TANavApp / TANA
World Requirement Specification

October 23, 2023

Alex Georgopoulos, Ambrose Sturgill, Brennan Duffy, Tommie Moore

github.com/bd150/cis201proj

Table of Contents

1 Introduction

1.1 Project Overview

The TANav App (TANA) is a user-friendly navigation app developed to assist visually impaired individuals when navigating within the Theater Arts building of Cal Poly Humboldt. The TANA will use indoor mapping to provide step-by-step directions and personalized routes to empower users with a sense of confidence when navigating between classes in the Theater Arts building.

1.2 Version Changes

Revision	Date	Description	Author
1.0.0	9/26/23	Create initial document	Alex, Brennan, Ambrose
1.1.0	10/3/23	Add to sections 2.2 and 2.3	Alex, Brennan, Ambrose
1.2.1	10/17/23	Complete section 2.2 and 2.3, begin on sections 1 and 3	Alex, Brennan, Ambrose
1.2.2	10/18/23	Add a project overview and fill in the improved understanding of domain, world, and requirements	Alex
1.2.3	10/19/23	Complete section 2.1, section 4, and section 5	Alex, Ambrose, Brennan
1.2.4	11/2/23	Modify title	Alex, Ambrose, Brennan
1.2.5	11/27/23	Standardize Version Changes table Implemente Table of Contents Update requirements	Brennan

2 Issues with Preliminary Definition Given

2.1 Issues with the Domain, Stakeholders, Function and Non-Functional Objectives

1. Domain is not defined to be within the Theater Arts Building (A1)
    a. Description:
        i. Initially the Navigation area was defined as the Cal Poly Humboldt campus, but has since been updated by stakeholders to be confined to the Theater Arts building for the initial prototypes.
    b. Potential Resolutions:
        i. Update the location of the domain to reflect the change in stakeholder needs
        ii. Keep the location domain the same
    c. Rationale:
        i. We chose resolution i because it more closely applies to the current scope of the project.
2. Primary stakeholders does not specify students (A2)
    a. Description:
        i. The initial domain suggested that the app would be used by blind people in general to navigate the campus, however our analysis suggests it is students who will be the main user of the app to navigate to their designated classrooms.
    b. Potential Resolutions:
        i. Update the primary stakeholder to reflect the target user of the app
        ii. Expand the target user of the app to reflect the original primary stakeholder definition
    c. Rationale:
        i. We have decided to update the primary stakeholder to better align with stakeholder interest and allow design decisions to better accommodate the primary stakeholders.
3. Functional definition conflicts with revised domain (A3)
    a. Description:
        i. The original definition describes the usage of the app as, “going from one location to another location in the same or different buildings”, however this conflicts with the location domain of the TA building.
    b. Potential Resolutions:
        i. Update functional definition to reflect the location domain
        ii. Update the location domain to reflect the original functional definition
    c. Rationale:
        i. We have chosen to follow resolution i and update the functional definition, as it more closely applies to the current scope of the project.

4. Comfortability has been removed as a Non-Functional Requirement (A4)
    a. Description:
        i. Our team has decided to not include comfortability as a necessary Non-Functional requirement.
    b. Potential Resolutions:
        i. Remove comfortability from the listed set of non-functional requirements in the objective
        ii. Re-introduce comfortability as a non-functional requirement.
    c. Rationale:
        i. We have decided to follow resolution i, because comfortability conflicted with the non-functional requirement agility and has been deemed not as important as the non-functional requirement safety, which will be applied in a similar manner.
5. Utilizing features of advanced smartphone is unnecessary (A5)
    a. Description:
        i. Despite the original objective's wording, more sensors do not necessarily lead to a more helpful product. Adding sensors takes developer resources. As more sensors are implemented, the relative help they provide gets diluted, resulting in diminishing returns. Our team will need to decide how important this objective is to the project. Additionally, this objective conflicts with requirement 3.2.2.6.
    b. Potential Resolutions:
        i. Remove this functional / non-functional objective entirely.
        ii. Reduce the severity of the statement to suggest that using more sensors may increase the helpfulness of the project.
    c. Rationale:
        i. We have chosen to apply resolution i. This objective is not in line with our project goals.

