EyeBallin’ WRS

Team Bob’s Bullies

Kenzo Banaag, Tim Borisenko, Dane Erosa, Rebecca Rothschild, Isaac Schultz

**Instructor**: Dr. Bolong Zeng

**Date of Report Submission**: October 13th, 2019

**CPTS 484**

School of Electrical Engineering and Computer Science

Washington State University



Table Of Contents

[**Revision History**](#_dwqkimrag94g) **2**

[**1. Introduction**](#_92p7bal91lhz) **3**

[1.1 Purpose](#_pox6lke07mjg) 3

[1.2 Scope](#_g55h9a7c264q) 3

[1.3 Objectives and Success Criteria](#_nq6vy62oyamq) 3

[1.4 Definitions, Acronyms, and Abbreviations](#_1pu2jvqd9t7r) 3

[**2. Preliminary Definition**](#_tmnwtafx55zh) **3**

[2.1 Preliminary Domain](#_b3vb10ji2zn) 3

[2.2 Preliminary Functional Requirements](#_zene55200j8f) 4

[Have an indoor map of the WSU Everett campus.](#_62s4hhak0nq3) 4

[Accept verbal input to navigate to the specified destination.](#_itqf5pvf2f2v) 4

[2.3 Preliminary Non-Functional Requirements](#_bgtm24k1z4wv) 4

[**3. Issues with the Preliminary Definition Given**](#_eeuj43yra8el) **4**

[3.1 Domain Issues](#_2s8eyo1) 4

[3.2 Functional Requirements Issues](#_2k9z7s962qc4) 5

[3.3 Non-Functional Requirements(NFR) Issues](#_5bnefdgdye81) 8

[**4. WRS**](#_lfig9p4psegd) **10**

[4.1 W](#_hv1p2f26fn3j) 10

[4.1.1 Problem](#_9012ltk2noe1) 10

[4.1.2 Goals](#_medtzx18ksbq) 11

[4.1.3 Improved Understanding of Domain, Stakeholders, Functional, and Non-Functional Objectives](#_jygotxx587vc) 11

[4.1.3.1 Improved Domain](#_2bd19mueweap) 11

[4.1.3.2 Stakeholders](#_ksae6lkv2z8v) 12

[4.1.3.3 Improved Functional Objectives](#_8d0h4oyee49h) 12

[4.1.3.4 Improved Non-Functional Objectives](#_mk17xcvadqq3) 13

[4.2 RS](#_jo6w4k85ewj) 13

[4.2.1 Functional Requirements](#_fgma9wrne1fs) 13

[4.2.2 Non-Functional Requirements](#_kcg24l2ls12b) 15

[4.2.3 Specifications](#_55sope5dg3tr) 16

[**5. Prototype Interface Mock-ups**](#_xn4v9xryc2y7) **18**

[Figure 5.1: Home Activity](#_qd9vqokb2pdh) 18

[Figure 5.2: Destination Prompt](#_sh6dpnnyt81) 18

[Figure 5.3: Destination Verification](#_39v9jbejzm1m) 18

[Figure 5.4: Fall Detection](#_z356zyu2yc9q) 19

[Figure 5.5: Navigation](#_8obxzw5d5rnj) 19

[Figure 5.6: Destination Reached](#_nv4wvnmx8ovz) 19

[**6. User Manual**](#_c60kfdgmlnky) **20**

# Revision History

Version 0.1 3:00pm 10-07-19

Preliminary layout of the document is set up.

Version 0.1.1 5:00pm 10-07-19

Added more tables and beginning writing.

Version 0.2 5:30pm 10-09-19

Rough draft of individual assignments

Version 0.3 10:00pm 10-10-19

Refined WRS. Need to clarify what the tables are actually supposed to contain

Version 0.4 1:00pm 10-11-19

Significantly re-wrote each section to provide more clarity. Added many more requirements and specifications to more accurately reflect the project goals.

Version 1.0 6:30pm 10-13-19

Significantly re-wrote each section to provide more clarity again, and updated all references between tables to be more accurate. Completed the mockup section and added a user manual

# 1. Introduction

## 1.1 Purpose

We are creating an android application to assist users in navigating the WSU building. This application is aimed to provide help for the visually impaired and is designed around their needs. Consequently, all other types of users should be able to also use this application. When used by the visually impaired, this application is intended to be used alongside existing tools such as a white cane in aiding the visually impaired. With the use of this application, the purpose is for the visually impaired to not need to use a guide dog.

## 1.2 Scope

In order to achieve the desired result in the given time constraint, we will be focusing only on the inside of the WSU Everett building. We are also limiting our development to android only, and assuming that all users will have a white cane and an android phone with an internet connection.

## 1.3 Objectives and Success Criteria

Our desired outcome is to have a complete a functional prototype by the end of the semester. The prototype will allow a user to navigate from one place to another within the WSU Everett campus using only voice directions.

|  |  |
| --- | --- |
| FO\_ID | **Functional Objective** |
| FO1 | User can interact with the application using only their voice. |
| FO2 | Application navigates user from one location on campus, to the room they specify. |
| FO3 | The route they go on is safe for the visually impaired, and it is as short as possible |

## 1.4 Definitions, Acronyms, and Abbreviations

White cane - a device used by many people who are blind or visually impaired.

Guide Dog - A specially trained dog to assist the blind with navigation.

