10-12-2025

CptS 484: Software Requirements

WRS Evolution

Requirements Elicitation

Anthony Devito, Eli Lawrence,
Ivan Quintero, Jace Dunn, Riley
Nielsen, Shawn Will

Revision History

Date	Version	Changes	Editor
9/26/2025	1.0	Initial Draft	Anthony Devito, Eli Lawrence, Ivan Quintero, Jace, Dunn, Riley Nielson, Shawn Will
10/12/2025	1.1	Final Draft	Anthony Devito, Eli Lawrence, Ivan Quintero, Jace, Dunn, Riley Nielson, Shawn Will

[1] Introduction

1.1.Purpose

The purpose of this app is to help those who are blind or visually impaired navigate specific buildings more easily, with less assistance from seeing companions. With our app, a primary user should be able to go from Point A to Point B without having to randomly navigate or remember the route.

1.2.Scope

The scope of our project is to create an app that has both a blind and companion mode (so that a seeing companion can also use the app), have a basic navigation feature that shows that the app can navigate a building, and finally that it has a functioning help mode that alerts an emergency contact if something occurs to the primary user.

1.3.Objectives and Success Criteria

1. Create a verbal navigation mode for the app
2. Create a visual/physical navigation mode for the app
3. Add a route navigation system that not only displays to the screen where to go, but verbally announces it for the primary user
4. Add personal settings
5. Add an emergency mode (possibly with fall detection)

1.4.Definitions, Acronyms, and Abbreviations

N/A

1.5.Overview

This document aims to provide detailed information on the planning, requirements, and implementation of our navigation app. It also includes some basic prototypes of the implementation to give a general idea of what we are aiming for. As we progress with the project, this document will be updated to reflect any changes that we feel would make an impact on the app.

[2] Preliminary Definition

2.1. Preliminary Domain

PD_ID	Preliminary Domain Description

PD1	The app supports navigation within and across connected buildings, allowing users to move from one location to another even if they are in different buildings.

2.2.Preliminary Functional Requirements

P FR_ ID	Preliminary FR Description
PFR1	Accepting from the user the destination location to go. It might even be able to suggest or confirm a possible destination location, utilizing the user's routine schedule or habit.
PFR2	Figuring out the routes to reach the destination, and informing the user of the options (if there are more than one), and accepting user's preference.
PFR3	Telling the user to walk a distance (e.g., 2 minutes before turning, or walk for 30 steps, etc.)
PFR4	Telling the user to stop at the right place to turn.
PFR5	Detecting obstacles and telling the user what to do in order to avoid collision.
PFR6	Placing emergency calls and messages, possibly after detecting a fall or when the system has lost its current location.
PFR7	Figuring out what the next action(s) would be, based on the user's schedule or habit, and suggesting/accepting the user's choice.

2.3.Preliminary Non-Functional Requirements

PNFR_ ID	Preliminary NFR Description
PNFR1	The system shall help the user safely navigate indoors.
PNFR2	The system shall lead the user through the fastest route.
PNFR3	The system shall lead the user through the route that the user would feel the most comfortable with.
PNFR4	The system shall be usable for blind people.
PNFR5	The system shall be ubiquitous.

PNFR6	The system shall be customizable to every user: e.g. volume, the interval of instructions, etc.
PNFR7	The system shall be easily extensible to accommodate the following typical variations: variations in interface, language, definitive needs of the user, new features, new sensors and hardware, etc.