2.2 Issues with Software System Requirements: Functional Requirements

1. Process for determining user location is undefined (FR-1)
    a. Description:
        i. There are many approaches to determining the user's location with varying degrees of precision/reliability. We need to decide which approach is most appropriate for our system.
    b. Potential Resolutions:
        i. Implement GPS tracking for the phone to identify where the user is
        ii. Allow the user to choose their initial location from a given list of possibilities
        iii. Allow the user to narrow down their location via landmark information (room number, stairs, elevator, etc.)
    c. Rationale:
        i. We chose options ii and iii because they avoid the pitfalls of GPS, notably disconnection and accuracy issues. GPS can be unreliable in buildings and for small distances. Our app will need to aid users in navigating varying distances indoors, so GPS isn't a good choice. The combination of resolutions ii and iii allow users to navigate the building without needing an internet connection. We also believe this approach would be quicker to implement into the system than connecting it to GPS.
2. Process for communicating directions to the user is undefined (FR-3)
    a. Description:
        i. There are many ways to communicate to the user important direction information, such as how far they need to walk and when to turn. We need to determine how the app will communicate these directions to the user.
    b. Potential Resolutions:
        i. Calculate the route using a given stride length, and tell the user how many steps to take
        ii. Calculate the route using a unit measurement, such as meters or feet, and tell the user how far to travel when given a new direction
        iii. Calculate the route using a unit measurement, and give the user frequent updates on how far away they are to the next node in the path
        iv. Calculate the route using a given stride length, and have the user tap a button that counts the number of steps they have taken
    c. Rationale:
        i. Because calculating distance using methods available in a smartphone would not be sufficient in the small scale of a building, we have decided to go with a combination of options iii and iv, where the phone would track distance using a step counter, while also giving frequent updates to the user of how far they have to travel to the next point in their route.

3. Insufficient technology for obstacle detection (FR-6)
    a. Description:
        i. We cannot expect all users to have advanced sensors on their phones for detecting obstacles, but the app still needs to help users avoid obstacles. We need to decide which method of detection would work best for the app and intended users.
    b. Potential Resolutions:
        i. Remove requirement
        ii. Soften requirement
        iii. Require users to use external tools for detecting obstacles (e.g. cane)
    c. Rationale:
        i. Option iii is the safest approach to this problem. Since the system won't be able to reliably detect obstacles, requiring that users use an external tool provides them with sufficient protection from obstacle collision.
4. Unnecessary emergency calls when the system cannot determine its location (FR-7)
    a. Description:
        i. Calling an emergency contact or emergency services when the system cannot determine its location is an unnecessary requirement for this application. Since the project scope was reduced to the TA building, users can determine their location without the use of the app via braille signs, meaning that an emergency call would no longer be necessary.
    b. Potential Resolutions:
        i. Don't make emergency calls when the system cannot determine its location
    c. Rationale:
        i. The system shouldn't make unnecessary emergency calls.
5. Difficulty in accurately identifying a "fall" (FR-7)
    a. Description:
        i. Accurately identifying when the user has fallen is a difficult task that runs the risk of false positives and/or false negatives. If the user sits down or drops their phone, an emergency call would be inappropriate.
    b. Potential Resolutions:
        i. Include an emergency button that a user can press in the case of an emergency
        ii. Include an "emergency mode" that activates if the app detects a fall, and declares an emergency if no action is taken within 30 seconds
        iii. Use voice detection to listen for key words, such as "I have fallen" to activate an emergency mode
    c. Rationale:
        i. Resolution ii solves this problem by allowing for an automated fall detection system while accounting for false positives and letting users cancel the emergency mode before making contact to emergency services.. While we recognize this resolution could be annoying if too

many false positives are tripped, some sort of tradeoff is necessary if we want to have effective automated fall detection.

