# Project Management Plan

for

# Theia: Indoor Navigation Assistance App

Preliminary Project Plan

Prepared by Felix Zheng, Ziming Fu, Tong Shen, Jinming Wang, Zirun Ye, Zijian Zhang

CPTS 484

9/14/25

# Table of Contents

# Revision History

| Name | Date | Reason for Changes | Version |
|---|---|---|---|
| Project Management Plan for Theia: Indoor Navigation Assistance App | Sep 14 | initial draft | 1.0 draft 1 |

# 1. Overview

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

The purpose of the Theia project is to deliver a smartphone application that assists blind and visually impaired individuals with accurate, accessible, and reliable indoor navigation. The app addresses the business need of improving accessibility within buildings (classrooms, offices, labs, restrooms, lounges, etc.) by providing a system to support blind people navigating inside buildings. Our primary stakeholders are blind people who will be directly using this app to navigate indoors. Secondary stakeholders will be anyone who is involved with helping navigate blind people, this includes the following: caretakers of blind people, public servants (e.g. police), hospitality staff, etc. Specific quantitative and measurable business needs that the app is expected to meet include the following:

- Reduce navigation time by at least 20% compared to unguided navigation (measured via pilot tests).
- Achieve at least 90% accuracy in obstacle detection and collision avoidance.
- Enable users to reach their intended destination with >95% successful route completion rate.
- Provide real-time navigation updates with latency <2 seconds.
- Emergency alert delivery within <10 seconds of activation.

Criteria to measure satisfaction of the previously mentioned needs according to the stakeholders:

- User testing surveys with blind participants (target satisfaction rating ≥ 4/5).
- Performance benchmarks for speed, safety, and reliability compared against baseline mobility tests.
- System usability testing (ease of use, voice command accuracy, customizability).
- Accessibility compliance (WCAG 2.1 AA standards and ADA guidelines).

Scope:

This app is only intended for indoor navigation and is not intended to be a comprehensive app for navigation. The mode of this app will be voice-based interaction at release, although our system should be easily modifiable for future upgrades. It will also have emergency situation services, with the extent of it being automated notifications for emergency services.

Objectives:

- Usability Objectives

  - Task success rate: ≥ 95% of users can start a route and reach the destination without human assistance on the first try.
  - Quiet Error-free voice command rate: ≥ 97% of critical commands (start/stop/confirm/cancel) recognized correctly in quiet indoor environments;
  - Noisy Error-free voice command rate: ≥ 92% in noisy halls (65–75 dB).
  - Learnability: Repeat task time improves by ≥ 20% from trial 1 to trial 3.
  - Accessibility conformance: Meets WCAG 2.1 AA for non-visual UX and platform screen reader guidelines (TalkBack/VoiceOver).
  - Quantifiable Customizability: Users are able to adjust settings on a numeric scale for volume, instruction interval, and playback speed.
  - Mode customizability: Users can choose between the fastest route and the most "comfortable" route.
  - User satisfaction: Mean SUS score ≥ 80
  - User prediction: Predict

- Performance Objectives
  - Instruction latency: Time from detected waypoint/turn to spoken prompt ≤ 500 ms
  - Position location from intialization: Initial indoor localization ≤ 5 s
  - Route recalculation: After deviation, new guidance in ≤ 2 s
  - Obstacle detection latency: From first sensor indication to alert ≤ 300 ms
  - Detection accuracy: Obstacle detection TPR ≥ 92%
  - Uptime: Backend/service availability ≥ 99.5% monthly

- Security Objectives
  - Data minimization: Store only pseudonymous user ID; location histories cleared
  - Encryption in transit: 100% of network traffic via TLS 1.2+ with HSTS
  - Encryption at rest: Sensitive data (emergency contacts, preferences) AES-256 on device; server data encrypted by KMS.
  - Access control: Principle of least privilege for end users
  - Consent & transparency: Explicit consent flows for mic, motion, location; privacy policy accessible via voice. 100% permission-gated features blocked until consent
  - Auditability: Security-relevant events (login, emergency trigger, settings change) logged with integrity protection; log retention ≥ 90 days.

