EyeBallin' WRS

Team Bob's Bullies

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

Instructor: Dr. Bolong Zeng

Date of Report Submission: December 8th, 2019

CPTS 484

School of Electrical Engineering and Computer Science
Washington State University

Table Of Contents

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
Version 1.1 4:00pm 12/4/2019
Added some of the KAOS goals model diagrams to the goals sections. Commented on what needs to To be revised in the coming week.
Version 2.0 2:55pm 12/8/2019

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, excluding faculty offices. 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 or cellular service.

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 effectively a user to navigate from one place to another within the WSU Everett campus using only voice directions. They must simulate their location and movements using onscreen arrows.

FO_ID	**Functional Objective**
FO1	The user can set a destination using their voice and receive the first direction, but they must use onscreen arrows to move their location to receive ensuing directions to the destination.
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 internal representation of the indoor map of the WSU Everett campus, excluding the faculty offices.
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 building security 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 of the user going off track
PNFR4	System must notify building security within 15 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 classrooms, bathrooms, seminar rooms, and main entrances, 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 3
	Rationale	The team will extend the Android location class and create a custom floor map for each level of the building and then populate the map manually.
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	Option 1
	Rationale	The rationale behind using option 1 is that the app is designed for the visually impaired. Generally the consensus is to tailor the app to the broad spectrum of the visually impared and the best way to cater to the wide spectrum is to have something easy to use like text to speech, that doesn’t require any type of vision to use.
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. Text directions will be displayed on screen for prototype testing purposes.
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
	Option 4	Always start the user at the North doors and simulate the users location in app.
	Choice	4
	Rationale	GPS signal is not accessible within the WSU Everett building. Option 4 will be used to discern where the user is in the building by simulating the users location in an extended Android Location class and using on screen buttons to move them around. The application will assume that the user is always beginning at the North doors
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 that the user has fallen
	How/ who will we notify that the user has fallen?	
	Option 1	Notify an emergency contact using the user phone.
	Option 2	Notify building security
	Option 3	Via call
	Option 4	Via text
	Choice	2 & 4
	Rationale	Texting building security will ensure they have a record that somebody fell including their location. The text will be received within 10 minutes.
Satisfied by	**FR4**	
FRI7	PFR6	Navigate the user from their current location to a specified room.
	How will the app instruct the user where to go?	
	Option 1	Use text-to-speech to output directional commands. Only one direction will be given at a time to avoid confusing the user.
	Option 2	Display the current direction text on the screen.
	Choice	1 & 2
	Rationale	The focus of the application will be on option 1, but option 2 will also be implemented to help with debugging.
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	A service for path generation was not found, an adaptation of Dijkstra’s shortest path algorithm must be implemented.
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 & 2
	Rationale	Once users enter a destination and begin traveling the route, they should not be able to return to the route creation interface. The user will have to open a new map instance to generate a new route.
Satisfied by	NFR1	
NFRI3	PNFR3	The app will recalculate the route given the user has gone at least three feet in the wrong direction.
	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

	Option 4	Use simulating arrows to know the direction the user should be going and detect when the user has taken multiple steps in a direction different from the simulated arrow
	Choice	Option 4
	Rationale	It will be trivial to implement this way with the other choice to use on screen arrows to move the user's location.
Satisfied by	NFR2	
NFRI4	PNFR4	System must notify emergency services within 15 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, wait 10 seconds for a response, 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	IFRO2, INFRO3
G2	Correctly determine the spoken command 90% of the time	P2	IFRO3, INFRO4, INFRO5
G3	Notify user if not on track if they travel more than 3 feet in the wrong direction	P3	INFRO3
G4	Detect falls with over 90% accuracy	P4	IFRO1
G5	Successfully navigate user from one room to another 80% of the time	P1, P2, P3, P5	IFRO2, IFRO4, INFRO1, INFRO2, INFRO3, INFRO5
G6	Have an internal representation of the WSU Everett classrooms, bathrooms, hallways, elevators, and entrances	P6	IFRO2