6. Prediction Method (FR-8)
    a. Description
        i. Accurately predicting which destination the user plans to go could be difficult, and getting a prediction wrong would be inconvenient for a user that cannot visually detect which route the system is taking them on.
    b. Potential Resolutions:
        i. The app could keep a history of users' routes, and suggest to them the route they previously had taken after arriving at their current destination.
        ii. The app could have a route planner, which allows the user to input a series of routes, such as their daily schedule, which the app could read in order.
        iii. Remove the prediction requirement
    c. Rationale:
        i. We chose resolution ii because this allows users to program their own set of routes into the app, without having intrusive predictions that may be inaccurate.

2.3 Issues with Software System Requirements: Non-Functional Requirements

1. The steps towards safety are not defined (NFR-1)
    a. Description:
        i. NFR-1 states that the app should help users navigate safely indoors without specifying how the app will do so.
    b. Potential Resolutions:
        i. Define safety as “minimizing obstacles along user’s route”
        ii. Define safety as “selecting safe paths and directing users down this path safely”.
        iii. Use the Oxford Dictionary definition of safety: “The state of being protected from or guarded against hurt or injury; freedom from danger.”
    c. Rationale:
        i. We chose resolution ii because it gives “safety” a more specific definition that is relevant to our project.
2. Conflicting goals of “comfortability” and “fastest” in determining the route (NFR-2, NFR-3)
    a. Description:
        i. Finding the quickest route may conflict with finding the most comfortable route. For example, taking a detour to the elevator in order to avoid stairs may take much longer than simply taking the stairs. We need to determine how our system will prioritize comfort and speed to create a path for the user.
    b. Potential Resolutions:
        i. Prioritize speed, suggesting the most direct route for the user to take to reach their destination
        ii. Prioritize comfort, providing a route that avoids obstacles even when that route would be longer than the direct route
        iii. Give the user the choice between the first two options
        iv. Determine which routes meet a set comfortability threshold and provide the fastest route of the options.
    c. Rationale:
        i. We will prioritize the most direct route, because we should not be suggesting overtly dangerous routes through the building, so there would be little benefit to suggesting a significantly slower route.
3. Usability is a vague descriptor (NFR-4)
    a. Description:
        i. We must define “usability” and determine which functions this requirement applies to.
    b. Potential Resolutions:
        i. Define usability as “including methods of access which allow a blind user to interact with the app”
        ii. Use the Oxford dictionary definition of usability: “The fact or quality of being usable.”
        iii. Apply usability to all functions of the application

        iv. Apply usability to only critical functions of the application, such as route setting and emergency functions
        v. Remove the requirement of “usability”
    c. Rationale:
        i. Our team chose i and iii to emphasize the system’s usability for blind people throughout the application.

4. Which features should be customizable (NFR-6)
    a. Description:
        i. Customization could be applied to many aspects of the app, but could be costly to implement and unnecessary for blind users
    b. Potential Resolutions:
        i. Remove the requirement of customization
        ii. Keep customization options already in the requirement (volume, speech speed/interval)
        iii. Allow customization of button layout/functionality
        iv. Allow customization of aesthetics, such as button color
        v. Allow language customization, both visual and audible
    c. Rationale:
        i. We chose ii and v as they provide the user with a substantial suite of options without presenting them with needless options. We considered option iii as it would allow users to better fit the layout to their preference, but since the app doesn’t currently need more than one button, it would be a moot customization option. We will review this requirement as the system grows.
5. Customizable features are loosely defined (NFR-6)
    a. Description:
        i. Customizable features should be defined concretely without any ambiguity.
    b. Potential Resolutions:
        i. Remove the “etc.” in the requirement
    c. Rationale:
        i. By choosing resolution i and removing the “etc.”, the list is cleared of any ambiguity.
6. How will the app determine “needs of the user” (NFR-7)
    a. Description:
        i. The app should be modular to meet needs of the user, but there is no specified way for these needs to be known by the app
    b. Potential Resolutions:
        i. Needs of the user should be determined by the user and entered as “customizable” options. This aspect of the NFR should be moved to NFR-6.
    c. Rationale:
        i. Resolution i is appropriate since it moves user choice into the customizability requirement.