- Safety Objectives
  - Collision avoidance: During guided sessions, 0 collisions with static obstacles; near-miss rate (within 20 cm) ≤ 2% of encounters. Observed trials with ground truth.
  - Fall detection: Sensitivity ≥ 95%, specificity ≥ 90%; false alerts ≤ 1 per 8 hours active use. Simulated and staged falls with harness.
  - Emergency dispatch time: From trigger to SMS/call placed ≤ 10 s (p95); confirmation feedback to user ≤ 2 s. End-to-end test with test contacts.
  - Safe-stop behavior: On localization loss or ambiguous path, app issues "pause and reorient" within ≤ 1 s and provides recovery steps. Fault-injection tests.
  - Instruction pacing: Minimum 2 s spacing between prompts; no overlapping instructions; critical alerts pre-empt lower-priority cues. HMI rule checks + logs.

- Risk controls validation: Known hazards (stairs, glass doors, silent obstacles) called out with ≥ 90% detection or pre-mapped warnings. Hazard scenario suite.
- Redundancy: Dual-modality sensing (vision + IMU) or (beacons + IMU) for critical events; failover verified in 100% of induced sensor-failure tests. Kill-sensor drills.
- Geofenced hazards: If building map marks restricted zones, entry prevention warnings issued with ≥ 95% reliability. Map-tag test passes.

Delivered Products:

Mobile app

Project dependencies:

- Relationship with other projects:
  - Aligns with institutional accessibility initiatives (e.g., ADA compliance, accessibility office programs).
  - May integrate with broader smart campus or smart building projects (IoT sensors, building maps).
- Integration with other products:
  - Integration with indoor positioning systems (e.g., Wi-Fi/Bluetooth beacons).
  - Emergency integration with SMS/call services, accessibility department systems, or campus security.
  - May use AI photo recognition for object detection
- Feature dependencies:
  - Indoor mapping/positioning services (Bluetooth beacons, Wi-Fi triangulation).
  - Smartphone sensors (accelerometer, gyroscope, camera).
  - Voice recognition services (e.g., Google Voice, SiriKit).
- Schedule dependencies:
  - Availability of pilot test participants (blind/visually impaired users).
  - Delivery of campus building maps or digital floor plans.
- Shared designs, code, and hardware:
  - Potential reuse of open-source navigation frameworks.
  - Shared access to Bluetooth/Wi-Fi beacon infrastructure.
  - Dependency on smartphone hardware capabilities.
- Availability of shared components:
  - Indoor positioning APIs.
  - Existing assistive technology APIs (e.g., VoiceOver, TalkBack).
  - Resource use of smartphone

## 1.2. Project Deliverables

| Deliverable | Delivery Date | Delivery Method |
|---|---|---|
| Project Phase I: Preliminary Project Plan | Sep 14 | Canvas |
| Project Phase I: Final Submission/Presentation | Oct 12 | Canvas |
| Project Phase II: Final Submission | Nov 2 | Canvas |
| Summary Report Submission | Dec 7 | Canvas |

## 1.3. Assumptions, Dependencies, and Constraints

Features - Success Driver:

- AS-1: Blind users have access to a smartphone with standard sensors (accelerometer, gyroscope, camera, microphone).
- AS-2: Digital building maps or layouts are available for route planning.-
- AS-3: Ease of use is a priority for our end users
- AS-5: Android 7.0 and up.

Quality - Success Driver:

- AS-4: The buildings have both wifi and Bluetooth beacons
- DE-1: Indoor positioning accuracy depends on external infrastructure (Bluetooth or Wi-Fi).
- DE-2: Emergency communication depends on network connectivity.

Schedule - Constraint:

- CO-2: Project Phase 1 due Oct 12
- CO-3: Phase 2 Due Nov 2
- CO-4: Final submission due Dec 7

Cost - Constraint:

- CO-1: Project budget limited to existing hardware/software resources (Bluetooth beacons, smartphones, free/open APIs).
- AS-6: Our development team has standard laptops with Windows and at least 16GB RAM and GPU acceleration for machine learning training, and an Android phone for testing.

