Opsis - Seal T6

Team members: Aryana Khaffaf Sharif Zamin, Tee Nguyen, Rayeed Zarif, Samantha Sarabia, Nowsin Anzum Mozumder, Nathaniel Norman, Amal Saeed, Erika Barron, Corina Salazar

Delivered to Dr. Chung

4 September, 2025

1   Introduction

People with vision impairments should be able to navigate indoors either through a building or from one building to another. The most effective method this is traditionally done is with a guide dog, however guide dogs can cost anywhere from 50,000-60,000$ [1] or a long wait list. They also require you to take care of the dog on a day to day basis.

1.1   Project overview

A smartphone app designed to assist blind and visually impaired users by providing ways for them to safely navigate a room. The app could use the camera and AI to deliver a safe and smooth experience.

1.2   Project deliverables

- September 4 (Thursday) - Preliminary Project Plan
- October 2(Tuesday)/October 7 (Tuesday) - Interim Project I
- October 14 (Thursday) - Final Project I Submission
- November 13 (Thursday) - Interim Project II
- December 2 (Tuesday) / December 4 (Thursday) - Final Project II Submission

1.3   Evolution of this document

This document will continue to be updated as the project progresses through the semester. We will add any changes or new challenges we face and revise this document as needed for the changing requirements and the things we have come to learn.

1.4   References

[1] K. Hitchcock, "How Much is a Guide Dog in 2025? Cost Breakdown & FAQs | Hepper Pet

Resources," Hepper Pet Resources, Jan. 08, 2025.

https://articles.hepper.com/how-much-is-a-guide-dog/ (accessed Sep. 04, 2025).

1.5   Definitions, acronyms, and abbreviations

UI/UX: User Interface and User Experience
ML: Machine Learning
AI: Artificial Intelligence

2   Project organization

2.1   Process model

We will use an Agile Scrum model with two-week sprints. This process provides regular opportunities to review progress, test features, and incorporate feedback from testing. By working in short, structured cycles, the team can quickly identify issues, adapt to changing requirements, and steadily improve the app's functionality.

2.2   Organization structure

- Project Manager/ Coordinator: Aryana Khaffaf Sharif Zamin
- Documentation: Corina Salazar, Erika Barron
- UI/UX Design: Rayeed Zarif
- Frontend Development: Samantha Sarabia
- Backend Development: Nowsin Anzum Mozumder
- Database & Data Management: Nathaniel Norman
- Deployment & Integration: Kruthik Kalidas
- Accessibility & Research: Amal Saeed, Tee Nguyen

This structure allows each subsystem of the app to have clear ownership.

2.3   Organizational boundaries and interfaces

Subsystem Boundaries:
- UI/UX Design will create wireframes and user flows that guide Frontend Development.
- Frontend and Backend will connect through agreed-upon APIs.
- Backend will rely on the Database to store indoor maps, navigation details, and user preferences.
- Accessibility will collaborate closely with Design and Frontend to ensure the app works well with screen readers and haptic feedback.
- Testing/QA will evaluate all parts of the system to confirm features function correctly and meet accessibility standards.
- Deployment/ Integration will maintain GitHub branches, CI/CD pipelines, and testing environments to keep all subsystems aligned.

Management Boundaries:
- The Project Manager (Aryana) is responsible for coordinating efforts across teams, tracking deadlines, and handling conflicts.
- Primary communication tools will be Discord for quick updates and GitHub for version control and issue tracking.

Interfaces
- Weekly team meetings will keep everyone aligned.

- UI/UX designs will be passed to Frontend developers using Figma.
- Backend and Database teams will work with diagrams and API documentation.
- Accessibility requirements and guidelines will be shared with all team members to ensure consistent implementation.