7. How will the app accommodate a user with a low-spec phone (NFR-7)
    a. Description:
        i. Certain features of the app will need advanced hardware, such as motion sensors, high resolution cameras, and GPS capability. How should the app accommodate users without these features?
    b. Potential Resolutions:
        i. Relax the requirement of modular to not include low-spec phones
        ii. Relax other requirements to not use advanced features
        iii. Determine the capabilities of the user's phone and block features that require better hardware
        iv. Determine the capabilities of the user's phone and use simplified features that could work on their current hardware
    c. Rationale:
        i. Looking at previous issues, we've gone with an approach that is congruent with ii. No GPS, no cameras, just user input via the screen/microphone.

3 Improved Understanding

3.1 World

3.1.1 Problems

3.1.1.1 Problem: Visually impaired individuals have problems navigating complex spaces

3.1.1.2 Problem: Visually impaired individuals have difficulty planning a route with provided navigation equipment (e.g. maps)

3.1.1.3 Problem: Visually impaired individuals can have problems avoiding obstacles

3.1.1.4 Problem: Asking for instructions can be difficult for visually impaired individuals

3.1.1.5 Problem: Some pathways, such as stairs, may be uncomfortable for visually impaired individuals to navigate

3.1.1.6 Problem: Some routes may take blind students too long to navigate

3.1.2 Goal

3.1.2.1 Goal: Blind students need to navigate to the correct room

3.1.2.2 Goal: Blind students need to reach their class on time

3.1.2.3 Goal: Blind students need to reach their destination without running into obstacles

3.1.3 Improved understanding of The Domain, Stakeholders, Functional and Non-Functional Objectives

3.1.3.1 Domain: Our app will be made for use within the Cal Poly Humboldt Theater Arts building. Destinations within the TA building include classrooms, offices, sound stage, and restrooms. These rooms are distributed among the first and second floor of the TA building. Between rooms are hallways, stairways, and elevators. There may also be obstacles such as students and bins that could obstruct certain pathways.

3.1.3.2 Stakeholders: The primary stakeholder of TANA would be visually impaired individuals, including blind students. Secondary stakeholders include caretakers, CDRC (Campus Disability Resource Center) staff, and emergency service people, such as UPD officers. Other stakeholders include the developers and the University.

3.2 Requirements

3.2.1 Improved Understanding of Functional Requirements

3.2.1.1 The system shall allow users to input their destination from a list of predefined rooms and landmarks.

3.2.1.2 When given a destination, the system shall identify and locate the nearest bluetooth beacon.

1. If a nearby beacon cannot be identified, the system shall notify the user that a beacon could not be found.

3.2.1.3 When the nearest beacon is identified, the system shall determine a path from the user's location to the user's destination.

3.2.1.4 When a path has been determined, the system shall communicate to the user the distance to the next node in the path.

3.2.1.5 Before the user has reached a node along their path, the system shall alert the user so that they can stop at the right place to turn.

3.2.1.6 The system shall detect when the user falls.

1. When a user fall is detected, the system shall enter an "emergency state" that can be canceled by the user.
2. When a user fall is detected, the system shall attempt to call an emergency contact.
3. If the system fails to reach an emergency contact, the system shall attempt to call the campus police.
4. If the user cancels the "emergency state", the system shall cancel any outbound calls.

3.2.1.7 The system shall suggest routes based on a predefined schedule planner once their current destination has been reached.