Staff - Constraint:

- CO-5: Staff consists of 6 employees

### 1.4. References

"ArcGIS IPS." *Indoor Positioning System & Location Tracking (Indoor GPS)*, www.esri.com/en-us/arcgis/products/arcgis-ips/overview. Accessed 13 Sept. 2025.

"Web Content Accessibility Guidelines (WCAG) 2.1." W3C, www.w3.org/TR/WCAG21/. Accessed 13 Sept. 2025.

"Guidance on Web Accessibility and the Ada." ADA.Gov, 29 July 2025, www.ada.gov/resources/web-guidance/.

"Extensible Markup Language (XML) 1.0 (Fifth Edition)." *W3C*, www.w3.org/TR/xml/. Accessed 13 Sept. 2025.

Rossum, Guido van, et al. "PEP 8 – Style Guide for Python Code." *Python Enhancement Proposals (PEPs)*, PHI Learning Pvt. Ltd., 5 July 2001, peps.python.org/pep-0008/.

"Management by Objectives (MBO): Definition and Process [2025] • Asana." *Asana*, Asana, 22 May 2025, asana.com/resources/management-by-objectives.

### 1.5. Definitions and Acronyms

Bluetooth beacon: Transmit Bluetooth signals to other Bluetooth enabled devices→ in our use case we will be using this to pinpoint location of smartphones

ADA: Americans with Disability Act

- US Government agency

IoT: Internet of Things

- Smart devices

SuS: System Usability Score

TPR: True Positive Rate

- True Positives /( False Negatives+ True Positives)

# 2. Project Organization

### 2.1. Process Model

We will use **Agile model**
Why we choose Agile model

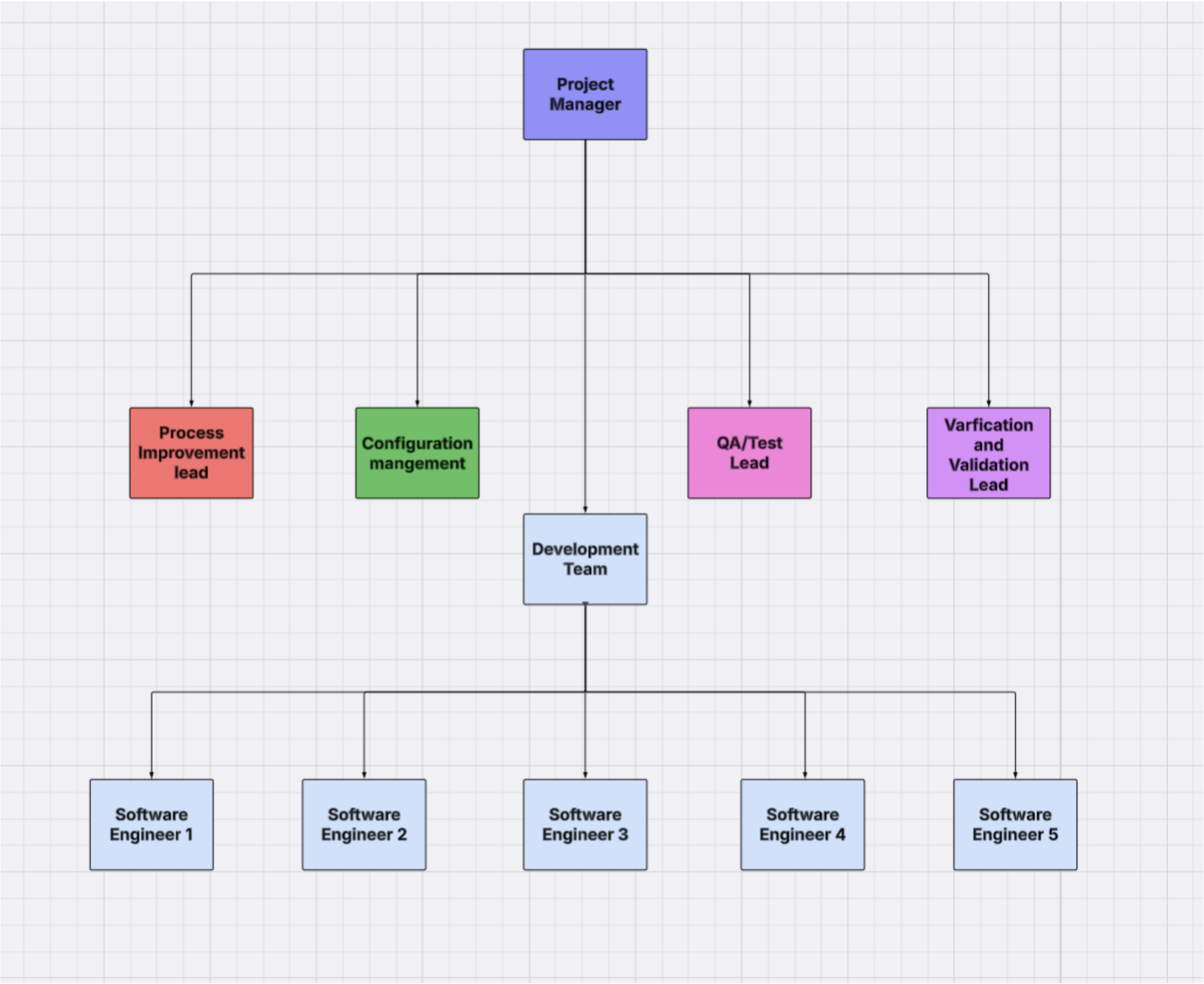
1. User needs are unclear: The experience blind users experience when using the system (whether voice prompts are clear, routing is reasonable, and positioning is accurate) requires real-world testing and cannot be fully defined in one go.
2. External software is relied on: As mentioned in DE-1, Indoor navigation may involve multiple technologies such as Bluetooth Beacons, WiFi, NFC, QR codes, and AI recognition, requiring ongoing testing.
3. Quick feedback is crucial: As mentioned in AS-3, this app is to help accessibility so it needs to be easy to use. Blind users are a special group, and continuous user feedback is essential to ensure the system is truly usable.

