1.7. Data Privacy & HIPPA Disclaimer 4
**2. Project Organization** 4
2.1. Process Model 4
2.2. Organizational Structure 4
2.3. Roles and Responsibilities 5
**3. Managerial Process Plans** 5
3.1. Management objectives and priorities 5
3.2. Assumptions, dependencies, and constraints 6
3.3. Risk management 6
3.4. Monitoring and Controlling Mechanisms 7
**4. Technical Process Plans** 8
4.1. Methods, tools, and techniques 8
4.1.1. Team Communication 8
4.1.2. Tools and Technologies 8
4.1.3. Sensor Utilization: 8
4.2. Software documentation 9
4.3. Project Timeline 10

# Revision History

| Name | Date | Reason for Changes | Version |
|---|---|---|---|
| Justin Keanini, Shaya Arya, Christian Flores Cruz, Hiruna(Yevin) Dissanayake, Russell Habib, Julia Lee | 09/14/25 | Project Phase I: Preliminary Plan Submission<br>Initial draft | 1.0 |
| Justin Keanini, Shaya Arya, Christian Flores Cruz, Hiruna(Yevin) Dissanayake, Russell Habib | 10/12/25 | Project Phase I: Preliminary Plan Resubmission | 1.1 |
| Julia Lee<br>Shaya Arya | 11/01/25 | Project Phase I: Preliminary Plan Revised | 2.0 |
| Andrew Neal | 11/20/25 | Added: sections 1.6, 1.7<br>Modified: section 4.1 | 2.1 |

# 1. Overview

It is difficult for individuals with vision disabilities to navigate inside buildings. They may be familiar with the route to their usual classroom, but locating an office, restroom, or any unfamiliar room becomes significantly more challenging. Currently, users must either rely on assistance from others or spend substantial time memorizing new routes. As a result, many visually impaired individuals avoid navigating to unfamiliar indoor locations altogether.

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

Globally, more than 285 million individuals experience blindness or significant visual impairment, many of whom face substantial challenges when navigating unfamiliar indoor environments. This project's purpose is to design and develop THEIA, an assistive mobile application that enables visually impaired users to navigate independently within buildings without needing help from others.

THEIA will achieve this purpose through the following objectives:

- Provide turn by turn indoor navigation that guides users from their current location to a desired destination.
- Detect and alert users to obstacles such as furniture, structural features, or moving individuals along their path.
- Operate entirely through voice commands and spoken responses to support hands-free interaction.
- Enable emergency assistance by detecting falls or allowing users to request help when disoriented.
- Determine optimal routes based on user selected criteria such as safety, travel time, or comfort.
- Allow customization of accessibility settings, including speech frequency, audio volume, and navigation feedback style.
- Support extensibility for future integration of additional sensors, features, or assistive technologies.

## 1.2. Project Deliverables

| Deliverable | Delivery Date | Delivery Method | Comments |
|---|---|---|---|
| Issues Analysis Document | Per course schedule | GitLab repository submission | Detailed analysis of deficiencies in the preliminary requirements and documentation of the corrective actions applied. |
| Requirements Specification (WRS) | Per course schedule | GitLab repository | Complete and formal description of the system’s functional and nonfunctional requirements based on the provided WRS template. |
| Working Prototype | Per course schedule | Mobile app demonstration | Early functional prototype demonstrating core navigation, feedback, and accessibility features. |

| | | | |
|---|---|---|---|
| User Manual | Per course schedule | Separate document | Accessible instructional guide for visually impaired users outlining how to operate the preliminary prototype. |
| Project Plan | Per course schedule | GitLab repository | Documentation outlining the project schedule, milestones, roles, and planned development activities. |
| Meeting Records | Ongoing / Final | GitLab repository | Documented summaries of team meetings, decisions made, and progress updates. |
| System Presentation | Per course schedule | In video conference | The final presentation demonstrates the system, scenarios, and results of the development process. |
| Requirements Models | Per course schedule | GitLab repository | Diagrams and structured models represent the system's behavior, data flows, and interactions. |

## 1.3. Assumptions, Dependencies, and Constraints

### 1.3.1 Assumptions (AS)

- **AS-1**: Visually impaired users will have access to a smartphone capable of running the THEIA application.
- **AS-2**: Users are already familiar with basic accessibility tools such as voice commands, screen readers, or audio feedback features.
- **AS-3**: Indoor building layouts used by the application remain stable and do not undergo frequent structural changes during the development and testing period.
- **AS-4**: The smartphone used by the user includes built-in calling and messaging capabilities for contacting emergency services or caregivers.
- **AS-5**: The device contains functioning onboard sensors, including accelerometers, gyroscopes, magnetometers, and GPS modules when applicable.