[3] Issues with the Preliminary Definition Given

3.1.Domain Issues

Domain Issue ID	**Domain Issue Description**	
DI1	PD_ID	PD1. The domain is "indoors," allowing navigation in the same or different buildings that are connected to each other.
	1. Ambiguous or incomplete: The terms "indoor" and "connected buildings" are too broad. This ambiguity makes it difficult to scope the project. For example, an airport is a very different indoor environment than a university building. The method of connection between buildings (open courtyard, covered walkway, etc.) is also undefined.	
	Option 1	Attempt to support all types of indoor environments and connections from the start.
	Option 2	Scope the initial domain to a specific environment, like university buildings, and define "connected" as having a physically accessible indoor path.
	Option 3	Limit the initial scope to single, unconnected buildings to avoid the complexity of in-building navigation.
	Choice	Option 2
	Rationale	Option 2 provides a focused, achievable scope for the initial project phases. It allows the team to develop and test the app in a known environment, which is a key use case for a blind student on campus. It is more practical than the overly broad option 1 and more useful than option 3 since it is more restrictive.
Revised wording		

3.2.Functional Requirements Issues

FR Issue ID	Description	
FRI1	PFR_ID	PFR1 & PFR7. Suggesting destinations based on “routine schedules or habits”.
	1. Incomplete and Ambiguous: It is unclear how the system is supposed to access or learn a user’s schedule or habits. The mechanism for gathering this data is not defined.	
	Option 1	Require the user to manually enter their class/work schedule into the app during setup.
	Option 2	Use passive machine learning to observe user navigation patterns over time to predict likely destinations.
	Option 3	Allow for explicit schedule input for high reliability, and supplement this with passive learning for situational suggestions.
	Choice	Option 3
	Rationale	This hybrid approach is the most robust and user-friendly. Explicit input provides immediate, reliable suggestions, while passive learning adds convenience by adapting to non-scheduled, repeated trips over time.
Satisfied by	FR1	

FR Issue ID	Description	
FRI2	PFR_ID	PFR2. Informing the user of the options.
	1. Incomplete: The requirement does not specify how to present multiple, potentially complex routes to a blind user via audio in a way that is clear and not overwhelming.	
	Option 1	Verbally describe each route in full detail, one after another.

	Option 2	Present only the key differentiating factors to simplify the choice.
	Option 3	Automatically select a route based on the user's preset preference and only announce that other options are available if the user asks.
	Choice	Option 2
	Rationale	This method is the most efficient. It provides the essential information needed for an informed decision without the cognitive load of memorizing multiple sets of detailed instructions, making it much more usable for the primary stakeholder.
Satisfied by	FR2	

FR Issue ID	**Description**	
FRI3	PFR_ID	PFR3. Telling the user to walk a distance.
	1. Ambiguous: The units for measuring distance are not consistently defined, with examples for time, steps, and implicit distance. User preference for these units can vary greatly.	
	Option 1	Standardize all instructions to be time-based.
	Option 2	Standardize all instructions to be distance-based.
	Option 3	Allow the user to select their preferred unit of measurement in the app settings.
	Choice	Option 3
	Rationale	This choice directly supports the non-functional requirement for the system to be customizable. It allows the users to select the unit they are most comfortable with.
Satisfied by	FR3	

FR Issue ID	**Description**

FRI4	PFR_ID	PFR5. Detecting obstacles and telling the user what to do in order to avoid collision.
	1. Incomplete and Ambiguous: This requirement does not specify what qualifies as an obstacle (static or dynamic). It also doesn't define the nature of the avoidance instructions ("move left", or "stop").	
	Option 1	Detect only large, static obstacles (walls, doors, etc.) and issue a generic "Obstacle ahead" warning.
	Option 2	Use the phone's camera and machine learning to identify a broader range of obstacles and provide more specific directional guidance.
	Option 3	Focus detection only on tripping hazards below waist level, providing simple avoidance commands.
	Choice	Option 2
	Rationale	This option provides the greatest safety benefit, a top priority for the application. Using advanced smartphone features like the camera aligns with the project goals and offers a more useful and extensible solution than the other options.
Satisfied by	FR4	

FR Issue ID	**Description**	
FRI5	PFR_ID	PFR6. Placing emergency calls and messages, possibly after detecting a fall or when the system has lost its current location.
	1. Incomplete: The protocol for an emergency is not defined. It is unclear if the system calls 911 or a caretaker, and there is no mention of a mechanism to prevent false alarms.	
	Option 1	Immediately call 911 upon detecting a fall.
	Option 2	After a fall is detected, start an audible countdown, giving the user time to provide a verbal "cancel" command. If no command is given, the app calls and messages a pre-configured emergency contact with the user's last known location.