Therefore, this project adopts the Agile development model, carries out development in an iterative manner using Scrum, and combines prototyping to gradually converge requirements and implementation.

We will divide this project into several sprints, each lasting several weeks. Each sprint will accomplish different goals, for example:

1. Sprint 1: Implement basic room search and voice prompts.
2. Sprint 2: Implement indoor path planning.
3. Sprint 3: Implement bug fixes and user testing.

## 2.2. Organizational Structure



1. Project Manager:
This project adopts a layered and functional organizational structure. The top layer is managed by the Project Manager, responsible for overall management and external communications. The middle layer comprises four key support functions (process improvement, configuration management, quality assurance and testing, and verification and validation). The bottom layer is comprised of a team of software engineers, responsible for specific development tasks.

2. Management Layer
The Project Manager
Located at the highest level of the organizational structure, is responsible for overall project planning, resource coordination, and schedule control.
Serving as the primary external interface, they maintain communication with clients, advisors, and external stakeholders.
They liaise with the four key functional departments and ensure coordinated operations.

3. Engineering Support Layer
The four key functional leaders report directly to the Project Manager:

The Process Improvement Lead
Drives continuous improvement of team processes, leads iteration reviews, and proposes optimization solutions.

4. Configuration Management
Responsible for source code version control, branch management, builds, and releases.
Ensures the order and traceability of all development artifacts (code, documentation, and configuration).
Maintains a unified baseline environment when collaborating across teams.

5. QA/Test Lead
Develops test plans and test cases. Lead system-level testing, regression testing, and accessibility testing to ensure product quality standards are met.
Work closely with the development and verification teams to identify and track defects.
Verification and Validation (V&V) Lead
Responsible for verification (confirming that the system correctly implements specifications) and validation (confirming that the system meets actual user needs).
In this project, the focus is on confirming whether blind users can successfully complete indoor navigation tasks using voice and tactile feedback.

### 2.3. Roles and Responsibilities

| **Role** | **Team member** |
|---|---|
| Project Manager | Zijian Zhang |
| Test Engineer(Verification and Validation) | Zirun Ye, Jinming Wang |
| Technical Engineer (Configuration) | Felix Zheng, Ziming Fu |
| Quality Engineer(Quality) | Ziming Fu, Felix Zheng |
| Requirement Analyst(process Improvement) | Zirun Ye |
| Software Engineer | Zijian Zhang |
| Front-end | Tong Shen |
| Back-end | Jinming Wang |
| Database | Zijian Zhang, Tong Shen |

**Project Manager:**
Responsible for the overall project progress, resource coordination, and communication with clients/mentors.