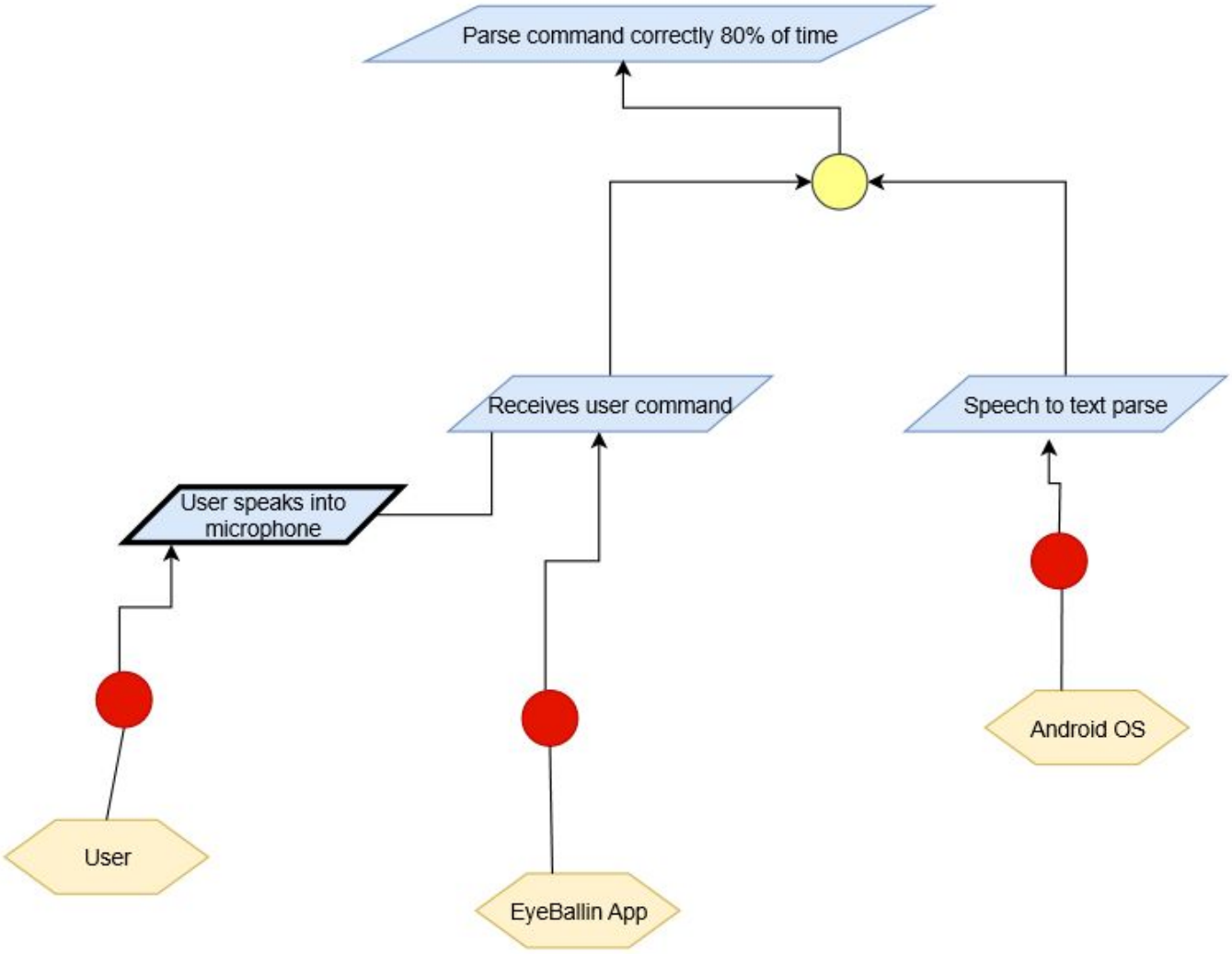

Figure 4.1.2.1 Parse Command correctly 80% of time

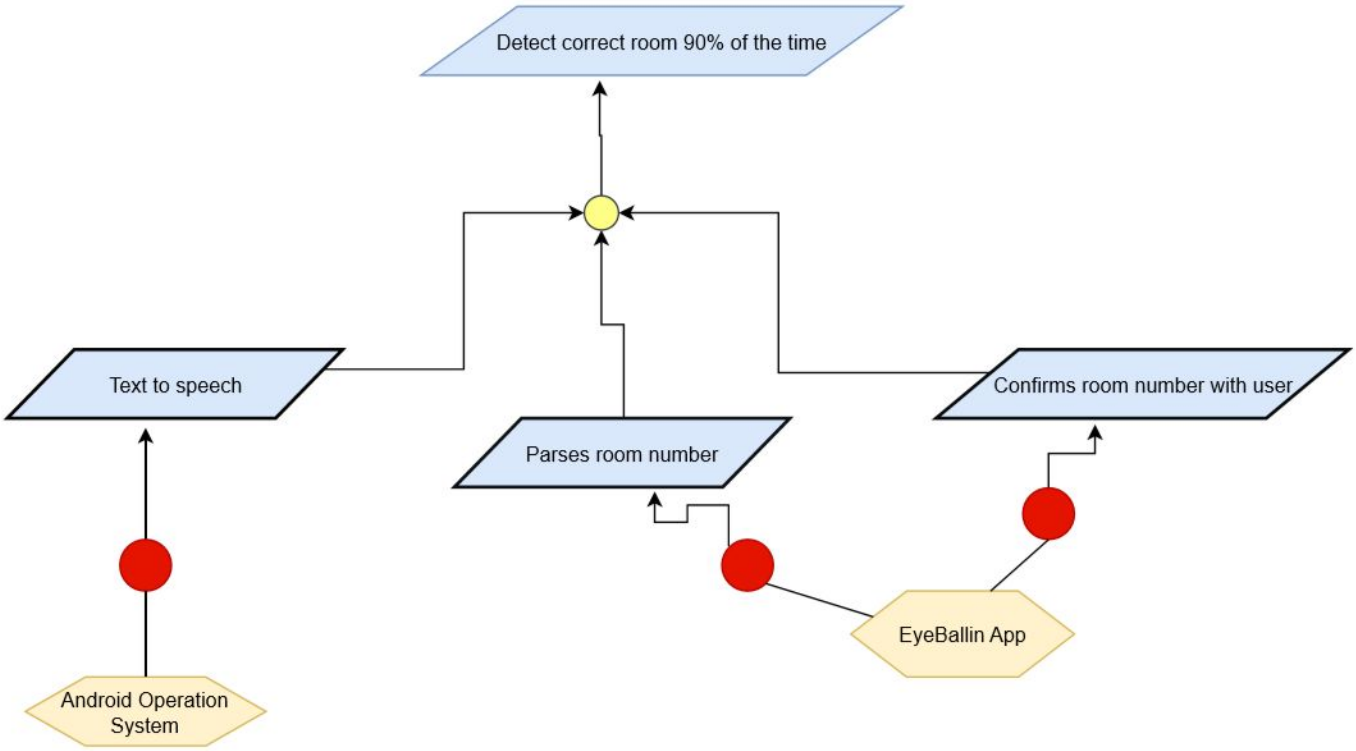

Figure 4.1.2.2: ***Detect correct room 90% of the time.***

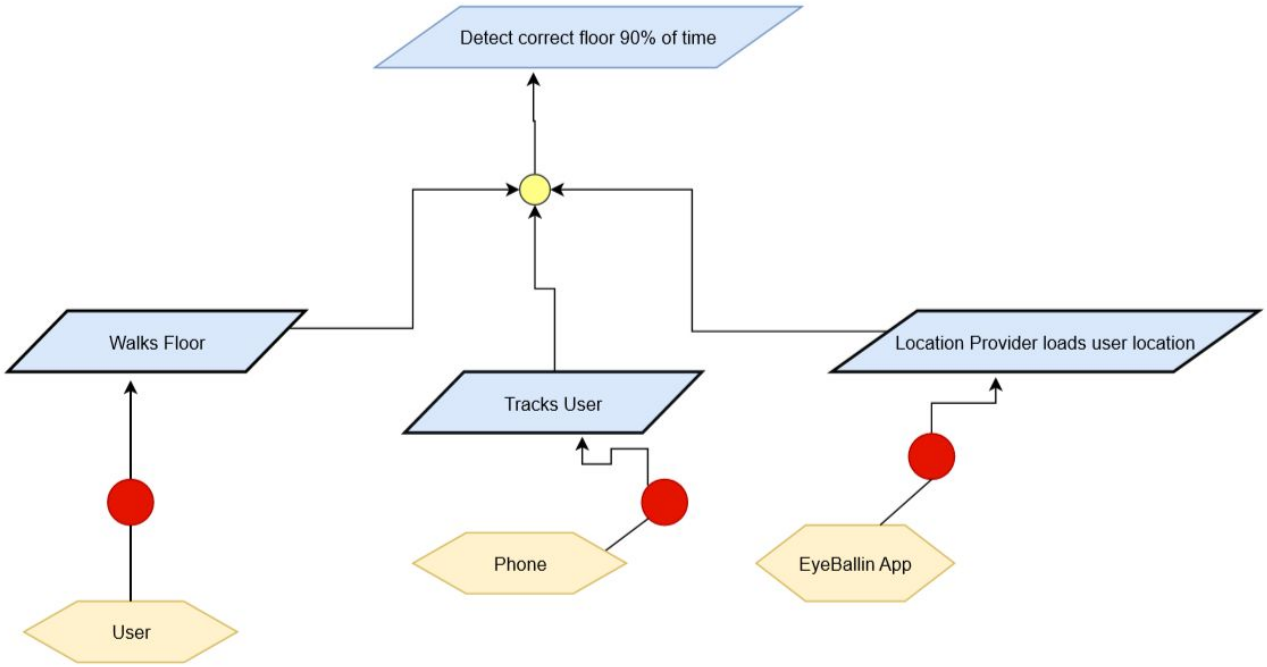

