WRS Document

Team 1

Members:
Kashir Khan
Sam Heidler
Fayez Yusuf
Padraig Flanary

Revision History

Version	Date	Comments	Author(s)
1.0	3/18/2024	Finsihed Intro + Section 2	Kashir Khan
2.0 (Final)	3/27/2024	Finished Rest of the document	Fayez Yusuf and Padraig Flanary

Table of Contents

Process

For the initial phase of our Vision Share project, the team was divided into two groups: one focused on creating the application's mockup and the other on refining our requirements. Starting with a compilation of preliminary requirements based on the project description, we collaboratively identified and addressed issues within those requirements. The team explored various solutions, weighing their pros and cons before making decisions through a democratic voting process, ensuring that each choice was well-justified and aligned with our project goals.

1 Introduction

1.1 Purpose

This document captures the decision-making process and the outcomes from our team's initial efforts to refine the Preliminary Project Plan for the development of the Vision Share app. It details encountered challenges, explored options for resolution, final decisions made, and the reasoning behind those decisions. This document is intended for the Vision Share project owners, allowing them to evaluate our team's capability in specifying project requirements, offer feedback on decisions made, and assess our ability to execute the project successfully.

1.2 Scope

Vision Share is envisioned as an application to facilitate indoor navigation for the visually impaired through real-time video assistance from volunteers. Targeting the widespread smartphone user base, this app aims to bridge the gap in current navigation aids by leveraging the convenience and familiarity of smartphones. It enables users to receive guidance without highlighting their impairments, offering a more discreet and integrated solution than traditional aids.

1.3 Overview

The introduction outlines the document's purpose, scope, and key terms. The following sections will address the identified issues within the PRD, propose solutions, detail the enhanced understanding of the project requirements, describe the prototype development, and establish a traceability matrix for requirements. The document concludes with a reference list utilized in its preparation.

2.1 Issues with Preliminary Definition Given (ambiguities, incompleteness, inconsistency, conflicts, etc.)

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

2.1.1 Inclusive Definition of Indoor Navigation Areas

I. Description: The domain of "indoors" is broad and includes various environments such as classrooms, offices, washrooms, lounges, and elevators. It is crucial to precisely define which indoor areas will be prioritized for navigation support by the app, ensuring the app's navigation system is tailored to the environments most relevant to the primary stakeholders.

II. Options:
   A. Prioritize public spaces within buildings, like hallways, main lobbies, and elevators, over private areas such as specific offices or classrooms.
   B. Include all indoor areas without prioritization, aiming for comprehensive coverage.
   C. Focus on areas where navigation support is typically most needed, based on user feedback or pilot studies.

III. Decision and Rationale: Option C was chosen, focusing on areas identified as high-priority through feedback from blind individuals and caretakers. This approach ensures that development efforts are concentrated on areas that will provide the most immediate and tangible benefits to users, aligning with the functional objective of safe, fast, and comfortable navigation.

2.1.2 Stakeholder Involvement and Expectations

I. Description: The primary stakeholder is identified as a blind person requiring indoor navigation assistance, with secondary stakeholders including caretakers, accessibility departments, and police. Clarifying the extent of involvement and expectations from each stakeholder group is necessary to ensure the app meets diverse needs effectively.

II. Options:
   A. Conduct detailed interviews and surveys with stakeholders to understand their specific needs and expectations.
   B. Assume primary focus on the needs of the blind person, with secondary stakeholders providing support as needed.
   C. Develop a flexible app configuration to cater to the varying needs of all stakeholders.

III. Decision and Rationale: Option A and C were selected, emphasizing the importance of stakeholder engagement in the development process. This ensures the app is user-centric while being adaptable to support the roles of secondary stakeholders, thus meeting the non-functional objective of usability and comfort.

2.1.3 Ensuring Effective Use of Smartphone Sensors

I. Description: The project description mentions leveraging as many smartphone sensors as possible to aid navigation. However, it's vital to determine which sensors are most effective for indoor navigation and how they can be used without overwhelming the user, especially considering the primary users are blind.

II. Options:
   A. Focus on the most commonly available sensors (e.g., accelerometer, gyroscope) for basic navigation.
   B. Incorporate advanced sensors (e.g., LiDAR in newer smartphones) for enhanced spatial awareness, while ensuring fallbacks for devices without such capabilities.
   C. Develop a modular sensor utilization strategy, allowing the app to adapt to the available sensors on the user's device.