# 2. Preliminary Definition

## 2.1 Preliminary Domain

|  |  |
| --- | --- |
| **PD\_ID** | **Preliminary Domain Description** |
| PD1 | Users are visually impaired |
| PD2 | Users have access to a handheld Android device with internet access. |
| PD3 | Users understand and fluently speak english |
| PD4 | Users will utilize a white cane |
| PD5 | Users that need assistance navigating WSU Everett |
| PD6 | Users understand braille and are capable of finding where it labels rooms, buttons on elevators, etc. |

## 2.2 Preliminary Functional Requirements

|  |  |
| --- | --- |
| P FR\_ ID | **Preliminary FR Description** |
| PFR1 | Have an indoor map of the WSU Everett campus. |
| PFR2 | Accept verbal input to navigate to the specified destination. |
| PFR3 | The application gives basic directions for walking indoors. |
| PFR4 | Detecting if the user falls. |
| PFR5 | Notifying emergency contacts in the event of a fall. |
| PFR6 | Navigate the user from their current location to a specified room. |

## 2.3 Preliminary Non-Functional Requirements

|  |  |
| --- | --- |
| **PNFR\_ ID** | **Preliminary NFR Description** |
| PNFR1 | Generate a clear walk path in less than 10 seconds |
| PNFR2 | Generate one walk path at a time |
| PNFR3 | System must recalculate route within 10 feet the user takes a wrong turn |
| PNFR4 | System must notify emergency contacts within 10 seconds of detecting fall. |

# 3. Issues with the Preliminary Definition Given

## 3.1 Domain Issues

|  |  |  |
| --- | --- | --- |
| **Domain Issue ID** | **Domain Issue Description** | |
| DI1 | PD1,  PD2,  PD4,  PD5,  PD6 | Users are visually impaired  Users have access to a handheld Android device with internet access.  Users will utilize a white cane  Users that need assistance navigating WSU Everett  Users understand braille and are capable of finding where it labels rooms, buttons on elevators, etc. |
| 1. Potentially ambiguous or incomplete definition. | |
| Solution | Assume the user is only visually impared. All other motor, cognitive and sense related capabilities are fully intact. The user participating is also expected to be using a white cane while using navigation functionality with the application. The user is also able to independently read braille in locations such as an elevator to press the correct button. A visually abled third party is expected to perform the initial setup for the application including download it and open it for the user. |
| Rationale | This provides the most complete domain knowledge of our specific target audience. |
| DI2 | PD3 | Individuals understand and fluently speak english |
| 1. Potentially ambiguous or incomplete definition. | |
| Solution | Assume the user is capable of speaking clear and understandable fluent english in addition to understanding english spoken by an android phone’s text to speech voice. |
| Rationale | This provides the most complete domain knowledge of our specific target audience. |

## 3.2 Functional Requirements Issues

|  |  |  |
| --- | --- | --- |
| **FR Issue ID** | **Description** | |
| FRI1 | PFR1 | Having an accurate and detailed map of the inside of WSU Everett, including all the floors. |
| 1. How to decide where the map data will come from. | |
| Option 1 | Use Google Maps Indoors API |
| Option 2 | Use Google Maps Indoors API and add our own floor data |
| Option 3 | Create our own floor map and add our own data |
| Choice | Option 2 |
| Rationale | The Google Maps Indoors API does not currently have the insides of the WSU Everett building mapped out. So we need to figure out either how to use their API with our own data, or how to add the floorplans to the API. |
| Satisfied by | **FR1** | |
| FRI2 | PFR2 | Receive input from the user |
| How will we get input from someone who is visually impaired? | |
| Option 1 | We can use speech to text software to translate what is being said into text, and process commands. |
| Option 2 | For the visually impaired, but not completely blind, we can have large and clear on-screen buttons to allow simple “yes” or “no” type input. |
| Option 3 | Use a combination of touches anywhere on the screen to correspond to different input parameters. |
| Choice | 1, possible with 2/3 as well. |
| Rationale | Since we want our application to be as simple as possible, using speech to text software will work for anyone who is visually impaired.  But this might be clunky for gathering simple input, it may make sense to implement all 3 and give the user the ability to switch between them depending on which is easiest at that time. |
| Satisfied by | **FR2, FR3** | |
| FRI3 | PFR3 | Give output to the user |
| How will we give output to someone who is visually impaired | |
| Option 1 | Use text to speech software |
| Option 2 | Display text in high contrast large print for the visually impaired, but not totally blind users. |
| Choice | 1, but maybe 2 along with it |
| Rationale | There is really only one way to ensure all users can receive the output from the application, and that is text to speech. It might be beneficial to also display some text on the screen for users who can see it, but it might be a large consumer of the devices power. |
| Satisfied by | **FR2,** **FR3** | |
| FRI4 | PFR3 | Determining where the user is within a building |
| How will we accurately discern where the user is in the building, including what floor they are on | |
| Option 1 | Use GPS for position data |
| Option 2 | Ask the user what floor they are on |
| Option 3 | Use sensors placed around the building to determine location |
| Choice | 1 |
| Rationale | Using GPS might not be extremely accurate for walking instructions within a building, as most commercial GPS are only accurate to about 10 feet. However, there is no other good alternative that will be possible within the scope of this project. |
| Satisfied by | **FR5** | |
| FRI5 | PFR4 | Detect a fall |
| How will we detect when a user falls? | |
| Option 1 | Use Accelerometer data to detect sudden acceleration. Ask the user if they are okay, |
| Option 2 | Use the microphone to detect a sudden sound |
| Choice | 1 |
| Rationale | The accelerometer is the best tool to detect a fall, as it specifically measures acceleration. Falling would produce a sudden spike, which would be easily distinguishable as a fall. |
| Satisfied by | **FR4** | |
| FRI6 | PFR5 | Notify emergency contacts of a fall |
| How will we notify the emergency contacts that the user has fallen? | |
| Option 1 | Call and text the emergency contact using the user phone. |
| Choice | 1 |
| Rationale | Calling and texting emergency contacts ensures that they have a precise location and immediate notification of a fall. |
| Satisfied by | **FR4** | |
| FRI7 | PFR6 | Navigate the user from their current location to a specified room. |
| How will we instruct the user where to go? | |
| Option 1 | Use text-to-speech to output directional commands. Giving the user the current command, and the next steps they must take after that. |
| Option 2 | Display the route, current navigation step, and future steps on the phone screen in large high contrast text optimized for low power use on an OLED screen. |
| Choice | 1 and if specified by the user, 2 |
| Rationale | It might be helpful to the user to have both a visual and verbal representation of the route they must take. The audio directions can assist all visually impaired users, while the on-screen directions can assist those who may have partial vision. |
| Satisfied by | **FR5, FR6** | |

