WRS Evolution Document

Opsis

Seal T6

Team members: Aryana Khaffaf Sharif Zamin, Tee Nguyen, Rayeed Zarif, Samantha Sarabia,

Nowsin Anzum Mozumder, Nathaniel Norman, Kruthik Kalidas, Amal Saeed, Erika Barron.

Team URL: https://sgs240001.wixsite.com/theia-spmp
Date: 09/28/2025
Course/Instructor : SE 4351.001 - Requirements Engineering - Lawrence Chung

Revision History

Version	Date	Author(s)	Description / Changes Made
0.1	09/24/2025	Aryana Khaffaf Sharif Zamin	Initial draft
0.1	09/28/2025	Nowsin Anzum Mozumder	Documentation
0.2	9/30/2025	Aryana Khaffaf Sharif Zamin	Information transferred from powerpoint
0.3	10/02/2025	Tee Nguyen	Team & project name changed
0.4	10/03/2025	Corina Salazar	Figma finalzied
1.0	10/05/2025	Nathaniel Norman	Final submission meeting & edits

Team Members & Roles:

- Aryana Khaffaf Sharif Zamin – Project Manager/Coordinator/Documentation
- Nathaniel Norman – Coordinator/Documentation
- Erika Barron – Documentation
- Samantha Sarabia – Frontend Development
- Nowsin Anzum Mozumder – Backend Development
- Tee Nguyen – Backend Development
- Corina Salazar - UI/UX Designer
- Amal Saeed – Accessibility & Research
- Rayeed Zarif – Accessibility & Research

Team Leader(s): Aryana Khaffaf Sharif Zamin
Meetings (frequency, medium, etc.): Weekly team meetings via Discord + in-person check-ins
Inputs:

Preliminary Project Plan

Accessibility standards (WCAG, screen reader guidelines)

User research feedback

Market research on existing assistive apps

Activities:

- Requirements elicitation & analysis
- Functional/Non-functional specification
- Wireframing and early mockups
- Testing with prototypes & accessibility tools

Outputs:

- Finalized WRS document
- Prototype mockups
- Traceability matrix

Resources Used:

- Figma (design)
- GitHub (version control)
- Google Docs (documentation)
- Discord (communication)
- Wix (Team Website)
- VS Code (Programming the application)
- SupaBase (Database)

1. Introduction

A smartphone software called Opsis was created to help blind and VI users navigate indoor environments. The developing baseline requirements (WRS) for additional specification, design, and testing are provided in this document.

2. Issues with Preliminary Definition Given

2.1 Issues with II.1 The Domain, Stakeholders, Functional and Non-Functional Objectives

2.1.1: Limited initial stakeholder feedback.

- Description: Direct feedback from visually impaired users is lacking in the preliminary criteria.
- Choices: Conduct user interviews or just use the literature review.
- Conclusion & Justification: Interview users to confirm usability hypotheses.

2.1.2: Underdefined scope for "indoor navigation."

- Description: Indoor navigation may refer to shopping centers, workplaces, academic institutions, or residences.
- Choices: Focus on a single kind of building or remain general.
- Decision and Justification: Before broadening the scope, begin with basic interior navigation (rooms, classrooms, hallways).

2.2 Issues with II.2 Software System Requirements: Functional Requirements

2.2.1 : Obstacle detection needs clarification.

- Description: Which sensors and inputs—camera, LiDAR, and accelerometer—will be utilized?
- Choices: external sensors versus camera-based machine learning.
- Decision & Justification: Make use of AI and smartphone cameras (scalable and affordable).

2.2.2 : Route guidance scope unclear.

- Description: Should the app only provide obstacle notifications or provide turn-by-turn navigation?
- Options: include restricted obstacle feedback versus turn-by-turn voice navigation.
- Decision and Justification: For independence, turn-by-turn is ideal, but begin MVP with obstacle + supervision.

2.3 Issues with II.3 Software System Non-Functional Requirements

2.3.1 Issue-A (Latency)

- Description: Our first draft did not specify how fast the app should respond to user commands. For navigation, slow or fast responses could make the app unsafe.
- Options: (a) keep it general, (b) define a measurable standard.
- Decision & Rationale: Define a standard: voice response should occur within 1-2 seconds. This allows for the requirement to be tested and see if the latency is within a reasonable time frame.

2.3.2 Issue-B (Device Compatibility)

- Description: The preliminary definition did not clarify if the app will work on both iOS and Android.
- Options: (a) start with one and eventually build the other, (b) build both simultaneously
- Decision & Rationale: We will focus on building one (Android) since it is more flexible for prototyping.