**Software Engineer:**
Front-end:
As the system's user interface (UI) and user experience (UX).
It is responsible for direct user interaction (entering the desired room → receiving a voice/vibration notification).
It converts user actions into requests sent to the Backend API and displays the results in an appropriate manner.
Can be very simple.

**Back-end:**
core logic layer of the entire system, responsible for:
Storing and managing data (room information, building maps, positioning signals);
Processing business logic (path planning, navigation algorithms);
Interfacing with the frontend (mobile app/voice interaction) and providing an API interface;
Ensuring system security, stability, and scalability.
Need a lot of work

**Database:**
Responsible for storing building space data, navigation maps, positioning anchors, accessibility information and operation logs, and provides highly consistent, scalable and low-latency data access for back-end path planning and front-end queries.

**Test Engineer:**
Responsible for Verification and Validation:
Verification: Checks whether the system is correctly implemented according to the specifications (functions, interfaces, and algorithms).
Validation: Checks whether the system truly meets the needs of blind users (usability, accessibility, and fault tolerance).

**Technical Lead:**
Ensure every developer is on the same development platform

- Project Manager
- Product Manager
- Technical Lead
- Software Lead
- Hardware Lead
- Architect
- Systems Engineer
- Requirements Analyst
- Software Engineer
- Hardware Engineer
- Test Engineer
- Configuration Control Board
- Configuration Management Manager or Coordinator

- Quality Assurance Manager, Coordinator, or Engineer
- Technical Applications Support
- Subject Matter Expert

Make sure deliverables in Section 1.2 is covered in the Responsibilities, and proper team members are assigned as owner of each responsibility.>

# 3. Managerial Process Plans

## 3.1. Management objectives and priorities

The management objective of the team is to collaborate efficiently through fair task distribution and to meet deadlines by assigning roles according to each member's strengths and areas for growth. To ensure a high-quality final product, the project will be broken down into smaller, manageable tasks, allowing all members to contribute efficiently, support one another, and complete work on time.

The management priorities are to maintain clear communication through weekly meetings and group chats on Discord, track deadlines, and assign roles using a shared Google File. Any ambiguities in requirements will be resolved collaboratively with all project stakeholders, either during team meetings or through the designated team liaison. Additionally, the team will meet before each deadline to review and revise all deliverables to measure and ensure quality. The finalized product will then be submitted collectively by the team liaison.

## 3.2. Assumptions, dependencies, and constraints

The functionality of the application depends on several assumptions and constraints; thus, management should develop quality measurement according to these constraints that are related to management. As mentioned in CO-2, CO-3, and CO-4, we have specific deadlines to meet, and management needs to schedule tasks based on these deadlines. In CO-1, we mention that our budget is limited, so we cannot afford to buy software online that could assist us, and we only have access to software that is open source or that we develop ourselves. In CO-5, we note that we have 6 employees, so we have a limited amount of programmers to divide tasks among.

## 3.3. Risk management

| No. | Risk | Type | Likelihood | Description |
|---|---|---|---|---|
| 1 | Inadequate sensor accuracy and reliability | Technical | Likely - High potential impact | Smartphone sensors may not provide sufficient accuracy for safe indoor navigation, especially in complex multi-floor buildings. |

| | | | | |
|---|---|---|---|---|
| 2 | Insufficient indoor mapping data | Technical | Likely - High potential impact | Lack of detailed, accurate indoor maps and floor plans for target buildings, making route calculations impossible for unreliable. |
| 3 | Safety hazards due to navigation errors | Safety | Unlikely - Critical impact | Incorrect navigation instructions could lead users into dangerous situations (stairs, obstacles, restricted areas) causing physical harm. |
| 4 | User interface complexity | Usability | Moderate - High potential impact | The voice-based interface may be too complex for blind users, reducing adoption and effectiveness of the applications. |
| 5 | Emergency response system failures | Safety | Unlikely - Critical impact | Failure of emergency calling or alert systems when users are lost or injured could result in delayed assistance and serious consequences. |
| 6 | Privacy and location data concerns | Legal/Ethical | Moderate - Medium potential impact | Handling sensitive location data of vulnerable users raises privacy concerns and potential regulatory compliance issues. |