III. Decision and Rationale: Option B and C were chosen to ensure the app utilizes available technology to its fullest potential while remaining accessible to users with varying device capabilities. This strategy supports the functional objective of safe and effective navigation and the non-functional objective of broad usability.

2.1.4 Defining "Safe Navigation" in Varied Indoor Environments

I. Description: While "safe navigation" is identified as a non-functional objective, its meaning could vary significantly across different indoor environments (e.g., navigating around obstacles in a crowded lobby versus finding a safe path through a quiet library). The lack of a clear, universally applicable definition of "safe navigation" introduces ambiguity.

II. Options:
   A. Develop a generic safety protocol applicable in most indoor environments.
   B. Customize safety features for specific types of environments based on potential risks and obstacles unique to those settings.
   C. Engage with primary users and safety experts to create a comprehensive safety framework that adapts to the user's current environment.

III. Decision and Rationale: Option C was selected, recognizing the complexity and variability of "safe navigation". This approach ensures that safety measures are robust, contextually relevant, and adaptable, directly addressing the ambiguities in the original objective.

2.1.5 Consistency in User Interface (UI) for Blind Users

I. Description: The objective mentions the importance of usability, particularly through voice recognition, for blind users. However, there may be inconsistency in how UI elements are presented or interacted with across different sections of the app, potentially confusing users.

II. Options:
   A. Standardize voice commands and feedback across all app functionalities.

B. Allow for customizable UI settings to accommodate individual user preferences.
C. Implement a guided tutorial specifically designed for blind users to familiarize them with the app's interface.

III. Decision and Rationale: Option A and C were chosen to ensure consistency and ease of use across the app. Standardization minimizes the learning curve and potential confusion, while a guided tutorial provides users with the tools they need to navigate the app confidently.

2.1.6 Integration with External Emergency Services

I. Description: The involvement of police and emergency services as secondary stakeholders suggests an integration with external emergency response systems. However, the document does not specify how this integration will be achieved, leading to incompleteness in stakeholder responsibilities and system functionality.

II. Options:
   A. Develop an API for emergency services, allowing for direct app-to-service communication in case of an emergency.
   B. Use existing emergency call functionalities on smartphones, supplemented with app-specific location and context information.
   C. Collaborate with emergency services to create a protocol for receiving and responding to alerts from the app.

III. Decision and Rationale: Options B and C were selected to leverage existing infrastructure while ensuring that emergency responses are informed and efficient. This dual approach addresses incompleteness in the system's design and strengthens stakeholder collaboration.

2.1.7 Conflict Between Fast and Comfortable Navigation

I. Description: The objectives of fast navigation and comfortable navigation could conflict, particularly in complex indoor environments. Fast navigation might prioritize speed over ease, potentially leading to user discomfort or safety concerns.

II. Options:
   A. Prioritize safety and comfort over speed, ensuring navigation paths are user-friendly even if slightly longer.
   B. Offer navigation options, allowing users to choose between the fastest route and the most comfortable route.
   C. Use adaptive algorithms that balance speed and comfort based on user feedback and context.

III. Decision and Rationale: Options B and C were chosen, acknowledging the potential conflict and providing users with the ability to customize their navigation experience. This flexibility ensures that the app can cater to diverse user preferences and situations, aligning with both functional and non-functional objectives.

2.2 Issues with Software System Requirements: Functional Requirements

2.2.1 Ambiguity in Destination Suggestions Based on User Routine

I. Description: The requirement to suggest or confirm possible destination locations utilizing the user's routine schedule or habit lacks clarity on how the system will learn and adapt to the user's routines. Without explicit mechanisms for learning or inputting routines, there's ambiguity in implementation.

II. Options:
   A. Implement a learning algorithm that adapts to the user's habits over time based on historical location data.
   B. Allow users to manually input and update their routine destinations and preferences within the app.
   C. Combine both automated learning and manual input for a hybrid approach to understanding user routines.

III. Decision and Rationale: Option C was chosen to provide flexibility and accuracy in suggesting destinations. This hybrid approach addresses the ambiguity by ensuring the system can adapt to user habits while allowing for user control and verification of the suggested destinations.