Figure 4.1.2.2:1 ***Detect correct floor 90% of the time.***

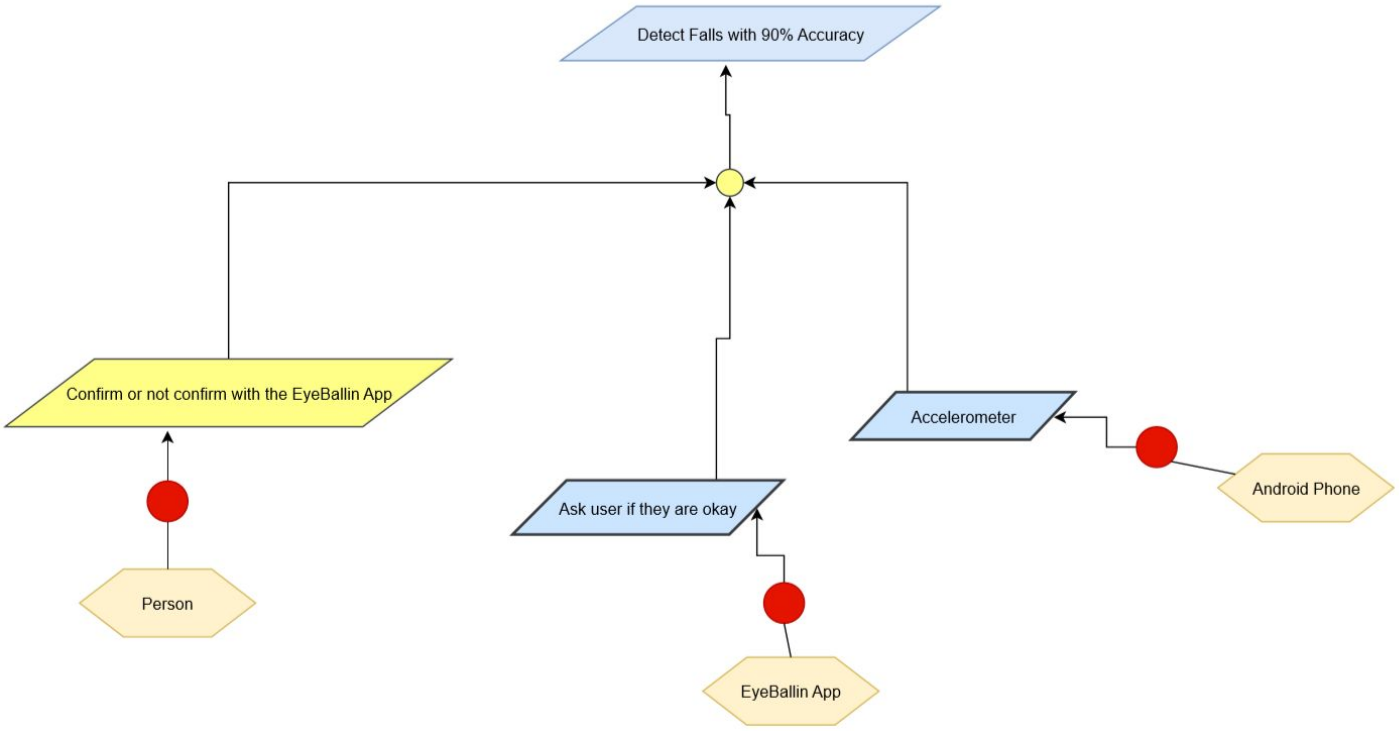

Figure 4.1.2.3 Detect Falls with 90% Accuracy.

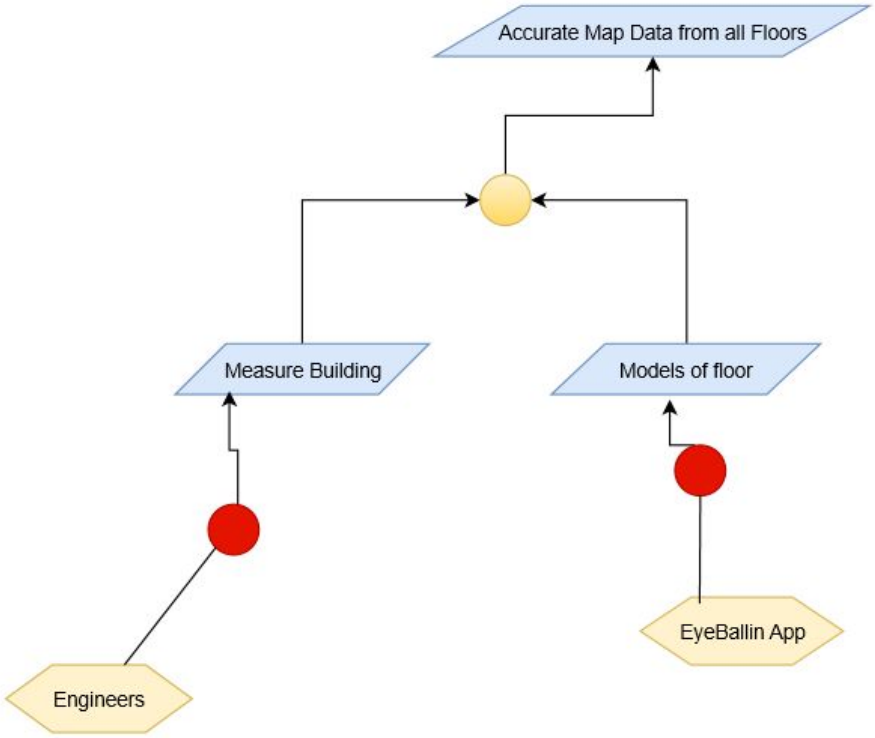

Figure 4.1.2.4 Internal Representation of WSU Everett floors.