3.2.2 Improved Understanding of Non-Functional Requirements

3.2.2.1 The system shall select safe paths and direct users to their destination safely.

3.2.2.2 The system shall prioritize the fastest route to the user's destination.

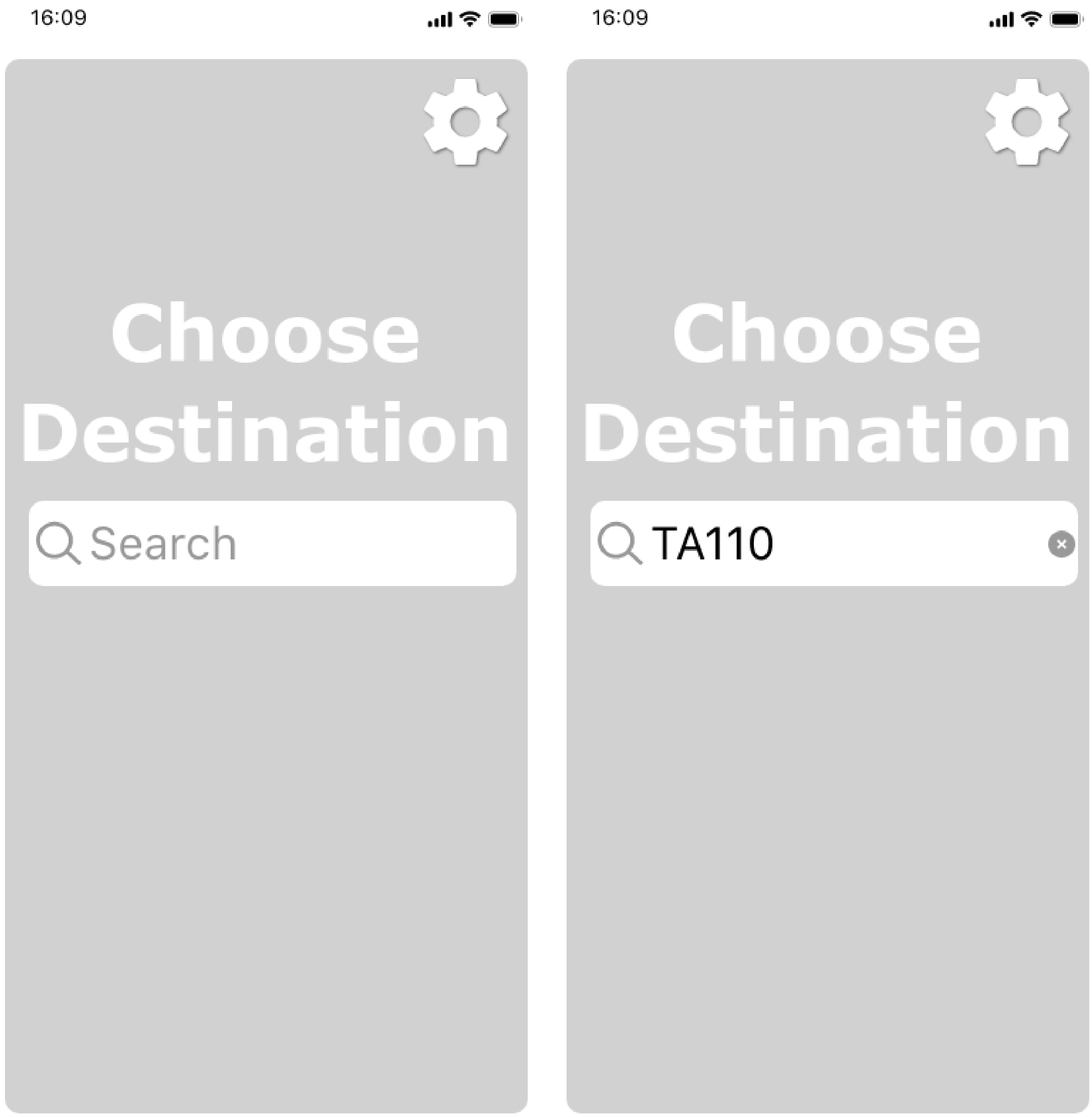
3.2.2.3 The system shall include methods of access which allow a blind user to interact with all functions of the application.

3.2.2.4 The system shall be available for use at all times.

3.2.2.5 The system shall allow customization options for volume, speech interval, and speech language to meet the needs of the user.