## 3.3 Non-Functional Requirements(NFR) Issues

|  |  |  |
| --- | --- | --- |
| **NFR Issues ID** | **Description** | |
| NFRI1 | PNFR1 | Generate a clear walking path in less than 10 seconds. |
| How will we ensure fast path generation? | |
| Option 1 | Use the Wifi capabilities of devices to utilize a service for creating paths. This will depend on which mapping option we use. |
| Option 2 | Implement a “shortest path” algorithm ourselves. This should take into account methods that are not accessible for the blind (like stairs). |
| Option 3 | Use sensors placed around the building to navigate users as they go. |
| Choice | 2 |
| Rationale | Since the secondary source for mapping buildings, does not provide navigation APIs, we must implement out own “shortest path” algorithm. |
| Satisfied by | NFR2 | |
| NFRI2 | PNFR2 | Generate one walk path at a time |
| How will we ensure that only one path is generated by a single user? | |
| Option 1 | Hide the route generation interface from the user until they indicate they’re done or reach their destination. |
| Option 2 | Only allow one route creation per app instance. |
| Choice | 1 |
| Rationale | Once users enter a destination and begin traveling the route, they should not be able to return to the route creation interface until they end the current navigation. |
| Satisfied by | NFR1 | |
| NFRI3 | PNFR3 | System must recalculate route within 10 feet the user takes a wrong turn |
| How will implement route recalculation? | |
| Option 1 | Use GPS to identify when a user is off track |
| Option 2 | Use sensors placed around building to identify when a user is off track |
| Option 3 | User can tell system when they are lost |
| Choice | 1 |
| Rationale | Since GPS will already be used, it is the best choice to continue using it for this purpose. |
| Satisfied by | NFR2 | |
| NFRI4 | PNFR4 | System must notify emergency services within 10 seconds of detecting a fall |
| How will we ensure fast detection and notification? | |
| Option 1 | Immediately notify when fall is detected |
| Option 2 | Ask the user if they want to notify emergency contact. |
| Option 3 | If user does not respond within 10 seconds, notify the emergency contact. |
| Choice | 3 |
| Rationale | When the application detects a fall, it should be able to ask the user if they are okay, then notify emergency contacts to remove potential false positives |
| Satisfied by | NFR2 | |

# 4. WRS

## 4.1 W

### 4.1.1 Problem

|  |  |  |
| --- | --- | --- |
| **Problem ID** | **Problem Description** | **Corresponding Goals** |
| P1 | Not having precise location to tell exactly what room or floor the user is on | G1 |
| P2 | Speech to text possibly misinterpreting a word, or text to speech pronouncing a word abnormally. | G2 |
| P3 | Not detecting quickly enough if a user travels in the wrong direction | G3, G6 |
| P4 | Not detecting a fall, or detecting a fall when there was no fall | G4 |
| P5 | Generating an incorrect, or inefficient path from one room to another | G5, G6 |
| P6 | The map of the building being inaccurate in some areas | G6 |

### 4.1.2 Goals

|  |  |  |  |
| --- | --- | --- | --- |
| **Goal ID** | **Goal Description** | **Backward Traceability** | **Forward Traceability** |
| G1 | Detect correct room and floor 90% of the time | P1 | IFRO1, IFRO3, INFRO3 |
| G2 | Correctly determine the spoken command 90% of the time | P2 | IFRO4, INFRO4, INFRO5 |
| G3 | Notify user if not on track if they travel more than 10 feet in the wrong direction | P3 | IFRO1, INFRO3 |
| G4 | Detect falls with over 90% accuracy | P4 | IFRO2 |
| G5 | Successfully navigate user from one room to another 80% of the time | P1, P2, P3, P5 | IFRO1, IFRO3, IFRO5, INFRO1, INFRO2, INFRO3, INFRO5 |
| G6 | Have an up to date map of the insides of WSU Everett that includes every room, hallway, door, stairwell, wall, and elevator. | P6 | IFRO3 |