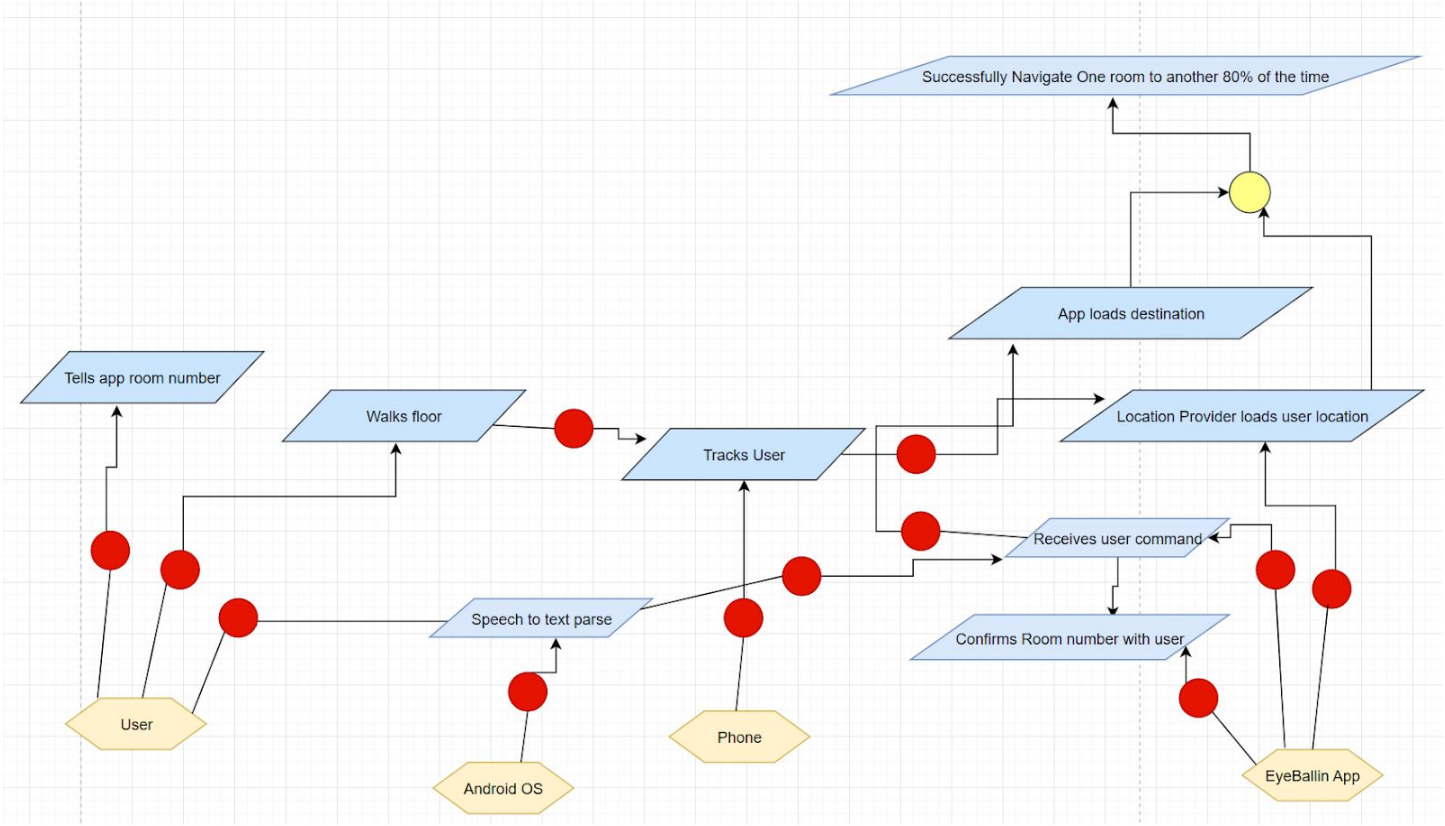

Figure 4.1.2.5 Successful navigation from one room to another 80% of the time.

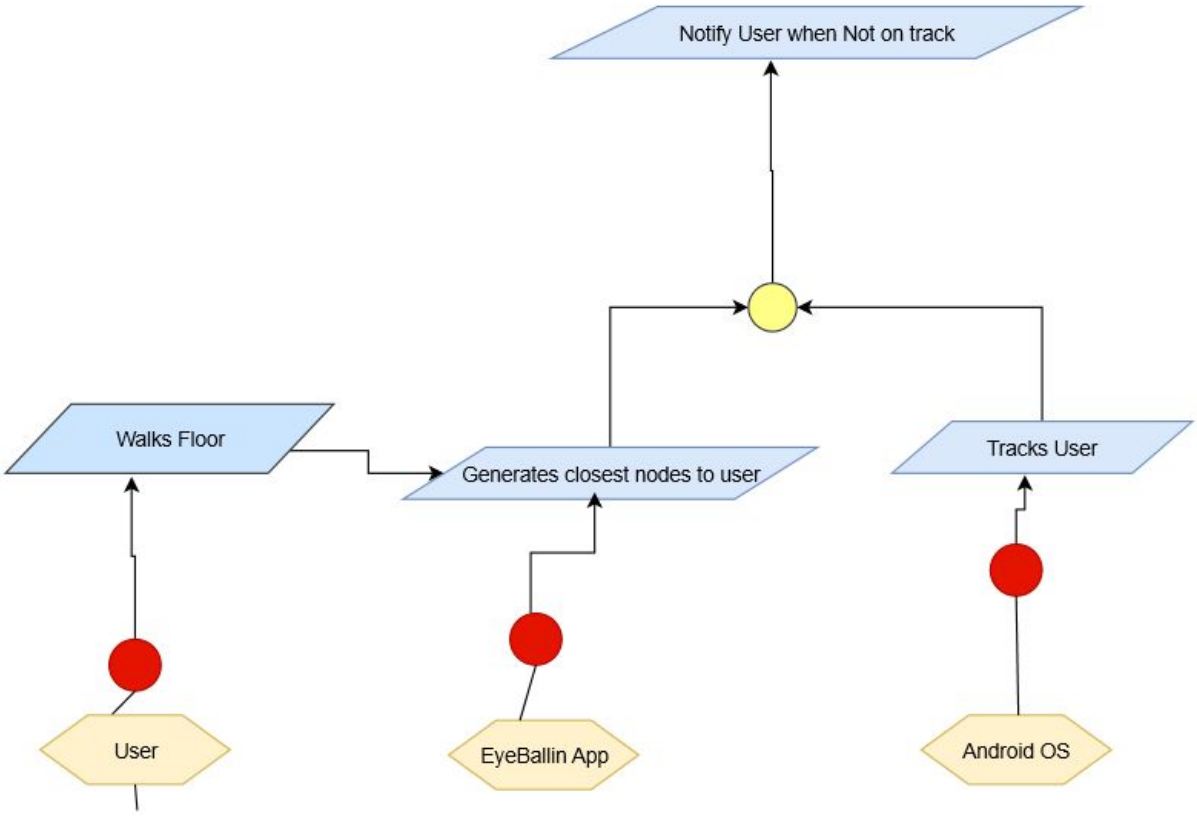

Figure 4.1.2.6 Notify user when not on track

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 the WSU Everett building.
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 detect falls, clarify with the user if they need help, and notify building security.	P4	G4
IFRO2	The system shall store an internal representation of the WSU Everett building, excluding faculty offices.	P6	G1, G6
IFRO3	The system shall interpret user's voice into text, and output text into speech.	P2	G2
IFRO4	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 5 feet of 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 objectively clear instructions.	P2, P5	G2, G5

4.2 RS

4.2.1 Functional Requirements

FR ID	**Description**
FR1	The system extends the android Location class and uses an internal adjacency matrix for mapping functionalities.
Satisfies Functional Requirement Issue	FRI1
Satisfies Objectives	IFRO2, INFRO2, INFRO3
Satisfied by prototype feature	Custom location mapping system
FR2	The application is able to receive input from the user and respond according to their commands
Satisfies Functional Requirement Issue	FRI2, FRI3
Satisfies Objectives	IFRO1, IFRO3, 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 input
Satisfies Functional Requirement Issue	FRI2, FRI3
Satisfies Objectives	IFRO1, IFRO3, INFRO4, INFRO5
Satisfied by prototype feature	Voice Interaction System
FR4	The application is able to detect when a user falls and notify the building security in case of an emergency
Satisfies Functional Requirement Issue	FRI5, FRI6
Satisfies Objectives	IFRO1
Satisfied by prototype feature	Fall Detection System
FR5	The application is able to simulate where a user is within the building using custom building coordinates and simulate the user's movement.
Satisfies Functional Requirement Issue	FRI1, FRI4, FRI7
Satisfies Objectives	IFRO2
Satisfied by prototype feature	Custom location mapping 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	IFRO2, IFRO3, IFRO4, INFRO1, INFRO2, INFRO3, INFRO4, INFRO5
Satisfied by prototype feature	Voice Interaction System