2.2.2 Incompleteness in Handling Varied Obstacle Types

I. Description: Detecting obstacles and instructing the user on avoidance is critical. However, the requirement does not specify the range of obstacles that the system can detect (e.g., static vs. dynamic obstacles) or how it will communicate avoidance strategies, leading to incompleteness in functionality.

II. Options:
   A. Focus on detecting static obstacles only, such as walls and doors.
   B. Integrate dynamic obstacle detection, such as people and pets, using more sophisticated sensors or data inputs.
   C. Offer customizable obstacle detection settings, allowing users to set preferences based on their environment.

III. Decision and Rationale: Option C was selected to ensure comprehensive support for users, addressing incompleteness by committing to both static and dynamic obstacle detection. This decision is based on the principle of maximizing safety and usability for blind individuals in various environments.

2.2.3 Inconsistency in Emergency Communication Features

I. Description: The requirements mention placing emergency calls and messages after detecting a fall or when lost, but there's inconsistency in how these features integrate with different emergency services and user contacts. The protocol for determining when and whom to contact is unclear.

II. Options:
   A. Standardize emergency contacts, allowing the user to pre-define whom the app contacts in different scenarios.
   B. Integrate with local emergency services based on the user's location, providing automatic contact in critical situations.
   C. Develop a user-guided setup process for defining emergency protocols, including a hierarchy of contacts and conditions for contact.

III. Decision and Rationale: Option C was chosen to provide a tailored and effective emergency response system. This approach mitigates inconsistency by empowering users to specify their emergency contact preferences, ensuring the app acts in their best interest during critical moments.

2.2.4 Conflict Between User Privacy and Location-Based Features

I. Description: Several features depend on tracking and analyzing the user's location and routines. This raises potential conflicts between providing personalized services and respecting user privacy, especially concerning data storage and sharing.

II. Options:
   A. Limit data collection to only what is necessary for app functionality, with clear user consent.
   B. Implement robust encryption and privacy safeguards for stored data, ensuring it is only used for intended functionalities.
   C. Offer users the option to opt out of location-based features if they are concerned about privacy.

III. Decision and Rationale: Options B and C were selected, balancing the need for personalized, location-based services with the imperative of protecting user privacy. These measures address potential conflicts by ensuring user data is handled transparently and securely, with respect for user preferences.

2.2.5 Unclear User Feedback Mechanism for Route Preferences

I. Description: While the app is expected to offer route options and accept user preferences, the method by which user feedback is collected and utilized to refine future suggestions is not detailed, leading to ambiguity in the user interaction model.

II. Options:
   A. Implement voice feedback mechanisms allowing users to easily communicate their route preferences and experiences.
   B. Introduce a haptic feedback system for users to select and confirm route choices without relying solely on audio.
   C. Develop a machine learning model that automatically adjusts route suggestions based on implicit user feedback, such as route choices and travel times.

III. Decision and Rationale: Options A and C were chosen to ensure that the system can collect user preferences in an accessible manner and use this information to improve route suggestions. This approach addresses ambiguity by clearly defining mechanisms for feedback and adaptation.

2.2.6 Incomplete Specification for "Next Actions" Suggestions

I. Description: The app aims to suggest next actions based on the user's schedule or habits, but there is an incompleteness in how these suggestions are generated, prioritized, and presented to the user, especially in a non-visual context.

II. Options:
   A. Use an algorithm to analyze the user's calendar and habitual data to generate suggestions with clear, prioritized voice prompts.
   B. Allow users to set preferences for how and when suggestions are made, giving them control over the process.
   C. Incorporate an option for users to request suggestions on-demand, rather than receiving unsolicited prompts.

III. Decision and Rationale: Options A and B were selected to provide a balanced approach to action suggestions. This solution addresses incompleteness by outlining a specific method for generating and communicating suggestions, while also offering user control over the process.

2.2.7 Inconsistency in Navigation Instructions for Complex Environments

I. Description: Providing navigation instructions such as walking distances and turning points may become inconsistent in complex indoor environments with variable acoustics, crowdedness, or layout changes, potentially confusing users.