2.3.3 Security and privacy

- Description: The draft did not address how user data will be protected. Even in a class project, ignoring privacy could weaken credibility.
- Options: (a) Skip for now, (b) Add a simple privacy requirement.
- Decision & Justification: We will include a requirement that no sensitive data (like navigation history or audio recordings) is stored unnecessarily.

3. WRS

3.1.0 Stakeholders

- Primary users: Blind and visually impaired (VI) individuals who need assistance with independent indoor navigation.
- Secondary stakeholders: Caregivers, accessibility coordinators, and advocacy groups who may help test or provide feedback.
- Project stakeholders: Developers (Seal T6), course instructor, and potential future collaborators (e.g., app testers).

3.1.1 Problem

Indoor navigation is often unsafe or stressful for blind and VI individuals. Current solutions, such as guide dogs or canes, have limitations:

- Guide dogs are expensive ($50,000–$60,000) and require long training and care.
- Canes help with detecting nearby objects but cannot provide route guidance.
- Human help reduces independence and may not always be available.

As a result, blind and VI users face barriers to moving safely and independently indoors.

3.1.2 Goal

The goal of Opsis is to make independent indoor navigation possible for blind and visually impaired individuals through a reasonably priced smartphone app. By using a device that most users already own, the app avoids the high costs of alternatives like guide dogs while offering more functionality than a cane. Opsis aims to provide real-time obstacle detection, voice and/or haptic feedback, and simple navigation assistance in common environments such as classrooms and hallways.

3.1.3 Improved understanding of II.1

Through refining our requirements, we now recognize that a smartphone-based AI solution provides the best balance between cost, accessibility, and practicality. Since most blind and visually impaired individuals already own smartphones, our approach avoids the high expenses of alternatives such as guide dogs or specialized equipment. By leveraging built-in tools like the camera, microphone, and accessibility settings, the system can deliver navigation support without requiring extra hardware.

3.2 R (Requirements)

3.2.1 Functional Requirements (FRs)

FR1: Help blind people navigate indoors.

FR2 : Ask the user for the current location to calculate the most convenient route

to the destination.

FR3: Alert user of any obstacles along route.

FR4: Actively calculate alternative routes in case of obstacles.

FR5: Guide users around obstacles or crowds of people.

3.2.2 Non-Functional Requirements (NFRs)

NFR1: The application shall be user-friendly, providing only simple and precise commands (Usability).

NFR2: The application shall be easy to test across multiple scenarios that simulate real-world usage (Testability).

NFR3: The application shall support voice commands and provide voice-based responses to enhance accessibility (Accessibility).

NFR4: The application shall remain responsive and functional under all conditions, and it shall not crash due to invalid or erroneous inputs (Operability).

NFR5: The application shall be designed with strong resistance to malware and other threats that could compromise its functionality.