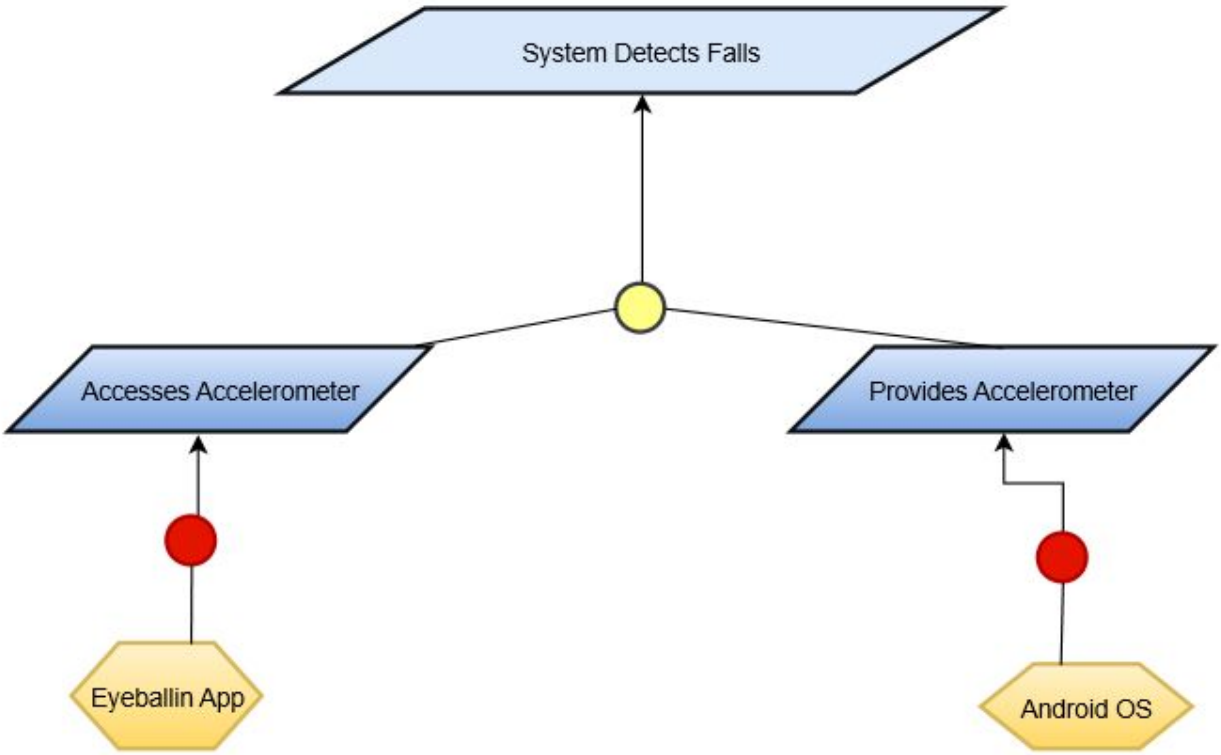

Figure 4.2.1.1: Functional Requirement System Detects Falls

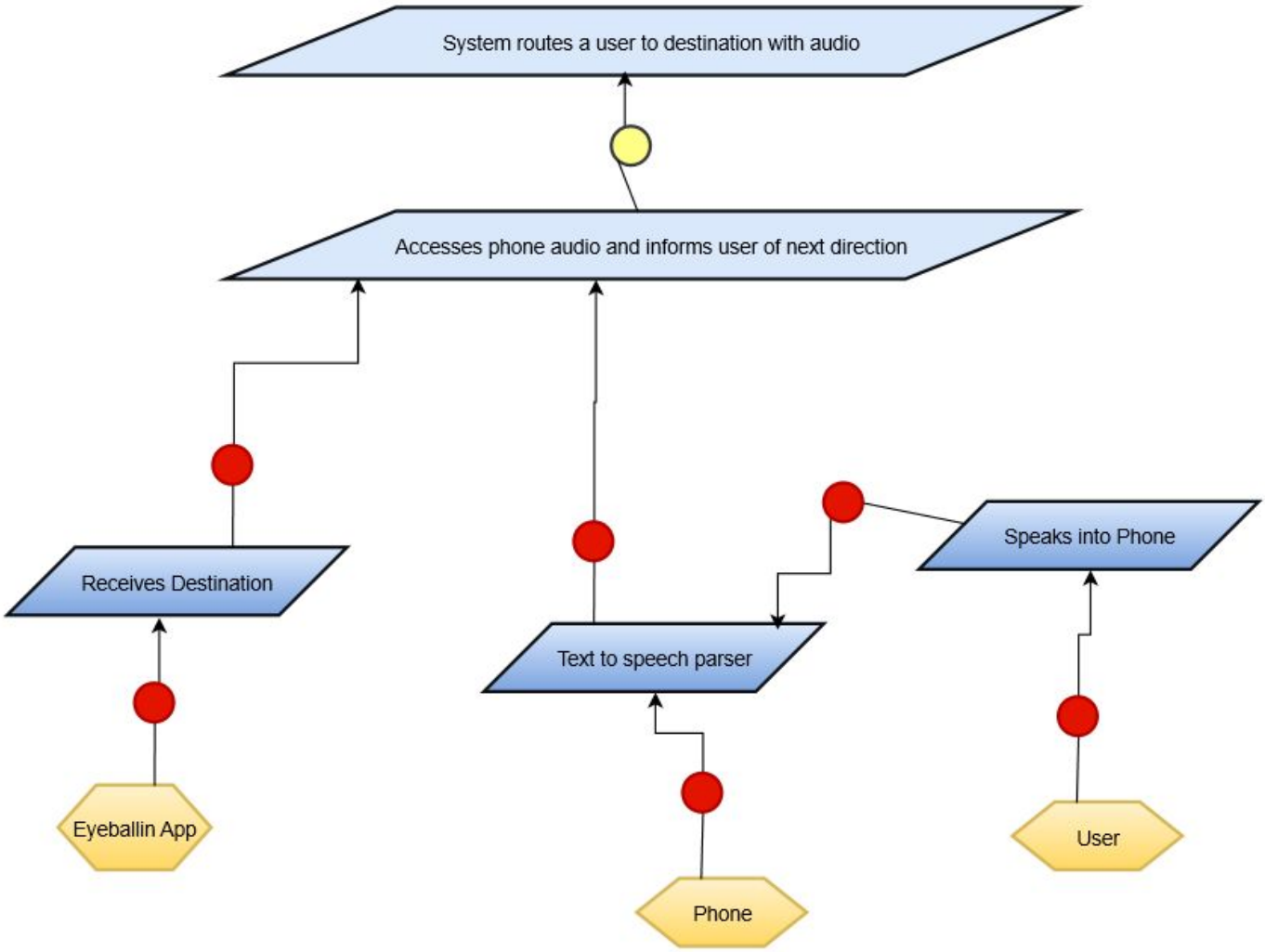

Figure 4.2.1.2: Functional Requirement System Routes User

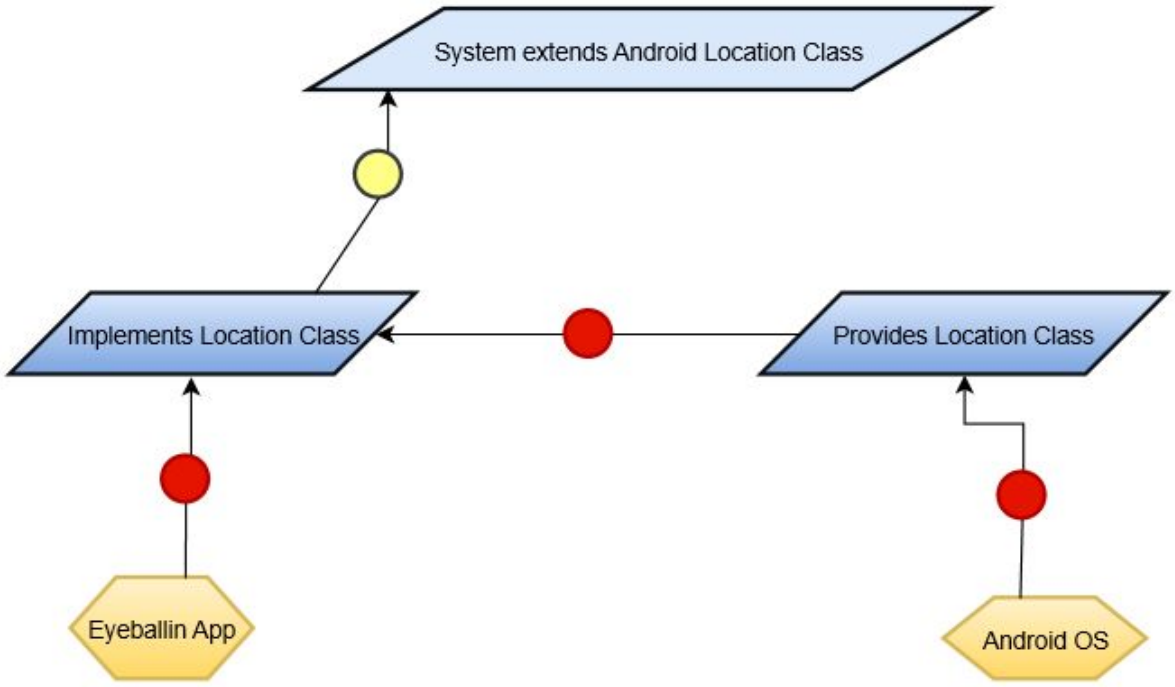

Figure 4.2.1.3: Functional Requirement System Extends Location Class