II. Options:
   A. Develop context-aware algorithms that adjust navigation instructions based on environmental factors.
   B. Introduce redundancy in navigation cues, using both auditory and haptic feedback to ensure clarity.
   C. Partner with venue managers to keep updated layouts and environmental data in the app database for accurate navigation assistance.

III. Decision and Rationale: Options A and B were chosen to enhance the reliability of navigation instructions. This decision addresses potential inconsistency by ensuring that instructions are adaptable and clear, even in challenging environments.

2.2.8 Conflicts Between Automatic Emergency Alerts and User Autonomy

I. Description: Automatically placing emergency calls or messages in certain situations (e.g., after detecting a fall) might conflict with the user's desire for autonomy and control over when and how to seek help.

II. Options:

    A. Allow users to configure the sensitivity and conditions under which automatic emergency alerts are triggered.
    B. Introduce a confirmation step where the user must validate the emergency alert before it is sent.
    C. Provide users with a quick way to cancel or override automatic alerts before they are transmitted.

III. Decision and Rationale: Options A and C were selected to balance safety features with user autonomy. By allowing users to customize and control emergency alert settings, the system respects user preferences while ensuring safety protocols are in place.

2.3 Issues with Software System Non-Functional Requirements

2.3.1 Trade-off Between Fastest and Most Comfortable Routes

I. Description: The requirements for the system to lead the user through both the fastest and the most comfortable routes might create conflicts, as the fastest route may not always be the most comfortable or safe for the user.

II. Options:
    A. Prioritize safety and comfort over speed by default, offering the fastest route as an alternative option when requested by the user.
    B. Develop a user profile system that learns from user feedback to understand and prioritize individual preferences for speed vs. comfort.
    C. Offer real-time options allowing users to choose between the fastest and most comfortable routes based on current circumstances.

III. Decision and Rationale: Option C was chosen to provide flexibility and control to the user, addressing the conflict by allowing users to make informed decisions based on their immediate preferences and the context of their journey.

2.3.2 Balancing Ubiquity with User Privacy and Data Security

I. Description: Achieving ubiquity implies constant access and functionality across various environments and devices, potentially raising concerns about user privacy and data security.

II. Options:
    A. Implement strict data encryption and anonymization techniques to protect user data while ensuring system availability.
    B. Develop a clear privacy policy detailing data usage, storage, and sharing practices, ensuring users are informed and can consent.
    C. Enable offline functionality for the app, reducing reliance on cloud services and enhancing privacy.

III. Decision and Rationale: Options A and B were selected to ensure that ubiquity does not compromise user privacy or security. These measures address concerns by establishing a robust framework for data protection while maintaining system accessibility.

2.3.3 Customization vs. Complexity for Users

I. Description: While customization is crucial for accessibility and user satisfaction, extensive customization options might overwhelm users, particularly if they are not tech-savvy, leading to underutilization of available features.

II. Options:
   A. Design an intuitive and guided customization process, simplifying the experience for users.
   B. Offer preset customization profiles that cater to common user preferences and needs.
   C. Provide accessible tutorials and support resources to assist users in personalizing the app according to their preferences.

III. Decision and Rationale: Options B and C were chosen to strike a balance between customization and usability. Preset profiles simplify initial setup, while tutorials and support resources empower users to explore and utilize advanced customization options as they become more comfortable with the app.

2.3.4 Ensuring Usability Across Diverse User Needs and Environments

I. Description: The requirement for the system to be usable encompasses a broad range of user needs and operating environments, which may lead to ambiguity in defining and implementing universal design principles.

II. Options:
   A. Conduct extensive user testing with a diverse group of users to identify and address usability challenges.
   B. Adopt a modular design approach, allowing for easy adaptation and customization to meet varied user needs.
   C. Collaborate with accessibility experts and organizations to ensure the app's design meets widely recognized accessibility standards.

III. Decision and Rationale: Options A and C were selected to ensure comprehensive usability. User testing provides direct feedback on design effectiveness, while collaboration with experts ensures that the app adheres to established accessibility guidelines, addressing ambiguities in usability requirements.

2.3.5 Extensibility for Future Technologies and User Needs

I. Description: While the system aims to be easily extensible, there is a lack of specificity regarding how new features, sensors, and hardware will be integrated, leading to potential incompleteness in future-proofing the system.