	Option 3	Simply sound a loud alarm from the phone to get the attention of people nearby.
	Choice	Option 2
	Rationale	This is the safest and most practical option. It prevents false alarms while ensuring that a designated person is contacted in a real emergency, which is more appropriate than simply making noise.
Satisfied by	FR5	

3.3.Non-Functional Requirements(NFR) Issues

NFR Issues ID	**Description**	
NFRI1	PNFR_ID	PNFR2. The system shall lead the user through the fastest route. PNFR3. The system shall lead the user through the route that the user would feel the most comfortable.
	Inconsistent and Ambiguous: These two requirements can conflict. The “fastest” route may not be the “most comfortable”. Furthermore, “comfortable” is a subjective term that is not clearly defined.	
	Option1	Default to the “fastest” route and only consider “comfort” as a tie-breaker.
	Option2	Default to the “most comfortable” route, defined as the most accessible, and treat speed as a secondary concern.
	Option3	Allow the user to select a navigation preference in the settings. The system will then generate routes based on that choice.
	Choice	Option 3
	Rationale	This approach resolves the inconsistency by letting the user define their own priorities. It

		directly supports the NFR that the system must be customizable to the user and provides the most user-friendly experience.
Satisfied by	NFR2, NFR3	

NFR Issues ID	**Description**	
NFRI2	PNFR_ID	PNFR4. The system shall be usable for blind people.
	Ambiguous: “Usable” is a subjective term and not a measurable requirement. To be effective, it must be broken down into specific operational criteria.	
	Option1	Define “usable” as the app being compatible with the phone’s built-in screen reader.
	Option2	Operationalize “usability” by requiring that all primary features can be completed using only voice commands and that all system feedback is provided audibly.
	Option3	Define usability through user testing, requiring a success rate of 95% for core tasks performed by blind users.
	Choice	Option 2
	Rationale	This option provides a clear, testable, and development focused definition of usability that is directly tied to the app’s core functionality and the primary interaction method for blind users.
Satisfied by	NFR4	

NFR Issues ID	**Description**	
NFRI3	PNFR_ID	PNFR5. The system shall be ubiquitous.

	Ambiguous and unsound: “Ubiquitous” is an unrealistic goal for a project of this scope, as it implies the app will work in any building, anywhere. The term must be re-scoped to something more achievable.	
	Option1	Interpret “ubiquitous” to mean the app must be available on both iOS and Android.
	Option2	Ignore the requirement as it may not be feasible.
	Option3	Redefine “ubiquitous” as an architectural principle. The system’s architecture should support the addition of new building maps and languages without requiring a fundamental rewrite of the application.
	Choice	Option 3
	Rationale	This option changes an unrealistic goal into a sound software engineering principle. It aligns with the requirement for the system to be extensible and ensures the project is built on a scalable foundation.
Satisfied by	NFR5	

[4] WRS

4.1.W

4.1.1. Problem

Problem ID	Problem Description	Corresponding Goals
P1	Blind and visually impaired people must rely on canes, animals, or human assistance to navigate buildings	G1

P2	Users might face obstacles such as open doors, furniture, or misplaced objects which could make navigation stressful and dangerous	G2
P3	Users might face issues setting up and configuring the tools without visual assistances	G3
P4	If the user drops their phone, it might be a issue to find it or call for help	G4

4.1.2. Goals

Goal ID	Goal Description	Backward Traceability	Forward Traceability
G1	Make an app that enables blind users to navigate buildings using voice and haptic feedback	P1	FO1
G2	Provide haptic and audio feedback in real time to alert users of obstacles during their route	P1	FO2
G3	Use phone sensors to detect obstacles and reroute the user	P2	FO2
G4	Implement fall detection and emergency contact alert	P4	FO3
G5	Add a guided setup mode that allows a secondary users to assist with setting up the app	P3	FO4