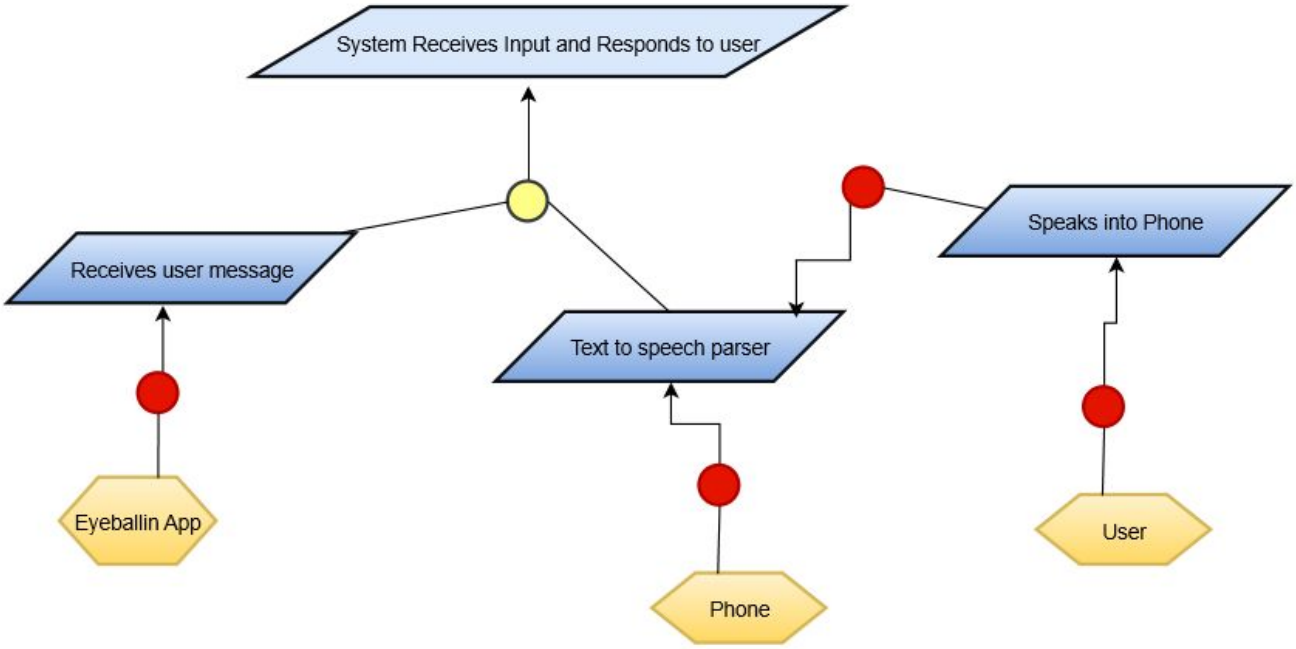

Figure 4.2.1.4: Functional Requirement System Receives Input and responds to user

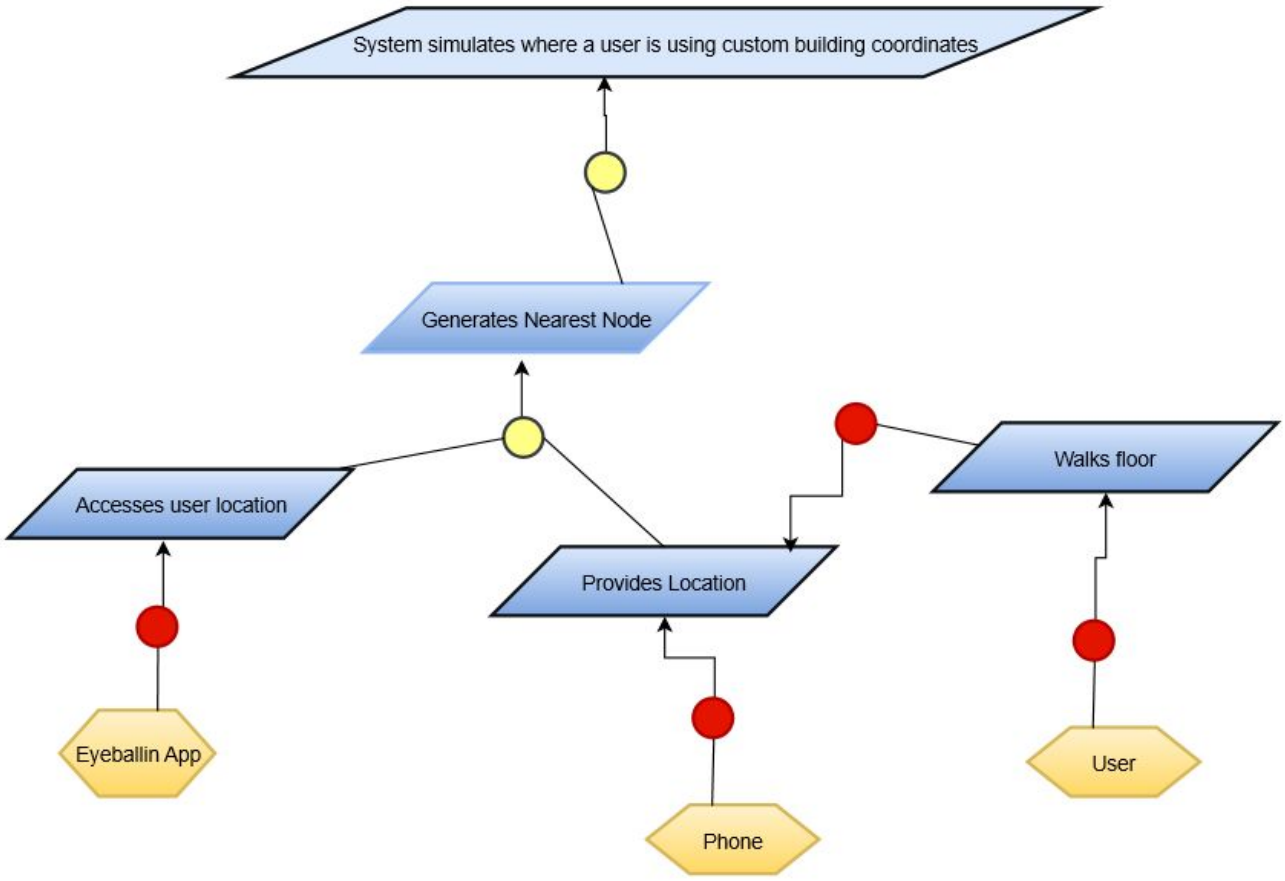

Figure 4.2.1.5: Functional Requirement System simulates where a user is using custom building coordinates.