II. Options:

A. Implement a plugin architecture that allows for the easy integration of new technologies and features without overhauling the core system.
B. Establish partnerships with technology providers and developers to ensure timely updates and compatibility with new hardware and sensors.
C. Create a dedicated development fund or resource pool to support ongoing research and development for future extensions.

III. Decision and Rationale: Options A and B were chosen to address incompleteness in planning for extensibility. A plugin architecture offers flexibility for future enhancements, while partnerships ensure the system remains compatible with evolving technology standards.

2.3.6 Achieving System Ubiquity Across Different Hardware Platforms

I. Description: The requirement for the system to be ubiquitous suggests it should work seamlessly across various devices and platforms. However, there's ambiguity in how the app will maintain performance and accessibility standards on diverse hardware, especially older or less capable devices.

II. Options:
A. Develop a lightweight core application with additional features as optional downloads to accommodate different device capabilities.
B. Use responsive design and testing on a wide range of devices to ensure compatibility and performance.
C. Partner with device manufacturers for optimized app versions for specific hardware.

III. Decision and Rationale: Options A and B were selected to ensure the app's ubiquity without sacrificing performance. A lightweight core ensures basic functionality on all devices, while responsive design principles allow for optimal usability across a range of hardware specifications.

2.3.7 Conflicting Requirements for Fastest Route and System Usability

I. Description: Prioritizing the fastest route may conflict with the NFR of system usability, especially for users who prefer more straightforward or safer paths over the quickest one. This can lead to confusion or dissatisfaction if users are routinely directed along routes they find uncomfortable or difficult to navigate.

II. Options:
A. Integrate a preference setting allowing users to prioritize route comfort or simplicity over speed.
B. Offer route options with clear descriptions of each route's characteristics, letting users make informed choices.
C. Implement adaptive routing that considers user feedback and route ratings over time to personalize route suggestions.

III. Decision and Rationale: Options B and C were chosen to address this conflict by providing users with choice and control over their navigation experience. Descriptive options empower users to make informed decisions, while adaptive routing ensures the system learns from user preferences for more personalized guidance.

2.3.8 Customization Complexity vs. Extensibility

I. Description: While the system aims to be customizable and easily extensible, there's a potential conflict between the complexity of implementing extensive customization options and maintaining a framework that allows for easy addition of new features, sensors, and hardware.

II. Options:
   A. Define a clear API and plugin architecture that separates core functionality from customizable features and extensions.
   B. Develop a comprehensive documentation and developer guide to facilitate the creation of new extensions and customization options by third parties.
   C. Organize user and developer workshops to gather feedback on customization and extension needs, ensuring the system's architecture can accommodate these seamlessly.

III. Decision and Rationale: Options A and B were selected to balance customization with extensibility. A well-defined API and architecture provide a solid foundation for both aims, while comprehensive documentation ensures that extending and customizing the system remains accessible to developers and potentially tech-savvy users.

2.3.9 Maintaining Usability Across Language Variations

I. Description: The NFR for the system to accommodate variations in language implies a need for multi-lingual support, which could introduce inconsistencies in usability and accessibility if not implemented carefully, considering the nuances and accessibility requirements of each language.

II. Options:
   A. Implement a robust internationalization framework in the app's design, allowing for easy addition of new languages and dialects.
   B. Collaborate with native speakers and accessibility experts in each target language to ensure translations and usability meet high standards.
   C. Use automated language detection and switch capabilities to provide users with an interface in their preferred language without manual settings.

III. Decision and Rationale: Options B and C were chosen to ensure that language variations do not compromise usability. Engaging with native speakers and experts ensures the quality of translations and language-specific accessibility, while automated detection enhances user experience by seamlessly providing the app in the user's preferred language.

2.3.10 Ensuring User Comfort with Varied Feedback Mechanisms

I. Description: The requirement for the system to be comfortable for the user encompasses varied feedback mechanisms (e.g., auditory, haptic). However, the specificity of how these mechanisms are implemented to cater to different user preferences or situational needs (e.g., noisy environments) is not addressed, leading to incompleteness.