4.1.3. Improved Understanding of Domain, Stakeholders, Functional, and Non-Functional Objectives

4.1.3.1. Improved Domain

Improved Domain ID	Improved Domain Description
ID1	The domain of this app focuses on assisting visually impaired individuals navigate buildings. The app will use voice commands and haptic feedback to help users move safely around buildings
ID2	The app will allow secondary users to assist with the setup and calibration of the app.
ID3	The app will have fall detection, and emergency contact features to help make accidents such as falling less of a big deal

4.1.3.2. Stakeholders

Name	Description
Primary user	The main user who is visually impaired who uses the app to help navigate buildings
Secondary User	The secondary user who is not visually impaired who will help setup the app for the primary user
Developers	The team responsible for building the app

4.1.3.3. Improved Functional Objectives

Based on the above information and our goals, the functional objectives of HOPE are:

Improved FR Objective ID	Objective Description	Alleviates Problems	Achieves Goals
IFRO1	The app will provide indoor navigation routes using the users location, and destination. Guiding the user with audio and haptic feedback	P1	G1
IFRO2	The app will use the phone's camera and sensors to detect obstacles in the way then push alerts to the user in real time	P2	G2

IFRO3	The app will include a setup that allows a secondary user to setup the app for the primary user	P3	G3
IFRO4	The system shall detect if the users' trips or drops their phone and it will alert an emergency contact with the user's location	P4	G4

4.1.3.4. Improved Non-Functional Objectives

Improved NFR Objective ID	Objective Description	Alleviates Problem	Achieves Goal
INFRO1	The app will be reliable and provide timely responses to ensure user safety	P1	G2
INFRO2	The app will be accurate in where it positions the users and how it detects obstacles	P2	G2
INFRO3	The interface should be WCAG compliant and be fully compatible with screen readers	P3	G5
INFRO4	User data such as emergency contact and fall location will be encrypted and transmitted securely	P4	G4

4.2.RS

4.2.1. Functional Requirements

FR ID	Description
FR1	The app will allow a user to set a destination to where they want to walk using a voice command
Satisfies Functional Requirement Issue	FRI1
Satisfies Objectives	FO1

Satisfied by prototype feature	Voice command feature that takes the info and puts it into the backend for route setting

FR ID	**Description**
FR2	The app will let a secondary user to setup and calibrate the app through a GUI
Satisfies Functional Requirement Issue	FRIO3
Satisfies Objectives	FO3
Satisfied by prototype feature	Graphical UI that has settings that will setup the app for the primary user

4.2.2. Non-Functional Requirements

NFR ID	**Nonfunctional Requirement 1**	
NFR1	The system shall be secure, which means location and emergency contacts are encrypted	
Operationalized Functional Requirements	OFR1 OFR2	The app needs to encrypt user data stored locally and what is sent to emergency contacts Emergency alerts can only be shared with user approved contacts
Satisfies Nonfunctional Requirement Issue	NFRI4	
Satisfies Non-functional Objective	NFO4	
Constrains	F04	
Satisfied by prototype feature	Emergency contact alert	

NFR ID	**Nonfunctional Requirement 2**

NFR2	The app will give reliable and fast feedback to ensure users safety while navigating	
Operationalized Functional Requirements	OFR1 OFR2	The app processes potential obstacles and provide feedback within a second The app will be accurate in identifying the obstacles
Satisfies Nonfunctional Requirement Issue	NFRI2	
Satisfies Non-functional Objective	INFR02	
Constrains	FO1, FO2	
Satisfied by prototype feature	The app has working navigation/maps	

4.2.3. Specifications

Functional Specification ID	**Functional Requirement**
FS1	The app will send emergency contact a message with the primary users location if a fall is detected
Satisfies Functional Requirement	FR2
Satisfies Objectives	FO4
Satisfied by prototype feature	Fall detection

Functional Specification ID	**Functional Requirement**
FS2	The app will allow the primary user to set a destination using voice input and confirm the destination through audio playback before starting their journey.
Satisfies Functional Requirement	FR1
Satisfies Objectives	FO1
Satisfied by prototype feature	Voice command input and confirmation