3.3 S (Specification)

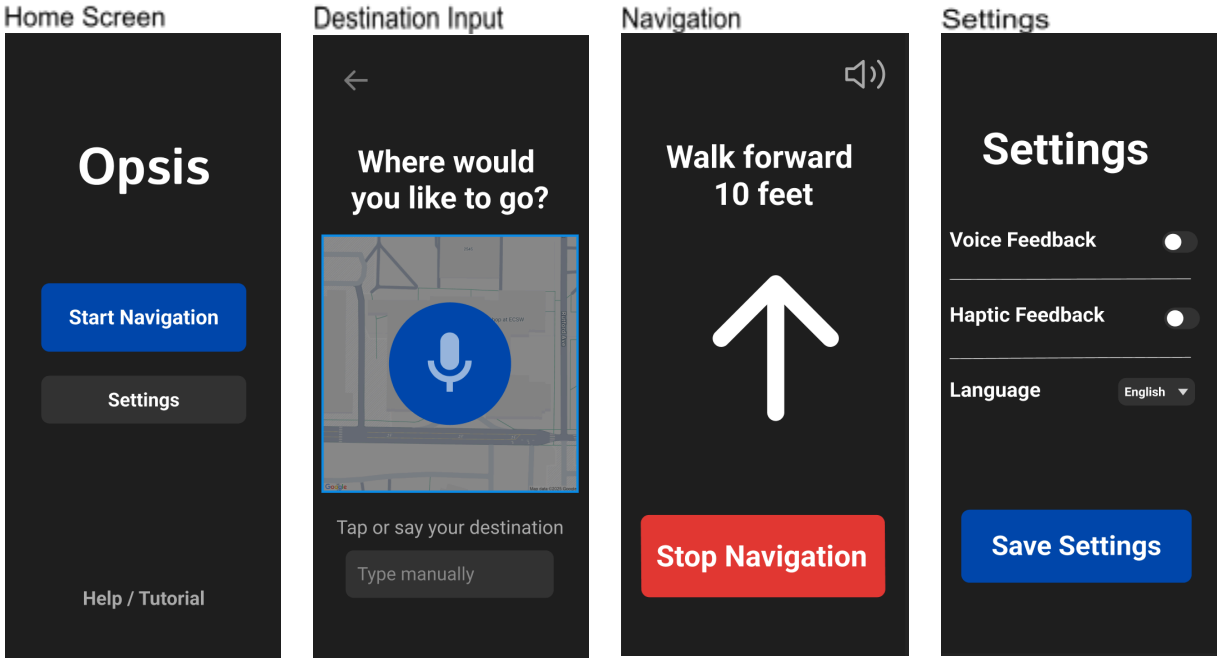
3.3.1 Functional Specification

- Users can indicate their destination using a voice input mechanism.
- Integration of a routing algorithm with an indoor map.
- AI and phone cameras are used to detect obstacles in real time.
- The rerouting function is continuous.
- Voice prompts that provide step-by-step instructions.

3.3.2 Non-Functional Specification

- UI restricted to necessary controls (high contrast, large buttons).
- Latency for voice interactions ≤ 1 second.
- The system was evaluated for misclicks and misvoice recognition.
- Data storage is governed by security protocols.

4. Preliminary (Mockup) Prototype and User Manual



5. Traceability

Functional Requirements Traceability:

- FR1 (Help blind people navigate indoors) maps to the core navigation module integrating routing algorithms and indoor maps.
- FR2 (Ask for user location) is traced to voice input and location detection submodules.
- FR3 (Alert user of obstacles) links to AI-powered obstacle detection via camera feed.
- FR4 (Active rerouting) is connected to real-time route recalculation logic.

- FR5 (Guide users around obstacles or crowds) ties to voice guidance and alternative route suggestions.

Non-Functional Requirements Traceability:

- NFR1 (Usability) relates to UI/UX design decisions: simple, clear voice commands and accessible layouts.
- NFR2 (Testability) corresponds to test plans simulating real indoor navigation scenarios.
- NFR3 (Accessibility) traces to voice command support, screen reader compatibility, and haptic feedback in design and implementation.
- NFR4 (Operability) ties to robustness testing, error handling, and performance benchmarks.
- NFR5 (Security) links to data privacy designs, secure storage, and malware resistance testing.

References

1. Requirements Engineering: From System Goals to UML Models to Software Specifications, Axel van Lamsweerde, John Wiley Sons

2. Managing Software Requirements: A Use Case Approach, 2nd edition, Dean Leffingwell, Don Widrig, Addison Wesley: Boston

3. Non-Functional Requirements in Software Engineering, L. Chung, B. Nixon, E. Yu and J. Mylopoulos, Kluwer Academic Publishing, 2000 (An earlier version of a framework book chapter)

4. System Requirements Engineering, P. Loucopoulos and V. Karakostas, McGraw-Hill: [all.rar] [Preface; Chap1; Chap2; Chap3; Chap5; Chap6] thanks to Varshada!

5. Mastering the Requirements Process: Getting Requirements Right (3rd Edition) by Suzanne Robertson & James Robertson ISBN-13: 978-0321815743 or ISBN-10: 0321815742

6. Software Requirements: Objects, Functions, & States, A. M. Davis, Prentice Hall: Englewood Cliffs, 1993.

7. System and Software Requirements Engineering: Tutorial, R. H. Thayer and M. Dortman (Editors),IEEE Computer Society Press

8. Problem Frames: Analyzing and Structuring Software Development Problems, M. Jackson, Addison-Wesley Professional; 1st edition (December 15, 2000)

9. Requirements Engineering: Processes and Techniques, G. Kotonya and I. Sommerville, John Wiley Sons

10. Requirements Engineering - A Good Practice Guide, I. Sommerville and P. Sawyer, Wiley

11. Scenarios, Stories, Use Cases Through the Systems Development Life-Cycle, I. Alexander and N. Maiden (eds.), John Wiley & Sons, 2004.