2.4   Project responsibilities

Project Planning and Management:
- Define project scope, goals, timeline and assign team roles
- Manage resources, communication and progress tracking

Research and Requirement
- Conduct user research to identify needs and challenges
- Review accessibility standards and outline key features

Design
- Create accessible UI/UX with voice, audio and haptic feedback
- Build prototypes and test with visual impaired users

Development
- Implement core features like obstacle detection and navigation
- Integrate assistive tech like text-to-speech and voice commands

Testing and Quality Assurance
- Perform functional and accessibility testing with real users
- Fix bugs and optimize performance for usability

Deployment and Maintenance
- Launch the app on app stores and monitor performance
- Release updates and address user feedbacks

3   Managerial process

3.1   Management objectives and priorities

- To be able to deliver a reliable and useable app within the project deadline
- Make sure the important features that are necessary for the app to function are implemented first
- Make sure the app is accessible and a good quality
- Make sure the features are functionable and easy to use.

3.2   Assumptions, dependencies, and constraints

Assumptions:
- Users will have access to a phone with a functioning camera microphone and internet connection

- Blind and visually impaired users will be able to provide meaningful feedback during testing with the assistance of accessibility tools.
- Team members will be available to contribute throughout the semester and communicate effectively.

Dependencies:
- The project depends highly on certain features (accessabilities) of a phone such as screen readers, text-to-speech etc.
- The success of the AI/ML components relies on if the models are trained accurately with accurate datasets.
- Development is dependent on external libraries, necessary frameworks, and APIs that are available for this project.

Constraints:

- Insufficient project timeline (one semester) requires focusing on core features first.
- Limited resources such as time, members and necessary tools might hinder the process of making improvements on the project.
- Accessibility and usability must be prioritized, making less efforts allocated for design choices.

3.3   Risk management

Technical Integration Risks
- Risk: Difficulties integrating the frontend with backend APIs and the database.
- Mitigation: Define clear API contracts early and conduct integration testing after each sprint.

Accessibility Risk
- Risk: Features may not fully meet accessibility standard for visually impaired users
- Mitigation: Invade accessibility lead (Amal) in all design reviews and test regularly with accessibility tools.

Scheduling Risk
- Risk: Team members may miss deadlines due to overlapping coursework of personal obligations
- Mitigation: Use agile sprints to break down tasks into smaller milestones and track progress through GitHub issues and weekly meetings.

Communications Risk
- Risk: Miscommunications between subsystem teams may cause delays
- Mitigation: Hold weekly check-ins, document, and delegate work when tasks overlap

Tool Risk
- Risk: Problems with tools, CI/CD pipelines, or other tools could delay development
- Mitigation: Keep backups of repositories, assign deployment lead, and prepare alternative collaboration methods if a tool fails.

3.4   Monitoring and controlling mechanisms

- Review the work done every 2 weeks (sprint reviews)
- Track everyone's status via the chosen communication platform
- Have meeting in person every thursday at 5:15pm to be able to keep track of everyone and ask for assistance

4. Technical process

4.1   Methods, tools, and techniques

- Prototype: Figma for the mockup
- Version Control: Github
- Communication: Email and Discord

4.2   Software documentation

- WRS
- User manual for the prototype
- Project plan document
- Design document

4.3   Project support functions

- Quality assurance through peer review and testing
- Maintaining communication through consistent updates through email or discord
- Keeping proper documentation

5. Work Elements, Schedule, and Budget

5.1 Work Elements

- Information Gathering:
    - Researching existing accessibility apps and tools.
    - Collecting system and user needs.
- Analysis and Documentation:
    - Defining functional requirements
    - Defining non-functional requirements
    - Use cases
- Implementation:

- ○ Coding accessibility features
- ● Testing:
  - ○ Accessibility compliance testing
- ● Deployment
  - ○ Updating and maintaining accessibility features

5.2 Schedule

Deliverable	Due date	Team Leader	Tools
PPP	September 4th, 2025	Aryana Khaffaf Sharif Zamin	Google Docs
Phase I: Interim	October 7th, 2025	Nathaniel Gregory Norman Plaud	Google Slides, Google Docs. Figma
PhaseI: Final	October 14th, 2025	TBD	TBD
Phase II: Interim	November 13th, 2025	TBD	TBD
Phase II: Final	December 4th, 2025	TBD	TBD

5.3 Budget

- ● All tools used thus far have been free.