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	The system shall generate accurate and safe paths
	OFR2	The system shall generate only one path to ensure reliability
	OFR3	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	IFRO4, INFRO5	
Satisfied by prototype feature	Internal map of half of the WSU Everett building (No offices included)	
NFR2	The system must perform in close to real time.	
Operationalized Functional Requirements	OFR4	The system must generate accurate walk paths in a matter of seconds
	OFR5	The system must recalculate in a matter of seconds after the user takes a wrong turn
	OFR6	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	IFRO4, INFRO5	
Satisfied by prototype feature	Internal map of WSU Everett building	

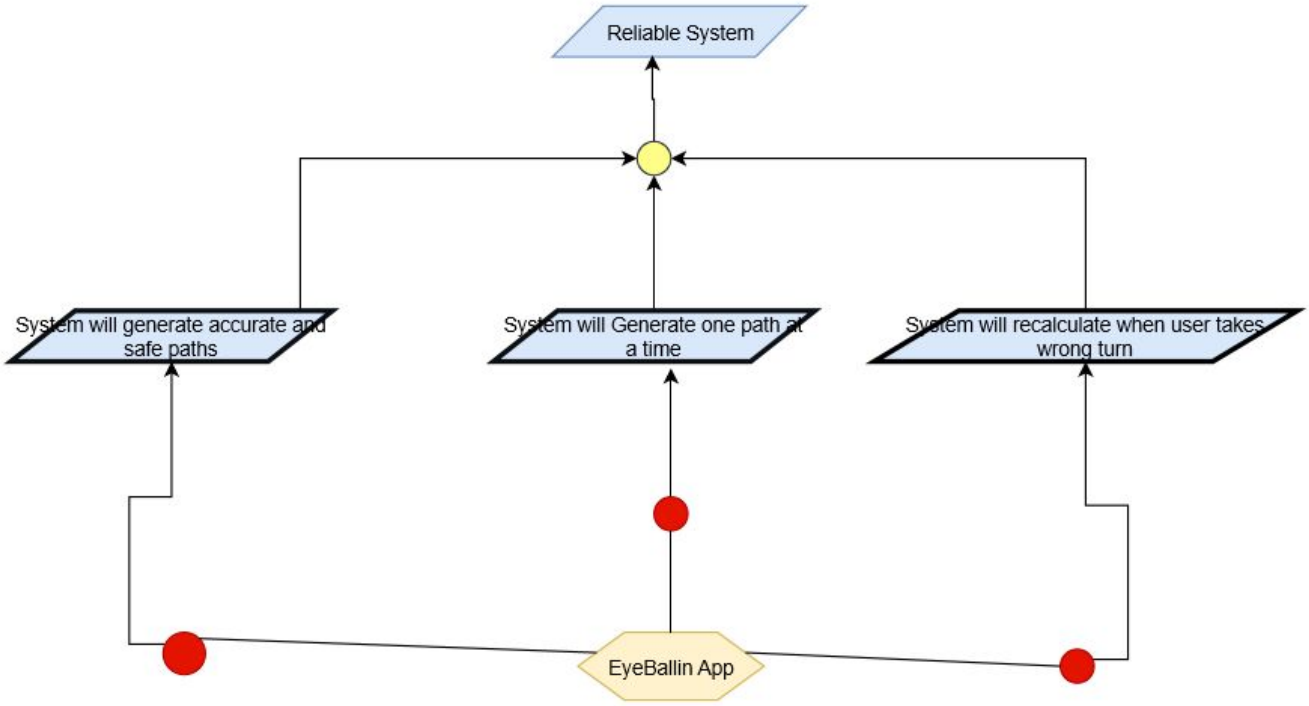

Figure 4.2.2.1 Reliable system non-functional requirement.

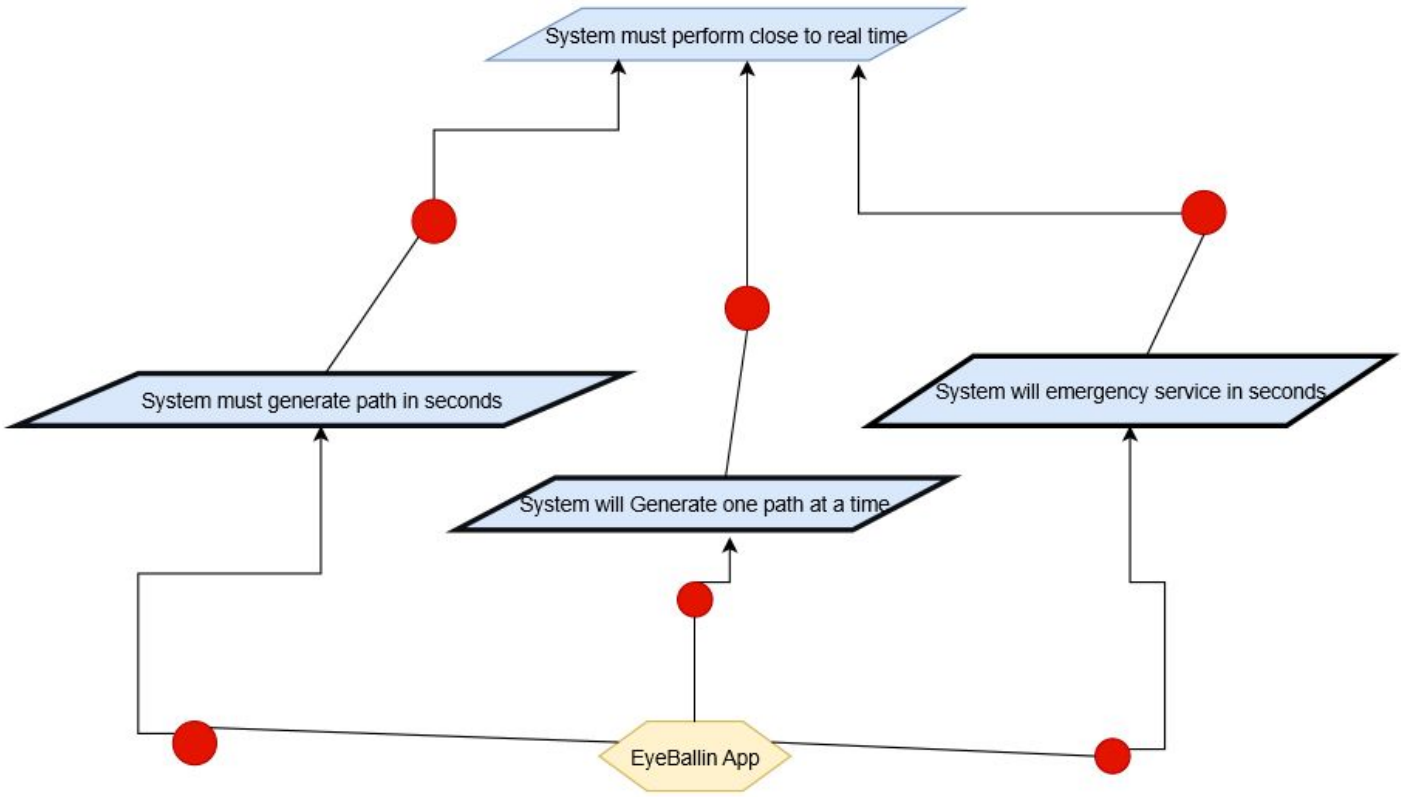

Figure 4.2.2.2 System must perform in close to real time.