### 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 | EyeBallin’ shall provide functions that allow a person living with visual impairment listed in 3.1 to augment their ability to navigate within buildings. |
| ID2 | In order to use EyeBallin’, as listed in 3.1, a user is required to have access to a white cane, along with a handheld Android device with a functioning display, microphone, and speaker. |
| ID3 | The user, as outlined in 3.1, must be able to fluently speak and understand english, as well as read braille. |

#### 4.1.3.2 Stakeholders

|  |
| --- |
| Stakeholders |
| Bolong |
| Visually Impaired people |
| WSU Everett Faculty |
| The Development Team |
| Caretakers for the Visually Impaired |

#### 4.1.3.3 Improved Functional Objectives

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

|  |  |  |  |
| --- | --- | --- | --- |
| **Improved FR**  **Objective ID** | **Objective Description** | **Alleviates Problems** | **Achieves Goals** |
| IFRO1 | The system shall use previous location history to predict with 90% accuracy what room and floor the user is in. | P1, P3, P5, P6 | G1, G3, G5 |
| IFRO2 | The system shall detect falls, clarify with the user if they need help, and notify emergency contacts. | P4 | G4 |
| IFRO3 | The system shall store a detailed map of WSU Everett | P6 | G1, G6 |
| IFRO4 | The system shall interpret user’s voice into text, and output text into speech. | P2 | G2 |
| IFRO5 | The system shall calculate a route from one location to a room in the building. | P5 | G1, G3, G5 |

#### 4.1.3.4 Improved Non-Functional Objectives

|  |  |  |  |
| --- | --- | --- | --- |
| **Improved NFR**  **Objective ID** | **Objective Description** | **Alleviates Problem** | **Achieves Goal** |
| INFRO1 | System should be able to generate accurate walk paths in less than a second. | P1, P3, P5, P6 | G5 |
| INFRO2 | Generate one walk path at a time | P5 | G5 |
| INFRO3 | System must accurately recalculate route within a few seconds the user moves in the wrong direction. | P3, P1, P5 | G3, G5, G1 |
| INFRO4 | System must accurately translate speech to text and text to speech when speaking normally. | P2 | G2 |
| INFRO5 | EyeBallin' shall lead the person with clear instructions. | P2, P5 | G2, G5 |

## 4.2 RS

### 4.2.1 Functional Requirements

|  |  |
| --- | --- |
| **FR ID** | **Description** |
| **FR1** | The system uses the google indoors API with the floorplan for the WSU Everett building. |
| Satisfies Functional Requirement Issue | FRI1 |
| Satisfies Objectives | IFRO1, IFRO3,, INFRO2, INFRO3 |
| Satisfied by prototype feature | Custom API for google indoors |
| **FR2** | The application is able to receive input from the user and respond according to their commands |
| Satisfies Functional Requirement Issue | FRl2, FRI3 |
| Satisfies Objectives | IFRO2, IFRO4, INFRO4, INFRO5 |
| Satisfied by prototype feature | Voice Interaction System |
| **FR3** | The application is able to give output to the user and respond according to their actions |
| Satisfies Functional Requirement Issue | FRI2, FRl3 |
| Satisfies Objectives | IFRO2, IFRO4, INFRO4, INFRO5 |
| Satisfied by prototype feature | Voice Interaction System |
| **FR4** | The application is able to detect when a user falls and notify the users emergency contacts in case of an emergency |
| Satisfies Functional Requirement Issue | FRl5, FRI6 |
| Satisfies Objectives | IFRO2 |
| Satisfied by prototype feature | Fall Detection System |
| **FR5** | The application is able to accurately detect where a user is within the building using GPS relative to a Google Maps Indoors floor map of the building. |
| Satisfies Functional Requirement Issue | FRl1, FRI4, FRI7 |
| Satisfies Objectives | IFRO1, IFRO3 |
| Satisfied by prototype feature | GPS location system |
| **FR6** | The System is able to route the user to their destination using a combination of audio directions, supplemented by written directions on the phone screen displayed in a high-contrast easily readable form to maximize utility for all users. |
| Satisfies Functional Requirement Issue | FRI7 |
| Satisfies Objectives | IFRO1, IFRO3, IFRO4, IFRO5, INFRO1, INFRO2, INFRO3, INFRO4, INFRO5 |
| Satisfied by prototype feature | Voice Interaction System |