12. Requirements Engineerng: Frameworks for Understanding, R. Wieringa, Wiley, 1997

13. Requirements Engineering: Social and Technical Issues, J. Goguen, and M. Jirotka (Eds.), Academic Press, 1994.

14. Requirements Engineering, L. Macaulay, Springer Verlag, 1996

15. User-Centered Requirements Analysis, C. F. Martin, Prentice-Hall, 1994

16. Information System Requirements: Determination and Analysis, D. Flynn, McGraw-Hill, 1992

17. Are Your Lights On?: How to Figure Out What the Problem Really Is, D. C. Gause and G. M. Weinberg, Dorset House, 1990.

18. Exploring Requirements, D. Gause and G. Weinberg, Dorset House, 1989

19. Managing Systems Requirements: Methods, Tools, and Cases, S. J. Andriole, McGraw-Hill, 1996.

20. System Requirements Analysis, J. O. Grady, McGraw Hill, 1993.

21. Systems Engineering: Coping with Complexity, R. Stevens, K. Jackson, P. Brook, and S. Arnold, Prentice Hall 1998.

22. Requirements Engineering and Rapid Development: A Rigorous, Object-Oriented Approach, I. S. Graham, Addison-Wesley, 1998.

23. Practical Software Requirements; A Manual Of Content And Style, B. L. Kovitz, Manning Publications, 1998

24. User-Centered Requirements: The Scenario-Based Engineering Process, K. L. McGraw and K. Harbison, Lawrence Erlbaum Associates, 1997.

25. The Complete Systems Analysis, J. Robertson and S. Robertson, Dorset House, 1998.

26. Applying Use Cases: A Practical Guide, G. Schneider and J. P. Winters, Addison-Wesley, 1998.

27. Object-Oriented Analysis and Design, with Applications, G. Booch, Benjamin-Cummings, 1994

28. Object-Oriented Methods: A Foundation, J. Martin and J. Odell, Prentice-Hall, 1995

29. The Unified Modeling Language Reference Manual, J. Rumbaugh, I. Jacobson and G. Booch, Addison-Wesley, 1998

30. The Unified Modeling Language User Manual, G. Booch, J. Rumbaugh and I. Jacobson, Addison-Wesley, 1998.

Appendix A : Glossary

- Accessibility: Creation of products that are usable by individuals with impairments, including those who are blind or visually impaired.
- AI (Artificial Intelligence): Technology that makes it possible for machines to carry out operations like obstacle detection that normally demand for human intellect.
- Backend Development: Programming that enables frontend features with an emphasis on server, database, and application logic.
- Functional Requirements (FRs): Certain actions or tasks that the system must carry out (such as route calculation and barrier detection).
- Haptic Feedback: A tactile response technology (such as vibrations to warn consumers) that communicates through touch.
- Indoor Navigation: The technique of guiding people through enclosed areas or buildings.
- Non-Functional Requirements (NFRs): The system needs to have quality features including security, usability, and accessibility.
- Prototype: An early sample or model to test concepts and gather user feedback.
- Routing Algorithm: A method for determining the best path from one location to another.
- Traceability: The capacity to connect needs at every level of development, from testing to specification.
- UI/UX Design: The design of the user interface and user experience emphasizes how people engage with and perceive the program.
- Voice Commands: Users' spoken inputs are used to operate the application and engage with its features.
- WRS (Work Requirements Specification): The evolving document detailing the requirements for the project.

Index

Term	**Section/Reference**
Accessibility	2.1, 3.2.2, 3.3.2, Appendix A
AI (Artificial Intelligence)	1, 3.3.1, Appendix A
Backend Development	Team Members & Roles
Functional Requirements	2.2, 3.2.1
Haptic Feedback	3.1.2, 3.3.1, Appendix A
Indoor Navigation	1, 3.1.1
Non-Functional Requirements	2.3, 3.2.2
Prototype	4, Appendix A
Routing Algorithm	3.3.1
Traceability	5

UI/UX Design	Team Members & Roles, 3.2.1
Voice Commands	3.2.2, 3.3.1
WRS (Work Requirements Specification)	Entire document

Additional Notes
- Why our team's work is the best (justification):

- Seal T6 is committed to rigorous stakeholder engagement, especially focusing on direct user feedback from the visually impaired community, ensuring the solution is relevant and impactful.
- We emphasize accessibility and usability from the start, integrating input from accessibility experts within the team.
- Agile methodology with regular sprint reviews allows for rapid adaptation and progressive refinement.
- Seal T6 leverages diverse skills—UI/UX design, backend and frontend development, deployment, and accessibility research—ensuring comprehensive coverage of all critical aspects of the project.
- Strong documentation and communication practices (use of GitHub, Discord, Figma) maintain high quality and clear coordination.