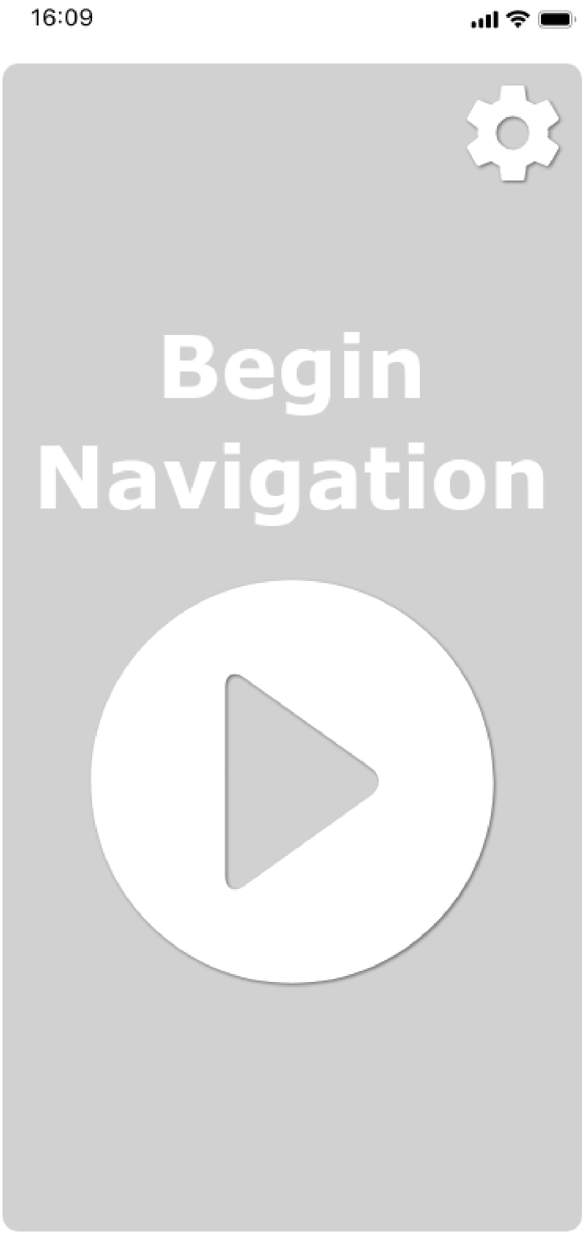
3.2.2.6 The system shall be modular to accommodate variations in interface, language, sensors, and hardware.