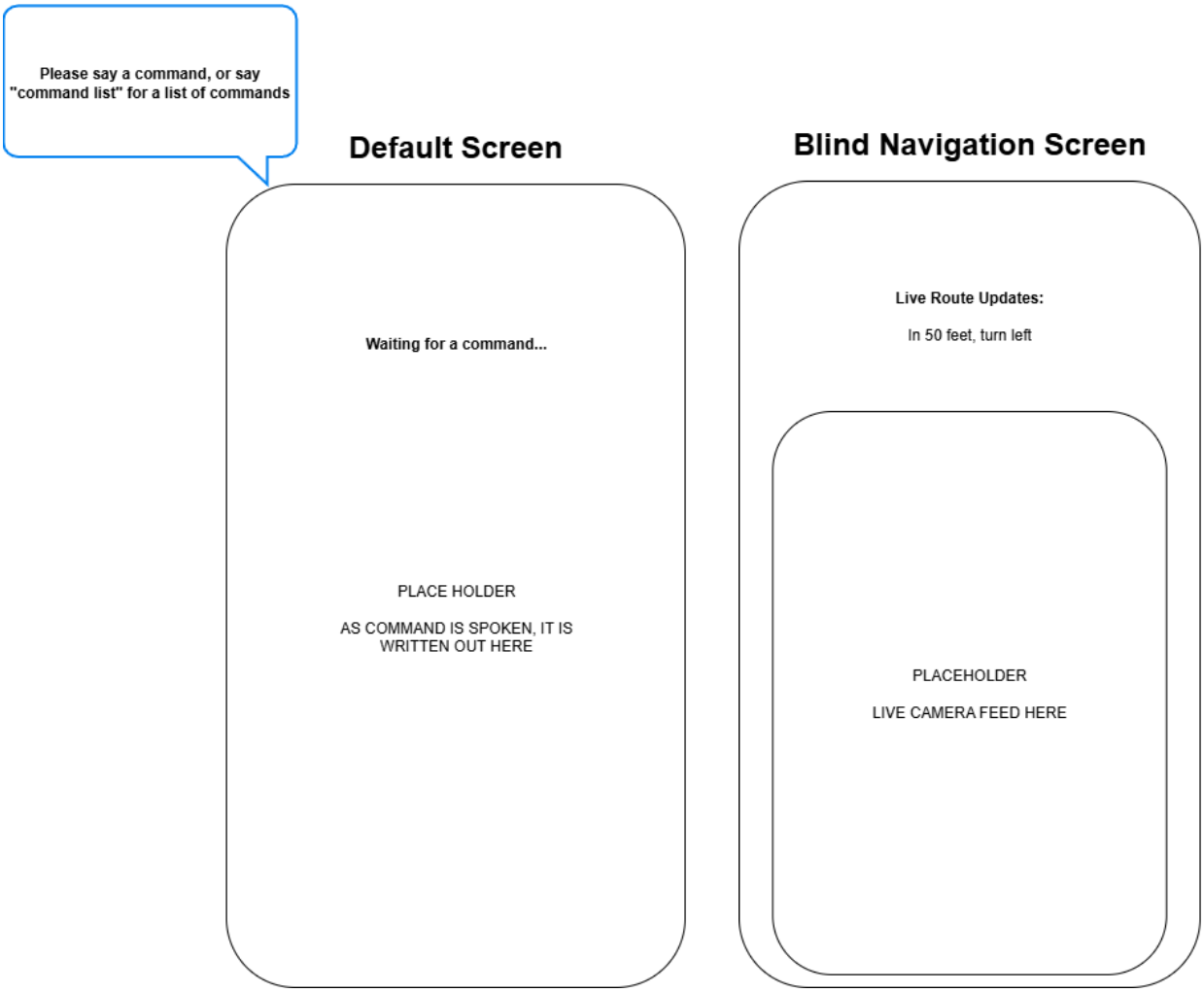
[5] Preliminary Prototype

Component	Route Navigation
Input	A destination given either verbally or textually.
Output	A list of directions that make up the route from the current location to the destination.
Implementation	This will be implemented by taking the floorplan of the building(s), creating the easiest route (the route with the least number of directions, stairs, etc.), and creating both a visual and auditory route list.
Validation	This will be simply validated by the app successfully getting the user to their destination, and if there are multiple possible paths, the app chooses the best path for the user.