II. Options:
   A. Develop user profiles that allow for detailed customization of feedback mechanisms based on the user's preferences and typical environments.
   B. Implement smart environment detection that automatically adjusts feedback mechanisms (e.g., increasing volume in noisy areas, switching to haptic feedback in quiet or socially sensitive environments).
   C. Offer a manual override feature that allows users to quickly change feedback settings in response to their immediate context.

III. Decision and Rationale: Options B and C were selected to address this incompleteness by ensuring feedback mechanisms are both adaptable to the environment and controllable by the user, enhancing comfort and usability in diverse situations.

Improved Understanding

3 WRS

3.1 W

3.1.1 Problem

For individuals grappling with blindness, the absence of sight poses significant hurdles in their daily lives. Communicating with others becomes inherently more challenging due to the reliance on alternative methods such as braille or auditory cues. While technological advancements aim to assist the blind community, many existing solutions are plagued by high costs, bulky designs, and usability issues. Moreover, these aids often overlook the pressing need for addressing personal emergencies, leaving blind individuals vulnerable in critical situations.

3.1.2 Goal

Our objective is to develop an application tailored to assist blind individuals in overcoming the barrier of the difficulty of walking stemming from their lack of vision. This application will offer intuitive functionality, catering specifically to the needs of the blind community. It will facilitate seamless communication in various contexts, including emergencies, ensuring blind individuals have a straightforward solution to their traversal challenges.

3.1.3 Improved understanding of The Domain, Stakeholders, Functional and NonFunctional Objectives

1. (DR001) The Vision Share application will be used by those who have impaired vision, or in other terms, they require vision aid because they cannot see properly compared to the average human.

    a. Vision impairment may be caused by a loss of visual acuity, where the eye does not see objects as clearly as usual. It may also be caused by a loss of visual field, where the eye cannot see as wide an area as usual without moving the eyes or turning the head. [4]

2. (DR002) The Vision Share application has to aid those who traverse through both indoor and outdoor areas.
    a. Indoor areas: home, buildings, etc.
    b. Outdoor area: park, sidewalk, road, etc.

3. (DR003) The visually impaired will communicate with their assistant (a non-visually impaired individual whom the impaired individual wants to communicate to) through the Vision Share application.

4. (DR004) The system will allow communication through the app to the assistant.
    a. The visually impaired individual will be able to call an assistant through their device (it will take into account the blindness when doing so)
    b. The visually impaired individual will be able to talk to the assistant

5. (DR005) The application will be able to provide methods where the visually impaired can easily communicate using verbal communication to the application to get it to do what the user intends it to perform.

6. (DR006) The application will provide the following verbal aids to the user when using the application.
    a. Audio of texts through the app
    b. Understanding visual images and explanations to the user verbally
    c. Verbal communication through the application to tell it what functions the user wants it to perform

7. (DR007) The system will have a multi-dimensional lexicon with word representations that are written, audio, and visual.

8. (DR008) The priority and kind of event will be used to categorize the emergency requests.

9. (DR009) The system must be able to transmit messages to a nearby response department, such as a hospital, and serve as an interface for emergency calls, including 911 calls, to any emergency contacts.

10. (DR0010) The assistant will be able to perform any of the following operations on the application in response to the needs of the visually impaired individual:
    a. Call emergency services
    b. Navigate the visually impaired through verbal communication

3.2 RS

3.2.1 Improved Understanding of Function Requirements

1. The app must generate sentences in voice format, with the option to represent them through sound or voice commands.

2. Emergency Assistance:
   a. A verbal command will trigger emergency calls and messages.
   b. The app will prioritize contacting care providers in a customizable hierarchy: from assistive personnel to the nearest hospitals and emergency services.
   c. Users can personalize the hierarchy and contacts for emergencies.

3. The app will assign specific verbal representations to images through the assistant to reduce the ambiguity of the images that the user cannot see.

4. Users can search for visual items using verbal descriptions, and metadata associated with the images, ensuring comprehensive accessibility. The assistant will help with this function.

5. The application uses portrait and landscape for the call to make it easier for the user to communicate with the assistant, and it can change to whatever mode the user needs at the moment.

6. The app stores the most recent and most visited sites that the user goes to so it can be easier for the assistant to get the routes the the user travels through the most often

7. The app makes sure that it can generate easily understandable statements and messages promptly so that it can verbally tell the user.