### 1.3.2 Dependencies (DE)

- **DE-1**: Open-source software libraries used for functionality such as text to speech or sensor processing remain maintained and compatible with current development tools.
- **DE-2**: Mobile operating systems continue to support required APIs, including accessibility services, background processing, and voice interaction.
- **DE-3**: Collaboration platforms such as GitHub, OneDrive, or equivalent services remain available and functional throughout the project timeline.
- **DE-4**: The team has access to an indoor testing environment where navigation and obstacle detection features can be evaluated consistently.

### 1.3.3 Constraints (CO)

- **CO-1**: **Timeline Constraint**. The project must be completed within the semester and delivered by the required deadline in early December
- **CO-2**: **Team Size and Expertise**. The scope of development is limited by the team's experience level and availability, which restricts the complexity and volume of features that can be implemented
- **CO-3**: **Budgetary Constraint**. Only free, open source, or university provided software, tools, and equipment may be used due to the absence of dedicated funding

- **CO-4**: **Scope Constraint**. The project emphasizes the development of a functional prototype focusing on indoor navigation and safety features rather than a fully deployable final product.

## 1.4. References


- Holton, B. (n.d.). Smartphone GPS navigation for people with visual impairments. American Foundation for the Blind. https://www.afb.org/blindness-and-low-vision/using-technology/smartphone-gps-navigation-people-visual-impairments
- American Foundation for the Blind. (n.d.). Accessible mobile apps. https://www.afb.org/blindness-and-low-vision/using-technology/assistive-technology-products/mobile-apps
- Rawat, A. (2025, September 9). How to design accessibility apps for visually impaired users. https://appinventiv.com/blog/design-accessibility-app-for-visually-impaired
- Murolo, S. (2024, May 21). Designing an app for visually impaired people is important. https://medium.com/@sossiomurolo/designing-an-app-for-visually-impaired-people-dc14cf622496


## 1.5. Definitions and Acronyms

• **WRS**: Requirements and Specification Document. A formal document that analyzes deficiencies in the initial requirements and defines the system's functional and non-functional requirements through structured goals and specifications.

• **Agile**: A flexible software development methodology that emphasizes iterative progress, collaboration, and incremental delivery.

• **Sprints**: Time for boxed development cycles, typically one week in duration, during which a set of features is planned, implemented, and reviewed before moving to the next cycle.

• **QA (Quality Assurance)**: The process of evaluating the system to ensure that all functional and non-functional requirements are met through systematic testing and verification.

• **Deliverables**: Tangible and intangible outputs produced during the project, such as reports, prototypes, documentation, or presentations.

• **UI (User Interface)**: The components through which users interact with the system, adapted to this project to support visually impaired users through accessible interaction methods.

• **SDKs (Software Development Kits)**: Collections of development tools, libraries, documentation, and testing components used to build applications for specific platforms.

• **APIs (Application Programming Interfaces)**: Interfaces that allow the application to access external services, system resources, or device level capabilities.

• **CI (Continuous Integration)**: A development practice in which code changes are built, tested, and merged frequently to maintain system stability and reduce integration issues.

## 1.6. Safety Considerations

As Theia is primarily designed to guide blind or visually impaired individuals, safety is a primary concern in both the design and deployment stages of the application. Theia must ensure that its navigation and guidance instructions minimize the risk of user injury, while also accounting for limitations or errors in both the software and hardware that the app is running on.

To promote our user's safety, Theia will incorporate the following features:

**Fall Detection and Response:**
Theia will use the accelerometer and gyroscope of the device that it is running on to detect sudden changes in the user's motion or orientation. If a large enough change is detected, Theia will immediately prompt the user via auditory feedback to confirm whether they are safe. If the user does not respond within 5 seconds of the prompt via voice response, the user's designated emergency contact provided during the user's initialization will be contacted. If the user did not choose to provide an emergency contact during initialization, emergency services will be contacted instead and provided with the user's precise location.