### 4.2.2 Non-Functional Requirements

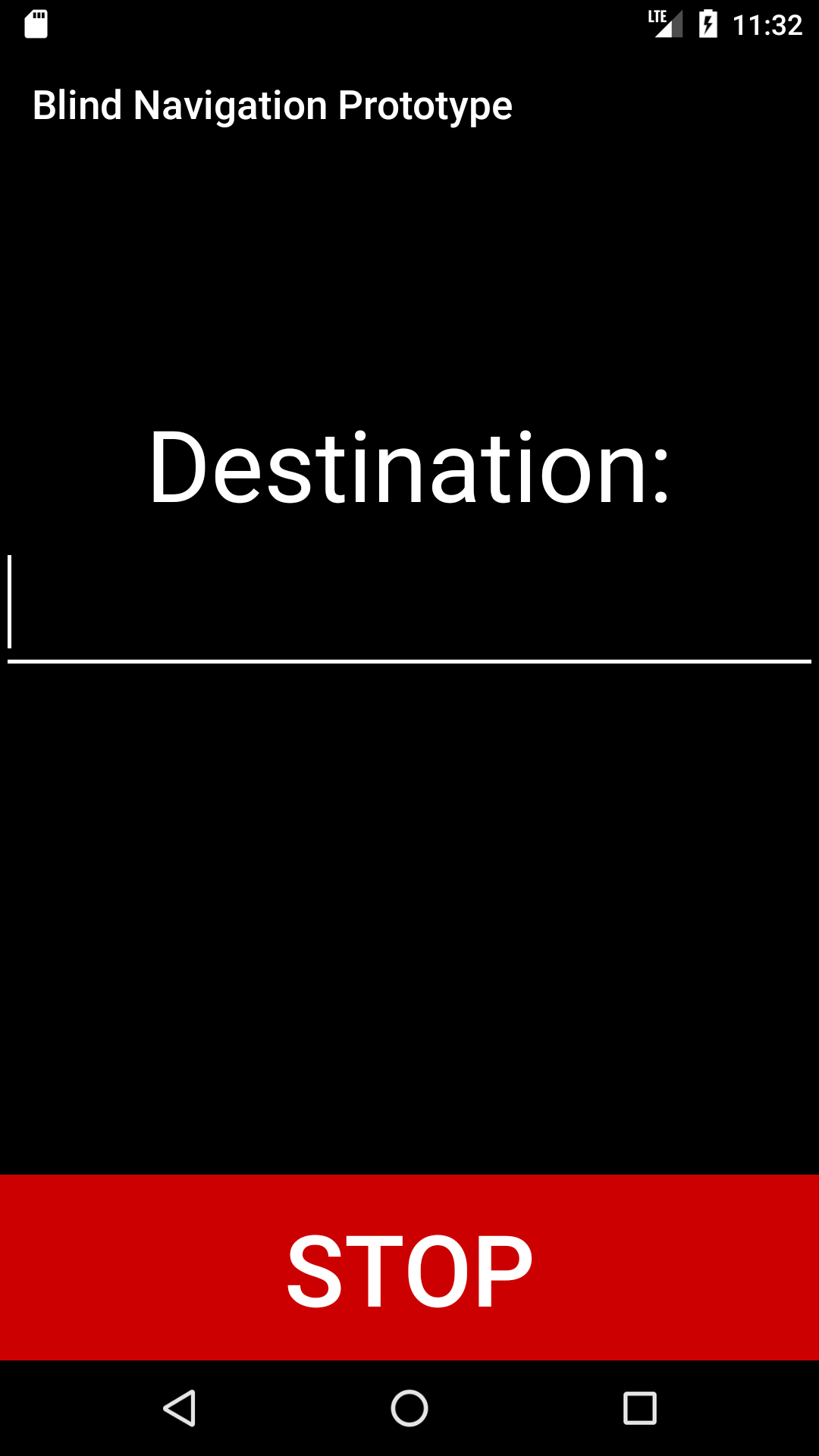
|  |  |  |
| --- | --- | --- |
| **NFR ID** | **Nonfunctional Requirement 1** | |
| NFR1 | The system shall be reliable, ensuring the system will not lead the user to dangerous and unsafe paths. | |
| Operationalized Functional Requirements | OFR1  OFR2  OFR3 | The system shall generate accurate and safe paths  The system shall generate only one path to ensure reliability  The system must recalculate when the user takes a wrong turn |
| Satisfies Nonfunctional Requirement Issue | NFRI1, NFRI2, NFRI3 | |
| Satisfies Non-functional Objective | INFRO1, INFRO2, INFRO3 | |
| Constrains | IFRO5, INFRO5 | |
| Satisfied by prototype feature | Custom API for google indoors | |
| NFR2 | The system must perform in close to real time. | |
| Operationalized Functional Requirements | OFR4  OFR5  OFR6 | The system must generate accurate walk paths in a matter of seconds  The system must recalculate in a matter of seconds after the user takes a wrong turn  The system must emergency services in a matter of seconds after the user is unresponsive |
| Satisfies Nonfunctional Requirement Issue | NFRI1, NFRI4 | |
| Satisfies Non-functional Objective | INFRO1, INFRO3 | |
| Constrains | IFRO5, INFRO5 | |
| Satisfied by prototype feature | Custom API for google indoors | |

### 4.2.3 Specifications

|  |  |
| --- | --- |
| **Functional Specification ID** | **Functional Requirement** |
| **FS1** | The system uses the google indoors API with the floorplan for the WSU Everett building. |
| Satisfies Functional Requirement Issue | FRI1 |
| Satisfies Objectives | IFRO3 |
| Satisfied by prototype feature | Custom API for google indoors |
| **FS2** | The system uses the built in android speech-to-text features to receive commands from the user. The system should have a set of acceptable commands that it is listening for. If it does not understand user input, it should ask for clarification. |
| Satisfies Functional Requirement Issue | FRl2 |
| Satisfies Objectives | IFRO4, INFRO4, INFRO5 |
| Satisfied by prototype feature | Voice Interaction System |
| **FS3** | The system uses built in android text-to-speech features to give output to the user. |
| Satisfies Functional Requirement Issue | FRl3 |
| Satisfies Objectives | IFRO4, INFRO4, INFRO5 |
| Satisfied by prototype feature | Voice Interaction System |
| **FS4** | The system is able to detect when a user falls using the device’s accelerometer. If it detects a fall, confirm with the user if they are OK. If the user does not respond, or says they are not OK, then notify their emergency contacts of the fall via phone call, and text them the user’s GPS coordinates. |
| Satisfies Functional Requirement Issue | FRl5, FRI6 |
| Satisfies Objectives | IFRO2 |
| Satisfied by prototype feature | Fall Detection System |
| **FS5** | The system is able to accurately detect where the user is within the building using a combination of GPS coordinates, and location history to determine if they have gained or lost altitude. |
| Satisfies Functional Requirement Issue | FRl4 |
| Satisfies Objectives | IFRO1, INFRO1 |
| Satisfied by prototype feature | GPS location system. |
| **FS6** | The System can route the user through the building using a combination of walking and turning instructions, along with possible directions for what buttons to press in an elevator. |
| Satisfies Functional Requirement Issue | FRI7 |
| Satisfies Objectives | IFRO1, IFRO4, IFRO5, INFRO1, INFRO2, INFRO3, INFRO4, INFRO5 |
| Satisfied by prototype feature | Voice Interaction System |