3.2.2 Improved Understanding of Non-Function Requirements

1. The app's usability will be assessed through evaluations by blind users.

   Functional requirements to meet this include:
   a. The system will use a basic survey system to evaluate the experience

2. Ensuring ease of use with various features to facilitate learning.

   Functional requirements include:
   a. Integration of voice commands

3. Clear and intuitive organization of vocabulary categories.

   Functional requirements include:
   a. Disjoint categories to avoid confusion.
   b. Categories consist of related menu items for coherence.

4. Enabling dynamic and flexible generation of sentences.

   Functional requirements include:
   a. Allow users to associate visuals with verbal sentences.
   b. Permit users to customize stored sentence lists.

5. Removing the need for clicks required for actions.

   Functional requirements include:
   a. Grouping related menu items into categories.
   b. Ensuring disjoint sets of items within categories.
   c. Allowing users to customize a list of essential items.

6. Designing the system to mirror real-world calls.

   Functional requirements include:
   a. Accessible list of emergency contacts.
   c. Extensible list of basic and commonly used visuals changed to verbal form.

7. Providing responsive performance and accurate emergency functionalities.

   Functional requirements include:
   a. Utilizing compressed sound file formats to minimize loading time.
   b. Employing efficient processing policies.
   c. Ensuring accuracy in placing emergency calls and messages through rigorous testing.

8. Allowing users to personalize verbal cues and generated speech.
   Functional requirements include:

   a. Enable users to modify any verbal cue.
   b. Permit users to customize generated speech from verbal cues.

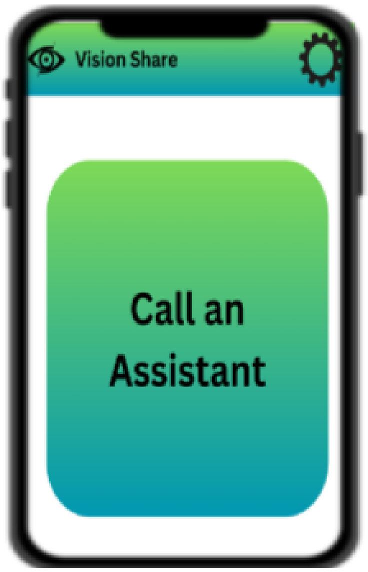
9. Easily accommodating variations in interface, language, user needs, features, and hardware.

   Functional requirements include:
   a. Allowing users to customize category verbal cues that are used.

b. Supporting language customization for audio.

4. Preliminary Prototype and User Manual

The preliminary prototype and user manual are mockups to give stakeholders an idea of what deliverables can be expected in future iterations. The team and stakeholders reserve the right to revise these mockups.



User Manual:
Note: it is assumed the user will have a screen reader to help them navigate Vision Share

To call a guide: After opening the Vision Share app select the “Call an Assistant” button in the middle of the screen. Then wait while one of the assistants connects with you.

To hang up: Select the middle “hang up” button at the bottom of the screen.

To toggle mute: Select the far left “mute” button at the bottom of the screen.

To toggle the phone light: Select the far right “light” button at the bottom of the screen.

5. Traceability

Requirement ID	Description	Forwards Traceability
NFR1	The system shall help the user safely navigate indoors. Safely is defined as having 0 injuries or incidents of physical danger to the user, or in the case of accidents or danger making emergency calls.	FR4, FR5, FR6
NFR2	The system shall lead the user through the fastest route. Fastest is defined as the route estimated to take the least time to travel.	FR1, FR3
NFR3	The system shall lead the user through the route the user would feel the most comfortable with. Comfortable is defined as the route the user gives the highest rating out of 5.	FR1, FR2, FR7
NFR4	The system shall be usable. Usable is defined as the user being able to navigate to the preferred feature with 90% accuracy.	FR1, FR2, FR4, FR8, FR11, FR12, FR13, FR14, FR15
NFR5	The system shall be ubiquitous. The application should have a 90% uptime.	FR9
NFR6	The system should be customizable to every user (e.g., the volume, the interval at which the system says something, the order whereby different things the system says, etc.) Customizable shall be defined as the user being able to make the desired change within 10 seconds.	FR10, FR11, FR12, FR13, FR14, FR15
NFR7	The system should be easily extensible to accommodate the following typical variations: variations in the interface, language, definitive needs of the user, new features, new sensors, and hardware, etc. Extensible shall be defined as the system having a modularity score of 70% or more (modularity score = extensible modules/number of modules).	FR16, FR17