**Obstacle Detection and Warning**
Theia will incorporate obstacle detection by utilizing computer vision models to provide users with reliable and safer indoor navigation. Since obstacles such as wet floor signs or boxes cannot always be provided in floorplans, Theia must provide users with real-time object detection and warnings to prevent injury.

## 1.7. Data Privacy & HIPPA Disclaimer

Theia does not collect, sell, or transmit any personal information provided to the user. All user data, including floor plans, are stored locally on the user's device and never connected to a server. All navigation is computed directly on the user's device, ensuring that current or previous routes cannot be tracked outside of the user's device. Theia is not a medical device or application and is intended solely to provide users with navigation and safety assistance tools.

# 2. Project Organization

## 2.1. Process Model

For this project, we adopted an agile software development process that supports flexibility, iterative refinement, and continuous integration of new features. Development will be conducted in structured sprints, during which the team will define specific tasks, evaluate progress, and adjust plans for subsequent cycles. This model is well suited for THEIA, as the system requires ongoing testing, validation, and revision based on user feedback and evolving functional requirements.

## 2.2. Organizational Structure

The project team consists of Julia Lee and Shaya Arya. The team will operate under a collaborative structure in which all members contribute to design, implementation, documentation, and testing activities. To support coordination and accountability, Julia Lee serves as the team lead, overseeing milestone alignment, communication, and task distribution while ensuring that progress remains consistent with the project plan.

### 2.3. Roles and Responsibilities

| Name | Role | Responsibility |
|---|---|---|
| Julia Lee | Technical Lead | Oversees overall technical development and coordinates integration across system components. Drafts, revises, and maintains the Requirements Specification. Manages meeting records and leads to the final system of presentation. Supports software development and ensures code quality and feature implementation. |
| Shaya Arya | UI Designer | Drafts, revises, and maintains the Requirements Specification. Creates high fidelity UI prototypes using tools such as Figma and designs user interaction flows. Develops accessibility focused interface documentation and user manual components. Builds UI and interaction elements required for prototype implementation. |
| Andrew Neal | Software Developer | Drafts, revises, and maintains the Requirements Specification. Supports software development and ensures code quality and feature implementation. |

## 3. Managerial Process Plans

### 3.1. Management objectives and priorities

The management approach for this project is based on dividing the work into clear, achievable tasks with well-defined deliverables. All tasks will be explicitly assigned and tracked, and communication will be maintained primarily through Discord and email. Accountability is ensured through the shared understanding that the success of the project depends on each member’s commitment and timely completion of assigned responsibilities.

Progress will be reviewed regularly, particularly given the uncertainty surrounding the final scope and implementation of the system. Due to current skill levels and evolving technical knowledge, creating long-term deadlines and highly detailed deliverables may not be feasible at the initial stages. As the team advances through research, prototyping, and early development, more detailed and extended deliverables will be established.

Clear communication regarding team availability will be essential. This will be supported through recurrent check-ins during meetings held multiple times per week. Communication with sponsors and mentors will be coordinated by the designated liaison, who will relay relevant information to the appropriate team members.

## 3.2. Assumptions, dependencies, and constraints

This project proceeds assuming that no dedicated budget is available. Team members will rely on their personal smartphones and development equipment, and any university-provided resources or loaner programs accessible through standard procedures. It is also assumed that students will retain access to collaboration platforms and an appropriate indoor testing environment, which will be used throughout the semester for receiving feedback from the instructor and teaching assistants in alignment with the course schedule.

On the dependency side, the project relies on continued access to essential technologies, including the selected backend framework, primary UI libraries, and the platform's target SDKs and accessibility of APIs. The availability of suitable hosting or version control environments, as well as continuous integration tools or equivalent manual support, is also required. In addition, external data sources or third-party APIs must remain accessible if integrated into the system.

The project is constrained by the semester timeline and the absence of financial resources, making scope control and careful prioritization critical. Safety, privacy, and accessibility requirements impose necessary restrictions on system design decisions. Additional constraints include platform level policies, device level limitations such as battery life and performance, and the variability of indoor network or radio conditions, all of which may restrict feasible features or testing methods. If direct user studies become necessary, the team will obtain prior approval and operate under a minimal data collection principle to ensure ethical compliance.