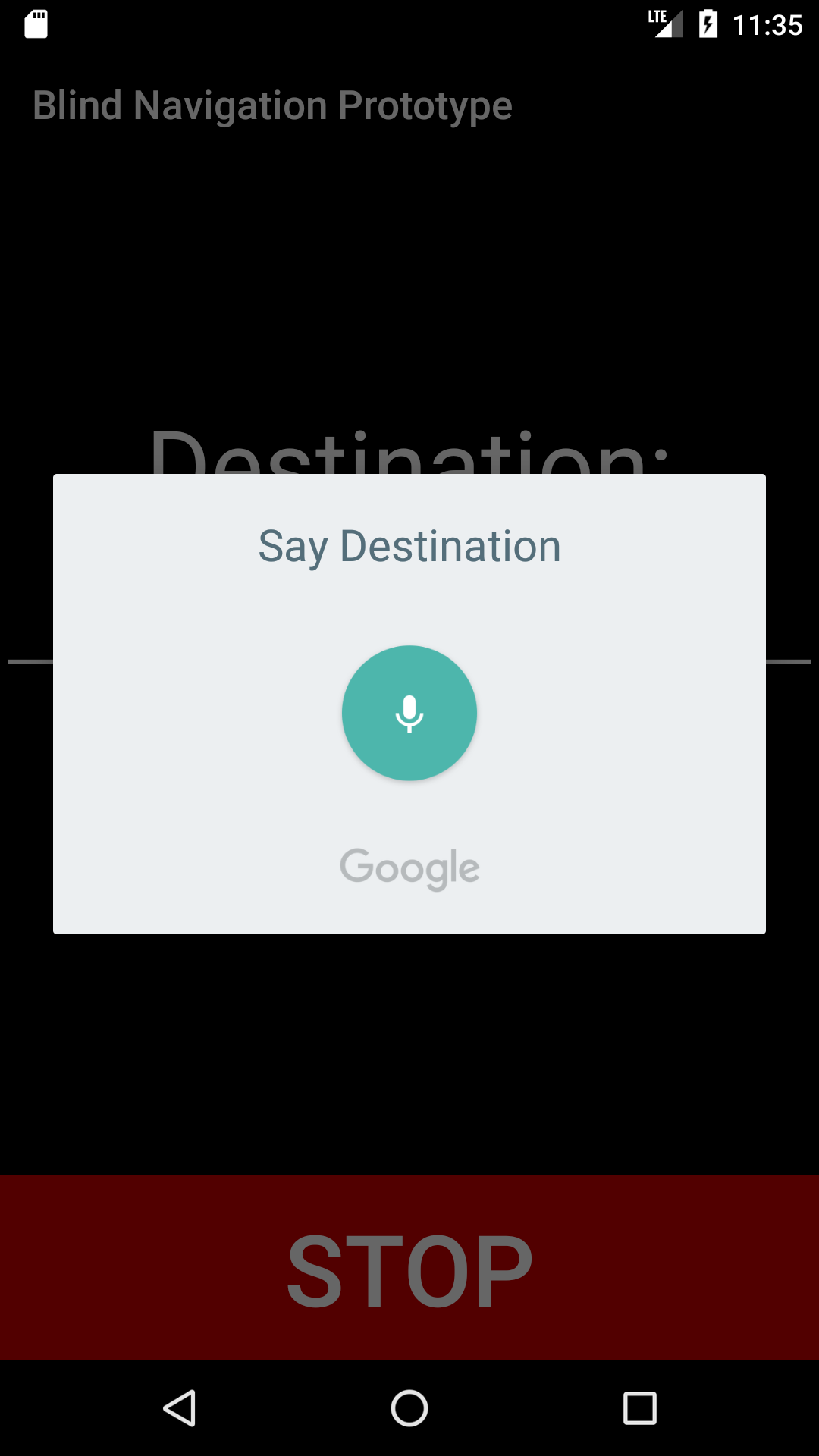
# 

# 5. Prototype Interface Mock-ups



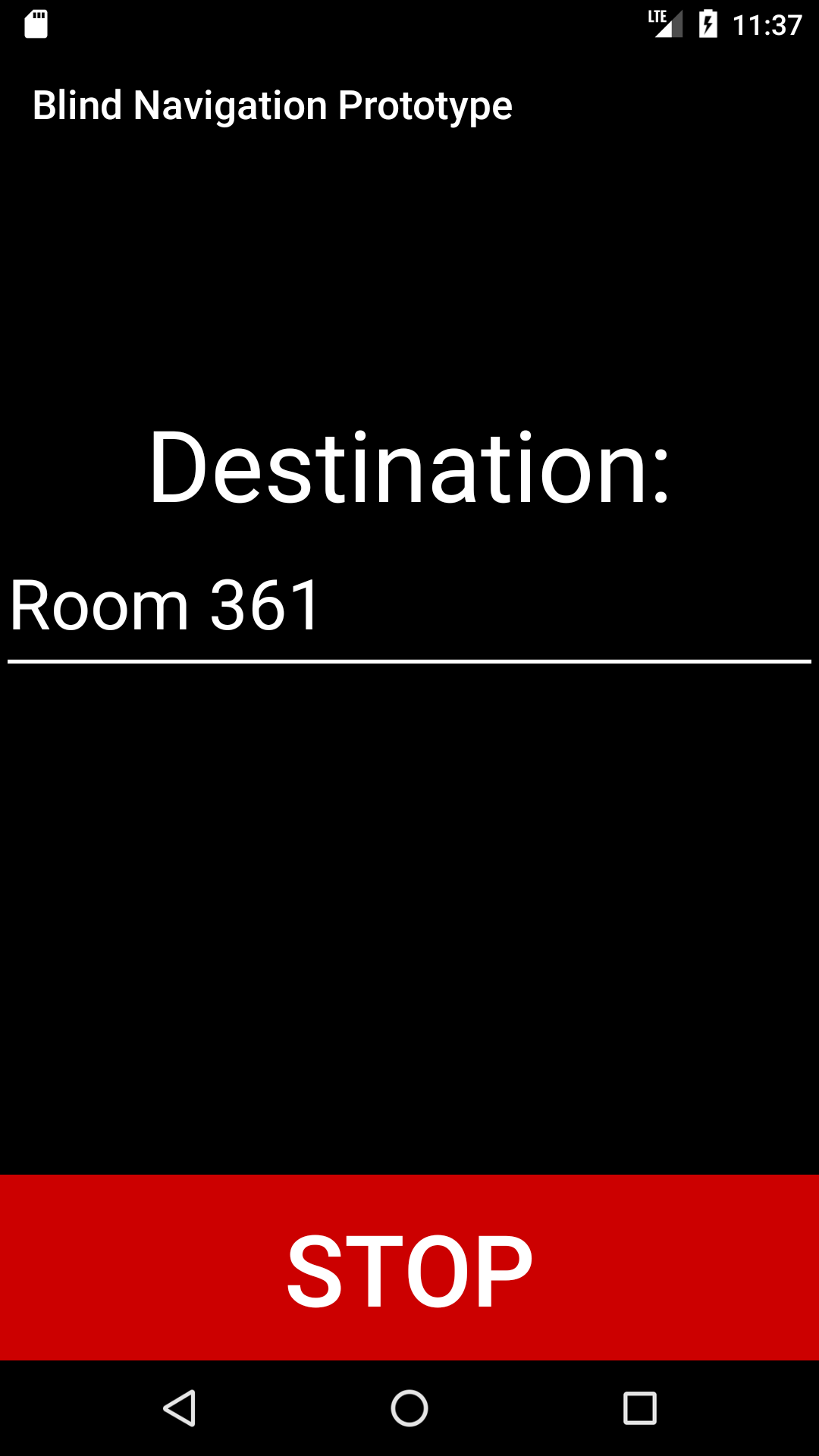
##### *Figure 5.1: Home Activity*

When the app is opened, this page will be loaded in and the app will automatically say a greeting and will prompt the user for a command. The user can say a destination, settings or stop.



##### *Figure 5.2: Destination Prompt*

This feature prompts as soon as the app is opened and the destination page appears. The user can click off and enter the destination manually in the menu from **Figure 5.1**.