4.2.3 Specifications

Functional Specification ID	Functional Requirement
FS1	The system uses a custom internal representation of half of the WSU Everett building floorplan. The app will only navigate users to classrooms, bathrooms, seminar rooms, and main entrances (North and South)
Satisfies Functional Requirement Issue	FRI1
Satisfies Objectives	IFRO2
Satisfied by prototype feature	Custom location mapping system

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	FRI2
Satisfies Objectives	IFRO3, 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	FRI3
Satisfies Objectives	IFRO3, 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 the building security via text.
Satisfies Functional Requirement Issue	FRI5, FRI6
Satisfies Objectives	IFRO1
Satisfied by prototype feature	Fall Detection System
FS5	The system is able to accurately simulate where the user is within the building using a custom coordinate system and an extended use of the Android Location class. Floor numbers are known because the user will always start on floor 1.
Satisfies Functional Requirement Issue	FRI4

Satisfies Objectives	INFRO1
Satisfied by prototype feature	Custom location mapping system
FS6	The System can route the user through the building using a combination directions that include the direction and distance needed, along with possible directions for what buttons to press in an elevator.
Satisfies Functional Requirement Issue	FRI7
Satisfies Objectives	IFRO3, IFRO4, INFRO1, INFRO2, INFRO3, INFRO4, INFRO5
Satisfied by prototype feature	Voice Interaction System

5. Prototype Interface Mock-ups

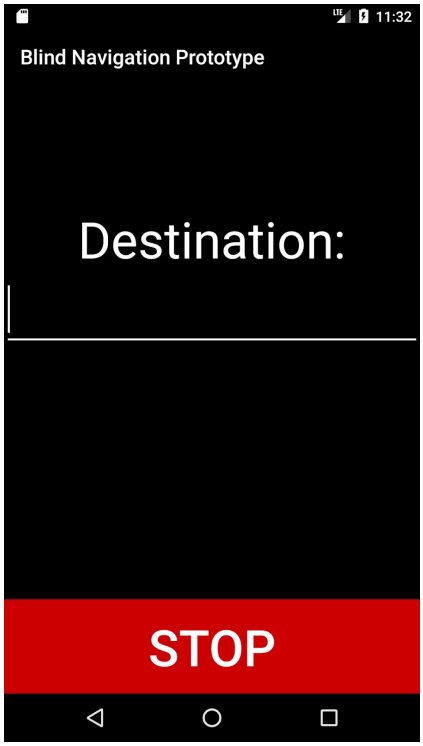

Figure 5.1: Home Activity

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.

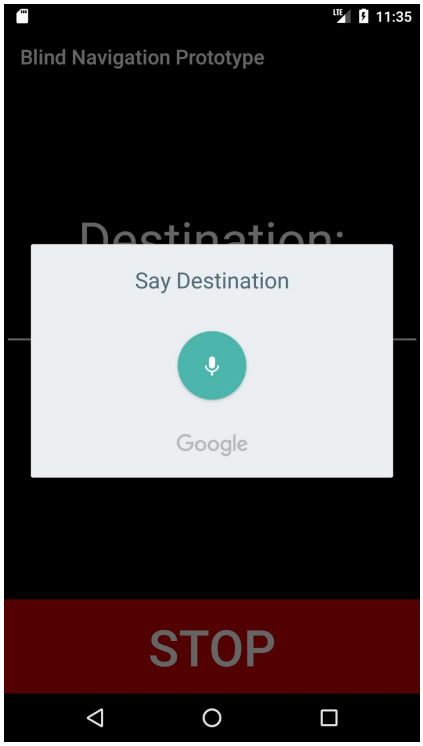

Figure 5.2: Destination Prompt

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.

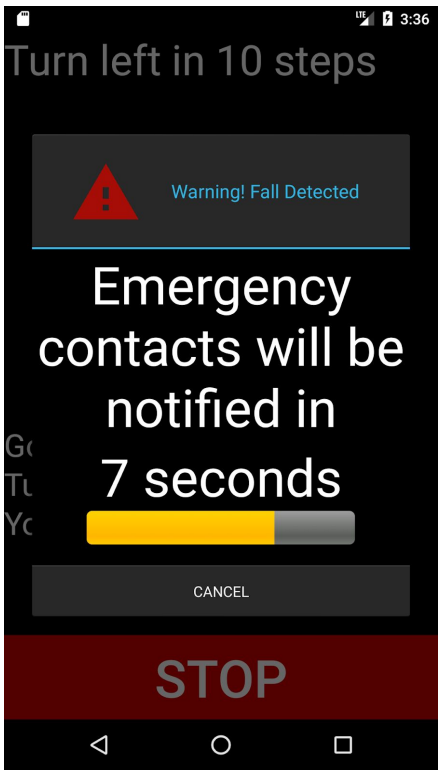

Figure 5.4: Fall Detection

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.

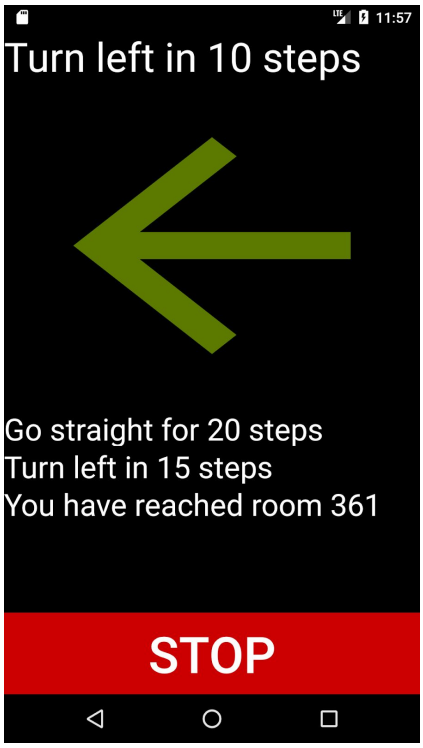

Figure 5.5: Navigation

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.

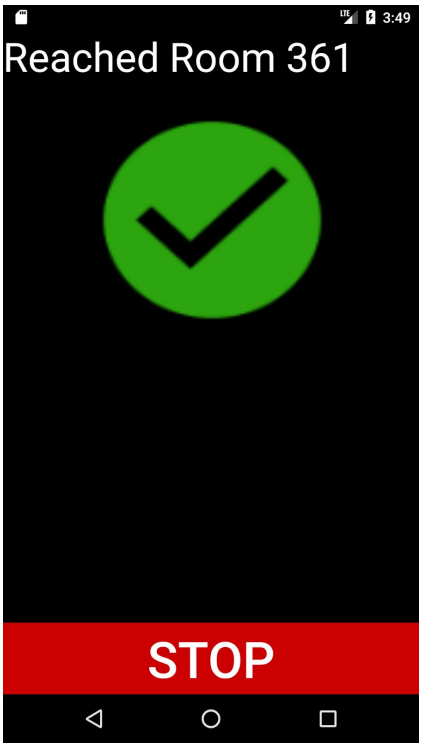

Figure 5.6: Destination Reached

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?”

5. 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 building security. 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.