## 3.3. Risk management

| ID | Risk | Category | Likelihood | Description |
|---|---|---|---|---|
| R1 | API Access Issues | Technical | Likely | The team may lose access to required SDKs, accessibility of APIs, or system permissions, which can restrict core feature development. |
| R2 | Failure to Meet Sprint Deliverables | Managerial | Very possible | Planned sprint tasks may not be completed on time, causing schedule delays and reduced progress toward major milestones. |
| R3 | Insufficient User Feedback | Technical | Not likely | Lack of feedback from real users, especially visually impaired individuals, may lead to poor design decisions and an ineffective navigation experience. |

| | | | | |
|---|---|---|---|---|
| R4 | Third Party Dependency Failures | Technical | Likely | The system may rely on external libraries such as text to speech, which could introduce instability if they become deprecated, updated incorrectly, or incompatible. |
| R5 | Data Loss or Version Control Errors | Technical | Unlikely | Poor Git practices or local storage issues may lead to loss of source code, prototypes, or documentation. |
| R6 | Limited Team Availability | Managerial | Likely | Team members may miss meetings or have limited availability, leading to delays, miscommunication, or incomplete work. |
| R7 | Scope Creep | Managerial | Unlikely | Additional feature requests beyond the team's capacity may expand the project's scope and exceed time or resource constraints. |

## 3.4. Monitoring and Controlling Mechanisms

| Number | Area | Monitoring and Controlling Mechanisms |
|---|---|---|
| 1 | Reporting Mechanisms | • Daily or frequent Scrum stand ups for progress updates and blocker identification.<br>• Milestone completion reports with supporting documentation.<br>• Centralized risk and issue log reviewed regularly during stand ups. |
| 2 | Reporting Formats | • Standardized progress reporting template established by the team.<br>• Milestone code reports including technical documentation such as README files.<br>• Feature request forms to evaluate and prevent scope creep by confirming feasibility. |
| 3 | Review Mechanisms | • Weekly Scrum review meetings to assess progress and plan upcoming work.<br>• Walkthroughs of completed system components to ensure adherence to quality and accessibility standards. |
| 4 | Quality Assurance (QA) | • Code reviews to maintain quality, readability, and consistency with coding guidelines.<br>• Functional and non-functional testing to confirm successful code performance. |
| 5 | Configuration Management | • Version history and compatibility documentation to ensure consistency across releases.<br>• Maintained changelog to record new features, revisions, and fixes for transparency. |
| 6 | Documentation Management | • Central repository for all code and documents, with sub-repositories or branches as needed.<br>• Continuous updates to documentation during Scrum meetings to reflect recent changes.<br>• Document reviews during code review cycles to ensure accuracy and completeness. |

# 4. Technical Process Plans

## 4.1. Methods, tools, and techniques

### 4.1.1. Team Communication

Our team will employ an Agile, iterative development process with weekly check-ins to refine requirements, update design artifacts, and incrementally advance the prototype. This approach supports continuous improvement as new insights emerge during development.

Communication and collaboration will be conducted using several tools:

- **Discord** for real time communication, scheduling, and quick issue resolution.
- **GitHub** for storing source code, finalized documents, and version-controlled project artifacts.
- **One Drive** for collaborative documentation drafting, shared editing, and organizing reference materials.

These tools collectively support coordination, transparency, and traceability throughout the development of a lifecycle.

### 4.1.2. Tools and Technologies

Theia is designed to work with most smart phones on the market including Android and Apple devices. To accomplish this, Theia will be developed using React 19 which enables cross-platform deployment from a single codebase, utilizing existing libraries to access the user's device's sensors. This approach allows for us to easily access key components of the user's device outside of the touchscreen

cross-platform deployment from a single codebase. This approach also allows us to utilize key components of the user's device beyond the screen such as the device's speaker, microphone, accelerometer, and gyroscope, all of which are key components of the application's safety features. These components allow Theia to provide real-time auditorial feedback directly to the user and detect their responses as well as detect sudden changes in the user's movement and orientation which may indicate a fall.