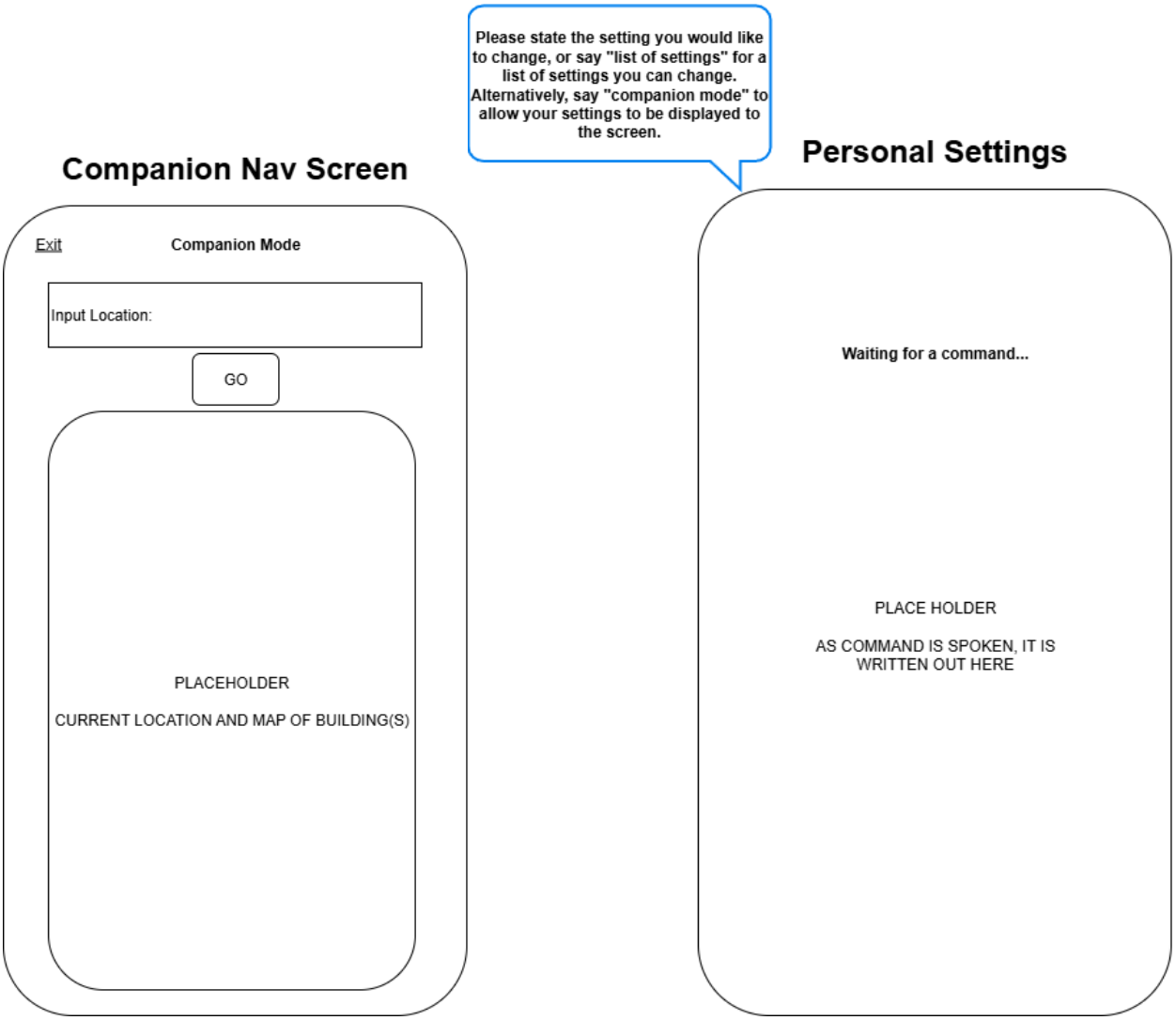
Component	Fall Detection
Input	Sensor inputs that could be a fall
Output	Verbal confirmation from the user if they are okay, or a call to their emergency contact (we won't do 911 because that would be impossible to test legally)
Implementation	This will be implemented by having the app check for harsh sensor readings that could be a fall, then appropriately dealing with the situation
Validation	This will be simply validated by the app successfully detecting a fall and acting on the detection appropriately.