## 3.4. Monitoring and controlling mechanisms

| **No.** | **Risk** | **Monitoring and Controlling** |
|---|---|---|
| 1 | Inadequate sensor accuracy and reliability | ● Conduct regular accuracy testing of sensor combination in various indoor environments<br>● Test with different smartphone models and OS versions<br>● Implement sensor fusion algorithms to improve reliability |
| 2 | Insufficient indoor mapping data | ● Establish partnership with building management<br>● Develop procedures for crowd-sourced mapping updates<br>● Create mapping validation processes |
| 3 | Safety hazards due to navigation errors | ● Implement multiple validation checks for navigation instructions<br>● Create comprehensive safety protocols and warnings<br>● Conduct safety testing in controlled environments before deployment |

| | | |
|---|---|---|
| 4 | User interface complexity | ● Conduct regular usability testing with blind users<br>● Implement user feedback collection mechanisms<br>● Create simple, voice command structures<br>● Provide comprehensive user training materials |
| 5 | Emergency response system failures | ● Implement redundant emergency communications systems<br>● Regular testing of emergency features<br>● Establish protocols with campus/building security |
| 6 | Privacy and location data concerns | ● Implement privacy by design principles<br>● Regular review of data collection and storage practices<br>● Ensure compliance with relevant privacy regulation<br>● Provide clear privacy policies and user consent mechanisms |

# 4. Technical Process Plans

## 4.1. Methods, tools, and techniques

1. Hardware, OS, and network environments
    a. Development hardware: Standard laptops with Windows and at least 16GB RAM and GPU acceleration for machine learning training.
    b. Android phone for testing
    c. Operating system:
        i. Development: Windows 11
        ii. Target platforms: As we mentioned in AS-5, we suppose all of the users have Android 7.0+
2. Software tools
    a. Requirements management: GitHub Issues & Trello boards for backlog and sprint planning.
    b. Design Modeling: UML diagrams using Draw.io
    c. Source Code Management: GitHub repository with a branching strategy.
    d. IDE/Compiler: VS Code for frontend (XML) and backend (Python) services.
    e. Document Management: LaTeX (Overleaf) for reports and Markdown stored in GitHub for lightweight docs.
3. Development Methodologies
    a. Requirements Practices
       -Iterative elicitation with AS-IS and TO-BE scenarios, validated by team role play as blind users.
    b. Design Methodology
       -MVC architecture for app structure.
       -Modular design for extensibility, such as new sensors or features.
    c. Programming Languages
       -Python, XML
    d. Coding Standards
       -W3C Extensible Markup Language (XML) 1.0
       -PEP8 for Python scripts.
    e. System Integration Procedure
       -Continuous integration on GitHub.
       -Automate unit and UI testing before merge.
       -Weekly integration builds for stakeholder review.
4. Quality Assurance Practices
    a. Peer Reviews
       -Weekly code reviews via GitHub merge requests.
    b. Unit Testing
       -Android Studio
    c. Debugging Tools
       -Android Studio
    d. Defect Tracking
       -GitHub Issue with severity labels.
    e. System Testing and Integration

-Test case for navigation flows, accessibility, such as voice commands, haptic feedback.

## 4.2. Software documentation

| Document | Template or Standard | Created By | Reviewed By | Target Date | Distribution |
|---|---|---|---|---|---|
| Software Requirements Specification (SRS) | IEEE 830 | Canvas | Team | Aug 18 | GitHub |
| Project Plan | Course template | Tong Shen, Zirun Ye | Zijian Zhang | Sep 14 | GitHub |
| User Manual | Course template | Zijian Zhang | Jinming Wang | Nov 11 | GitHub |
| Test Plan | IEEE 829 | Jinming Wang | Zirun Ye | Nov 23 | GitHub |
| Final Report | Course template | Team | Team | Dec 7 | Github |