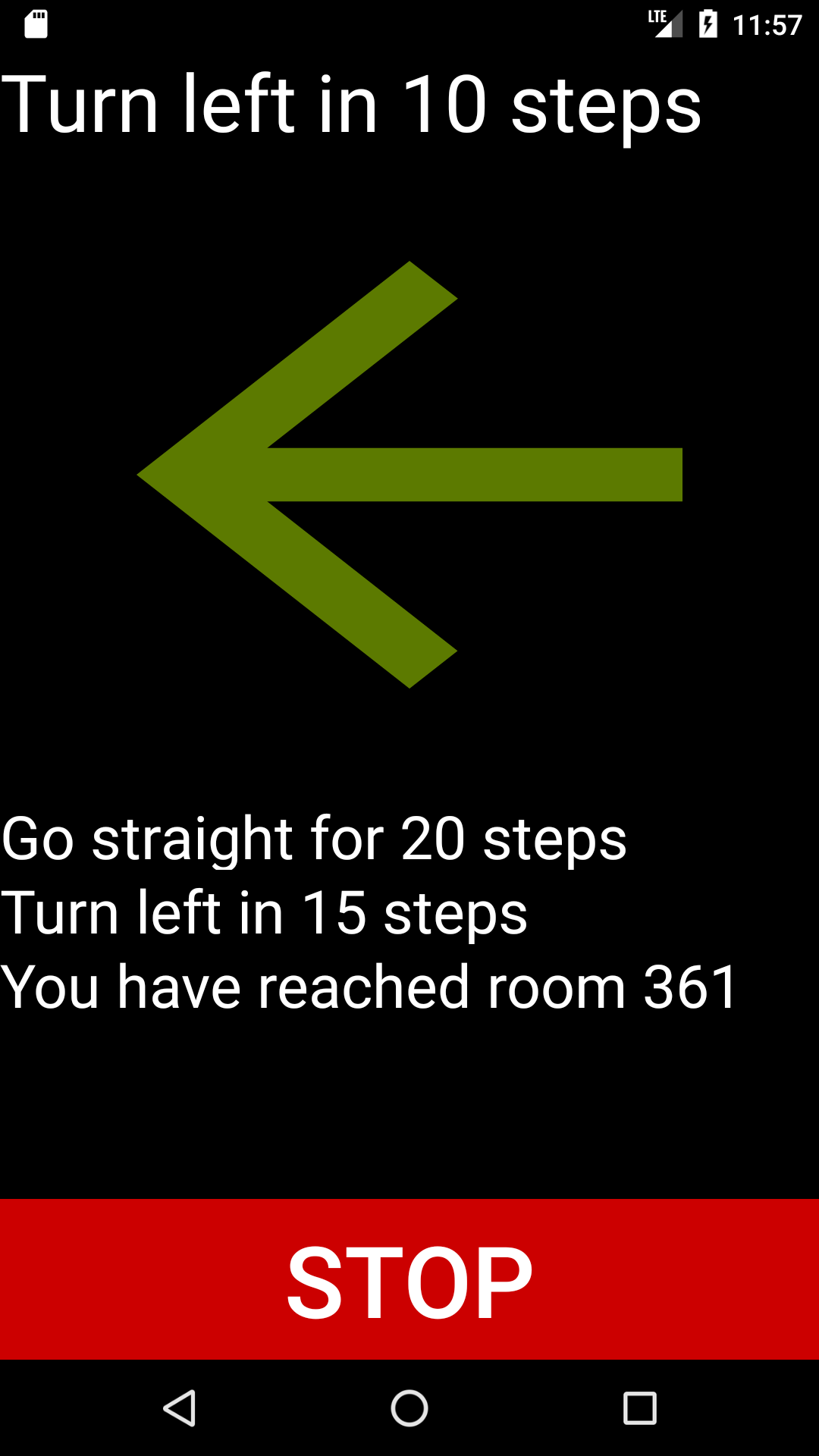
##### *Figure 5.3: Destination Verification*

The application will ask the user if the destination was correctly stated. If it was it will proceed to **Figure 5.5.** If it wasn't it will go back to **Figure 5.2** and repeat the process.

##### 

##### *Figure 5.4: Fall Detection*

This feature prompts if the user falls on the ground. The app will ask the user if they are okay with a 10 second window for the user to respond notifying the app that they are okay.



##### Figure 5.5: Navigation

The application will both display and say directions to the user. The visual arrow will show the user the direction to go. The next set of directions will also be displayed in the bottom of the arrow.



##### Figure 5.6: Destination Reached

The application will prompt the user that they have arrived at their destination and will close in a few seconds.

# 

# 6. User Manual

\*Assumptions: the User Manual is created for a sighted assistant to help a visually impaired individual learn the application

**EyeBallin' Navigation System**

EyeBallin’ is a system created to help you navigate indoors. This application was designed to work for the visually impaired as well as anyone who may need assistance navigating a building. We are committed to helping others and keeping our users satisfied. If for any reason you may have questions or complaints, don't hesitate to contact us at 1-800-IBA-LLIN.

**Navigating to desired location**

1. Open EyeBallin'.
2. The application will automatically prompt you for a location you wish to navigate to. It will say “Where would you like to go?” The user must say what room they want to navigate to.
3. The application will proceed to navigate them there.
4. If the user doesn’t respond or the system cannot find the specified location. The system will say “Cannot find room” and “Where would you like to go?” and will wait for the user to specify the location they want to go to.

Ex. If you want to navigate to room 363 at WSU Everett you would say “Room 363” after the navigation system asks “Where would you like to go?”

1. If they user wants to go to the settings menu they can say “settings” instead of a physical location such as “Room 363”

**Navigation settings**

1. You may lower the sound volume of the navigation by lowering the volume on your device.
2. You may stop the navigation at any time by saying “Stop” or clicking the stop button on the bottom of the screen.

**Emergency Contact**

*This application is designed to notify an emergency contact when the user falls*

*or gets hurt. Thus, it is recommended that you input the emergency contact(s) in the settings menu. In case of an emergency the system will call the immediate contact and will say an automatic voice message. The system will also text the emergency contact the building and coordinates of the users location.*

1. Click on the settings menu located in the upper right corner -> Click on emergency contact info in the settings.
2. Input your preferred emergency contact and click save.
3. Alternatively the user can navigate to the settings menu and input emergency contact information via voice input.

Ex. User can say “settings” when system asks “where do you want to go?” The system will then ask “what would you like to do?” the user should respond “emergency contact”, the system will then ask “what phone number should we dial in case of an emergency”? The user should respond with an actual phone number such as “425-738-4441”.

**Emergency Safety Confirmation**

*When the system detects a fall, it will assume that the user is in immediate danger and will ask if they're okay. The user must confirm by saying “yes” to the system, otherwise after 10 seconds of no response the system will notify the emergency contact. If the emergency contact has not been specified, the WSU Everett Building Security will be notified. Optionally the user can confirm they are okay by clicking on the “cancel” button.*