4 Preliminary Prototype and User Manual

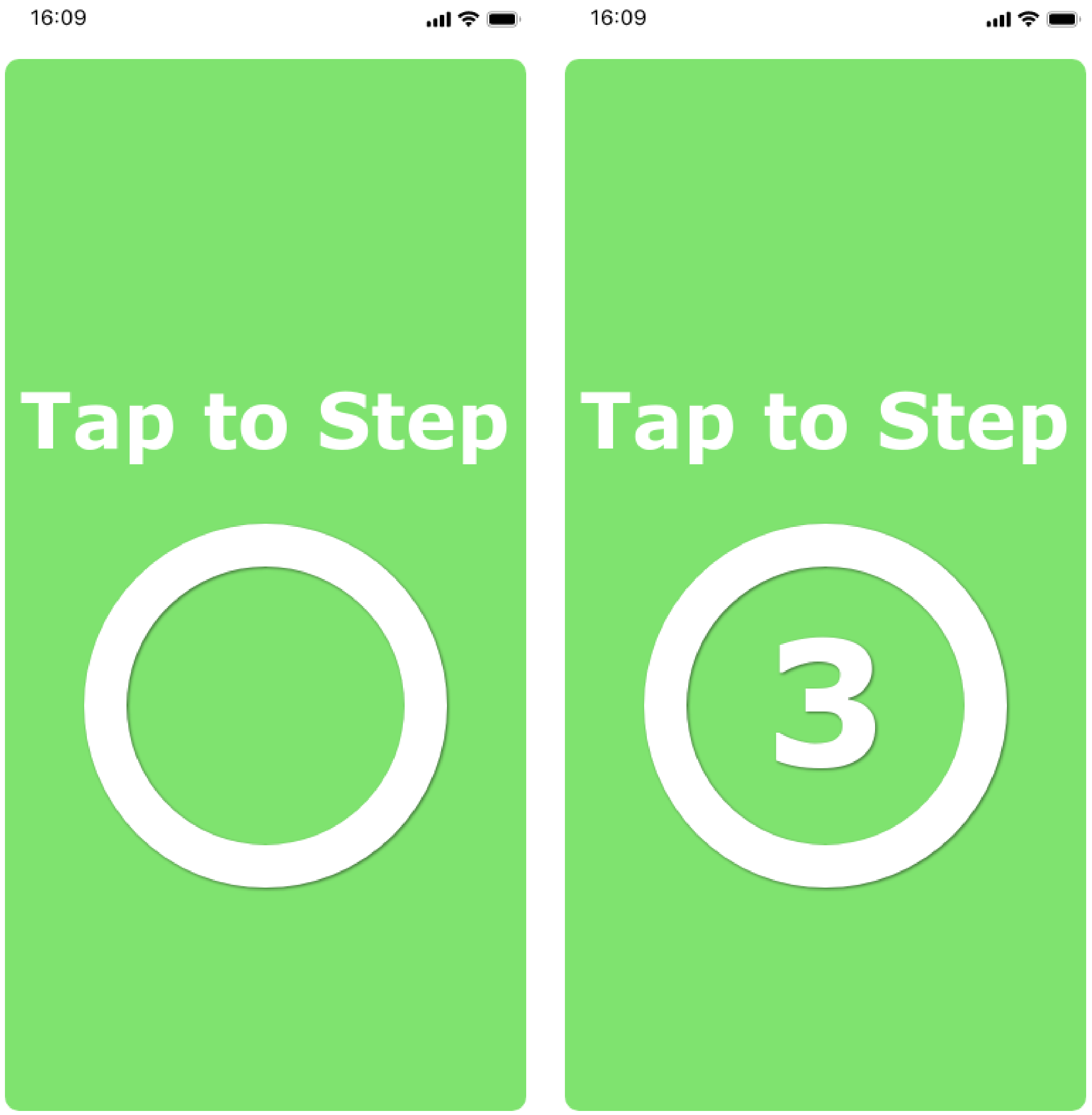


TA110

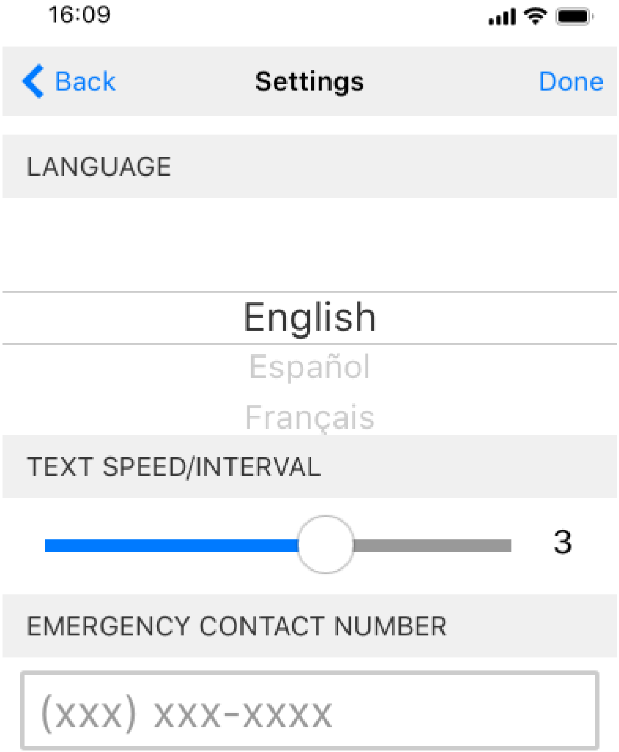
Home page. Enter your destination into the search field. App will auto-complete with available destinations. When selected, progress to the next screen.



Press the screen to begin navigation.