To support development and ensure technical feasibility, the following tools and libraries will be used during the development of Theia:

- **React Native + Expo:** Used for cross platform development and UI design of Theia
- **React Native TTS:** Used to convert text on the user's screen into clear, easily understandable auditory output.
- **Local Storage:** For storing users downloaded floor plans and saved routes securely on the user's device
- **React Native Sensors:** Used to gain access to the user's device's accelerometer and gyroscope for fall detection
- **Environmental Analysis via Google Gemini:** Used in connection with the device's camera sensor to detect objects in real time by describing the users' immediate environment

### 4.1.3. Sensor Utilization:

Theia utilizes the following sensors on the user's device to maximize the applications functionality for the user:

- **Camera:** Enables theia to view what is in front of the user for real time object detection
- **Microphone:** Allows the user to interact with Theia verbally and confirm safety if a fall is detected
- **Speaker:** Allows Theia to provide the user with auditory instructions and alerts
- **Accelerometer:** Utilized to detect a possible fall from the user by detecting sudden changes in movement
- **Gyroscope:** Utilized to detect a possible fall from the user by detecting sudden changes in the user's orientation

## 4.2. Software documentation

All project documentation will adhere to the structure and standards defined in the WRS template course. This includes:

- Detailed functional and non-functional requirements
- Issues analysis and improved understanding documentation
- Design notes and architectural descriptions
- Testing plans, including verification steps for core features

Also, a concise user guide will be developed to support the preliminary prototype demonstration. This guide will focus on accessibility, ensuring that visually impaired users can understand how to operate the system effectively.

| Document | Template or Standard | Created By | Reviewed By | Target Date | Distribution |
|---|---|---|---|---|---|
| Issues Analysis Document | WRS Template | Whole Team | Instructor/ Peers | September 14 | Canvas Submission |
| Requirements Specification (WRS) | WRS Template | Whole Team | Instructor/ Peers | December 1 | Canvas Submission |
| Mockup Sketch of GUI | Team Formatted | Shaya Arya | Instructor/ Peers | December 1 | Canvas Submission |
| Description of user interfaces | Team Formatted | Shaya Arya | Instructor/ Peers | December 1 | Canvas Submission |
| AS-IS, TO-BE Project Power point slides | Power Point Slides (Team Formatted) | Julia Lee | Instructor/ Peers | December 1 | In Zoom |
| User Manual | Team Formatted | Shaya Arya | Instructor/ Peers | December 1 | Canvas Submission |
| Peer Review | Individually Formatted | Whole Team | Instructor | December 1 | Canvas Submission |
| Final Project Plan | Project Plan Template | Whole Team | Instructor | December 1 | Canvas Submission |

| | | | | | |
|---|---|---|---|---|---|
| Meeting Records | Standard Minutes | Julia Lee | Instructor | December 1 | Canvas Submission |
| System Prototype Development | Code | Whole Team | Instructor | December 1 | GitHub |
| System Prototype Presentation | Power Point Slides (Team Formatted) | Whole Team | Instructor | December 1 | In Zoom |

## 4.3. Project Timeline

| Week | Date Range | Deliverables |
|---|---|---|
| Week 1 | Aug 18 – Aug 24 | Form Team |
| Week 2 | Aug 25 – Aug 31 | Meet Team |
| Week 3 – 4 | Sep 1 – Sep 14 | Preliminary Project Plan |
| Week 5 – 6 | Sep 15 – Sep 28 | Continue elicitation and start issue analysis |
| Week 7 | Sep 29 – Oct 5 | AS-IS, TO-BE PowerPoint Slides |
| Week 8 | Oct 6 – Oct 12 | WRS, GUI Mockup, Description of User Interfaces, Revised Preliminary Project Plan, Meeting Records, Peer Review |
| Week 9 – 10 | Oct 13 – Oct 26 | Begin Phase II planning |
| Week 11 – 12 | Oct 27 – Nov 9 | Optional Phase II Checkup meeting |
| Week 13 – 14 | Nov 10 – Nov 23 | Work on phase II and begin Prototype development |
| Thanksgiving break | Nov 24 – Nov 30 | - |
| Week 15 | Dec 1 – Dec 7 | Project Phase II Final Submission, Prototype Demo |