Component	Companion Mode
Input	Verbal activation of the companion mode.
Output	A visual/physical screen setup that allows a seeing companion to navigate the app.
Implementation	After companion mode is activated, the user's companion will be able to navigate a visual version of the app. This includes personal settings and the basic navigation system.
Validation	This will simply be tested by having a functioning physical interface that a user can interact with after swapping to companion mode.

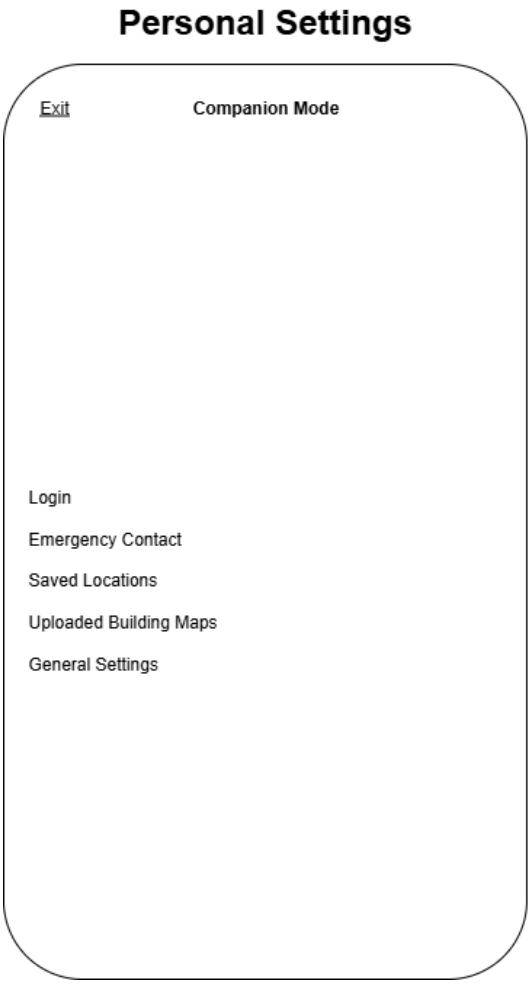
[6] Prototype Interface Mock-ups



6.1. Prototype Interface Mock-ups cont.



6.2. Prototype Interface Mock-ups cont.



[7] User Manual

1. Download the app.
2. On initial start, please grant the app access to your location, camera, and microphone (you may need companion assistance depending on your device).
3. After initial setup, the app will walk you through a handful of personal settings; you may answer these settings verbally, or swap to assistance mode and have a companion finish the setup for you.
4. Now that your personal settings are complete, on every boot of the app it will ask what you would like to do. You can say things like:
    a. “Navigate to [area]”
        i. This will start the navigation system, which will prompt you to hold your phone outright if you haven’t already, and the route will begin if the area you specified has been downloaded to the app.
    b. “Open settings”

i. This will open the personal settings menu in primary user mode where you can navigate the settings audibly.

c. “Activate companion mode”
    i. This swaps the app over to companion mode where a seeing companion can more easily navigate the app for you.

d. “Help I need assistance”
    i. If an emergency contact was added to your personal settings, this command will alert your contact that you need some sort of assistance. The app will also start alerting anyone nearby that something has gone wrong and that you need help.

[8] References

Appendix I: Process Details