App will play audio instructions directing the user to take a certain number of steps before turning, using an elevator, etc.. Tapping the screen will tell the app that the user has taken a step.



LANGUAGE

English

Español

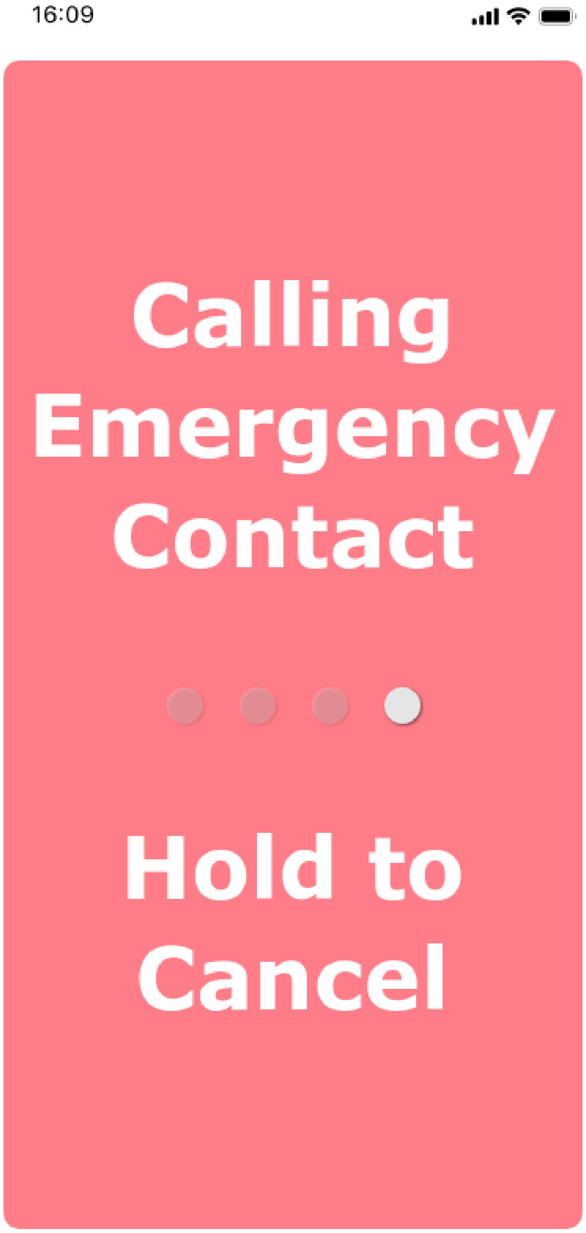
Français

TEXT SPEED/INTERVAL

3

EMERGENCY CONTACT NUMBER

(xxx) xxx-xxxx



Calling Emergency Contact

Hold to Cancel

Hitting the cog wheel will access the settings, where uses can change language, text speed, and their emergency contact number.

If the phone is dropped/detects a fall, it will enter an “emergency state” where it will wait 30 seconds before calling the emergency contact. Users can exit the emergency state by holding the screen.

5 Traceability

Requirement ID	Category	Issue ID	Source ID
3.2.1.1	Functional Requirement	2.2.1	FR-1
3.2.1.3	Functional Requirement	N/A	FR-2
3.2.1.4	Functional Requirement	2.2.2	FR-3, FR-4
3.2.1.5	Functional Requirement	N/A	FR-5
N/A	Functional Requirement	2.2.3	FR-6
3.2.1.6	Functional Requirement	2.2.4 2.2.5	FR-7
3.2.1.7	Functional Requirement	2.2.6	FR-8
3.2.2.1	Non-Functional Requirement	2.3.1	NFR-1
3.2.2.2	Non-Functional Requirement	2.3.2	NFR-2 NFR-3
3.2.2.3	Non-Functional Requirement	2.3.3	NFR-4
3.2.2.4	Non-Functional Requirement	N/A	NFR-5
3.2.2.5	Non-Functional Requirement	2.3.4 2.3.5 2.3.6	NFR-6
3.2.2.6	Non-Functional Requirement	2.3.7	NFR-7