Requirement ID	Description	Backward Traceability
FR1	Accepting from the user the destination location to go (The system might even suggest or confirm a possible destination location, utilizing the user’s routine schedule or habit),	NFR2, NFR3, NFR4
FR2	Figuring out and telling the user which routes can reach the destination location, and accepting the user’s preference,	NFR3, NFR4, NFR6
FR3	Telling the user to walk a distance (e.g., 2 minutes to reach a turning point, 30 steps to take)	NFR2
FR4	Telling the user to stop at the right place to turn	NFR1, NFR4
FR5	Detecting obstacles and telling the user what to do in order to avoid collision,	NFR1
FR6	Placing emergency calls and messages, possibly after detecting a fall or when the system cannot figure out the current location	NFR1
FR7	Figuring out what would be the next action(s), based on the user’s schedule or habit, and suggesting and accepting the user’s choice)	NFR3
FR8	Large Buttons to make it easier for the user to tap.	NFR4

FR9	At least one redundant server for the application to function in case of a main server failure.	NFR5
FR10	Accept input from user volume buttons.	NFR6
FR11	Make video calls	NFR4, NFR6
FR12	Interface with the user’s microphone to accept sound inputs.	NFR4, NFR6
FR13	Output audio.	NFR4, NFR6
FR14	Toggle the user’s microphone	NFR4, NFR6
FR15	Toggle the user’s flashlight	NFR4, NFR6
FR16	Follow an agile software development approach.	NFR7
FR17	Refactor the coding modules currently under development in each iteration to comply with the modularity score.	NFR7

References

[1] L. Chung, “Project Phase I: Requirements Elicitation: Initial Understanding.” personal.utdallas.edu.” University of Texas at Dallas, Richardson, Texas. [Online]. Available: https://personal.utdallas.edu/~chung/SE4351/Project1.pdf (accessed 3/28/2024)

[2] L. Chung, “A Template for WRS evolution” personal.utdallas.edu.” University of Texas at Dallas, Richardson, Texas. [Online]. Available: https://personal.utdallas.edu/~chung/SE4351/Project1.pdf (accessed 3/28/2024)

[3] L. Chung, “Requirements Engineering.” personal.utdallas.edu. personal.utdallas.edu/~chung/SE4351/syllabus.htm (accessed 3/28/2024)

[4] O. Legunsen et al., “WRS Document Project Phase 2 - Final Team Obiwan.” personal.utdallas.edu.” University of Texas at Dallas, Richardson, Texas. Accessed: Mar. 28, 2024. [Online]. Available: https://personal.utdallas.edu/~chung/RE/Presentations10F/Team-hope/2%20-%20WRS.pdf

[5] “What is Vision Impairment?” ophthalmology.pitt.edu. https://ophthalmology.pitt.edu/vision-impairment/what-vision-impairment (accessed Mar. 27, 2024).

Appendix

Tools Utilized:

- Photoshop: mockups
- Canva: mockups
- Google Drive: repository
- Google slides: presentation
- Github: website publication, repository

Technology Required to Use Application:

- GPS: determine the user's location
- Accelerometer: drop detection
- Gyroscope: orthogonal orientation
- Camera: For volunteer to see ahead of user
- Flashlight: to help light the path
- Microphone: for the user to communicate with volunteer
- Speaker: So that the user can hear the volunteer

Development Process:

Index

How much Requirements Creeping Rate can your team handle – an estimate with some rationale?

Our team has agreed on a 25% creep rate with a variance of 10% based on the simplicity of our original design. Thus we expect the number of requirements to increase a fair amount to satisfy the desires of the customer and users.

Why do you think your team's work is the best, or at least as good as other teams' work.

Our application is simple to use and easy to understand, making screen readers easy to interpret for the user. Additionally, Utilizing volunteers over call, as well as sensors is more reliable than sensors alone. Our model of utilizing volunteers is building on and improving already proven methods of utilizing technology to guide the visually impaired. Our unique and proven method is easily adaptable, due to its simplicity and the fact that a volunteer is the one that can be asked to adjust their instructions based on the user's feedback.