uNav Application
Design Specification

iTeam
David Han, Russell Perry, Victoria Wellington and Rhea Arora

Table of Contents

The Design Problem 3
The Design Approach 3
Scope 4
Personas
- Persona A: Sally 5
- Persona B: Adam 5
- Persona C: Geoffrey 6

Use Cases
- Use Case A: Finding way through an unfamiliar building 7
- Use Case B: Creating a new note 8
- Use Case C: Removing a route from favorites 8

Conceptual Model 9
Design Details
- General Notes 10
- Start-up Screen 11
- Search Screen 14
- Amenities 16
- Favorites 17
- Notes 19
- Options 20
- Accessibility 22
- Font Options 23
- Call for Help 25
- Download 26
- Sound and Volume Bar 27
- Help and Tutorial 29

Brainstorming 32
Developing and Prototyping
- Storyboard 34
- Paper Prototype and User Session 36

The Final Prototype 41
High Fidelity Prototyping Constraints 41
High Fidelity User Session 42
User Testing and Heuristics
- Paper Prototype Expert User Testing 42
- High Fidelity Prototype i4Designers Testing 43

The Design Problem

One challenge blind people face is way-finding indoors. Current GPS solutions are not fine grained enough to accurately pinpoint detailed information such as where a particular office door is located, and the exact placement of building features, for example, restrooms, coffee machines, or staircases. Further, some blind people may not know what features a building even has and may thus never take advantage of them. A solution to this problem is an application that focuses on providing users with real-time, crowd-sourced indoor navigation of public spaces. A kind of "digital breadcrumbs" application, this tool will allow users to locate specific rooms, offices, restrooms, lecture halls, water fountains, etc., as updated by others using the same tool. The following design attempts to solve this design problem.

The Design Approach

The design approach in which we based many of our design details and functionality is social accessibility. This design approach focuses on the social and aesthetic aspects of the design and how they are just as important as function and usability. Our goal was to create a design that was aesthetically appealing for our target users. Our design not only focuses on the functionality of navigation but also functionality and design based in what is socially acceptable for the target user, in this case both visually impaired and non-visually impaired users. Our aim is to make the application usable and functional for both target audiences. We also based a lot of our user testing on questions that were related to consciousness while using our application. Using the feedback we got from that, we have improved our application by adding or removing features that made them interact less self-consciously. Our user's feedback towards their social experience with the application is what determined if the users needs are met. Our goal was to make sure that users don't feel self-conscious using the technology, and that our visually impaired users feel just as comfortable using the technology as our non-visually impaired users, in a social setting.

Scope

To address the problem, we created uNav, a mobile application that is developed to allow both visually impaired and non-visually impaired users to navigate indoors. uNav is a free mobile application that has been designed specifically for any version of the iPhone. The prototype is initially built as a web application on Axure to allow for rapid development and testing, but the real mobile application will be built using iOS. This application will be coded as an Apple application. It will look like any regular IOS application on the iPhone.

The application will use GPS and crowd-sources as tools to provide users with information on where they are within a building and what directions they need to take to get to a specific location within a building. uNav will give users information relevant to their current location and landmarks that are around them. Navigation for this application is only limited to public spaces (e.g. campus buildings and office buildings) as opposed to a person's private home. It will not include any information or directions for driving, public transportation or walking to/from the specific building. The application will only give directions once the user is inside a building by reading their current location within the building. The application doesn't map out their location outside the building and only shows a map of the floor plan within a building. Any location search that is not a building will be considered a location that is "not found". This application currently does not allow for users to input any information on locations details and directions, we are only making the assumption that there is sufficient amount of information already crowd-sourced into the application for the user to obtain. This is because we currently don't have the technology to make sure that all crowd-sourcing is available and accurate. This is an engineering problem that we haven't yet solved. Our prototyped application only displays a location searched as an example of how the application would interact if it were to be crowd-sourced with location information and GPS.

Personas

Persona A:

Sally is a 24 year old, who recently graduated from her local state college, University of Washington Bothell, with a degree in Sociology. She is visually impaired and lives with her mother, who usually helps her out with daily tasks and getting to and from places. She has just now landed her first off campus job interview. Although she is fairly comfortable with her condition, she is nervous and self-conscious to be interviewing. It is important to her that she is able to navigate as a strong and independent person to the interview itself inside a new building, without the aid of her mother, as she usually might do.

Sally is completely blind, and has been from birth. She makes use of a seeing eye dog, cane, and assistive apps to navigate around. Sally was fortunate enough to grow up using assistive technologies and is rather comfortable with her ability to use and learn new technologies. While in school, Sally always had access to a Disabilities Services center and is very used to being around people who are comfortable being around those with vision impairments.

Sally loves to listen to stories, as well as write her own. She can always be found with a story in her hands, be in planning her next shot at writing a novel or listening to books as she commutes. She often uses her phone to take notes on ideas she has throughout the day, even as she walks, and prefers to use dictation to creatively use a word processor. In her spare time, she also enjoys baking with her friends and spending time on the beach.

Sally yearns to be an independent individual. She looks forward to soon getting a job and plans on soon after move into an apartment with her best friends from college, nearer the city. Her friends describe her as hard working and motivated.

Persona B:

Adam is a 32 year old accountant, working for a fortune 500 company. He graduated from University of Missouri- Columbia, with his certification in accounting information systems. Adam is serious and hardworking, and this trait often makes him seem overly professional to his colleagues, however if he's alone with some friends, he knows how to unwind. On weekends, he can often be found on the green of his local golf club, or exploring new restaurants downtown. He enjoys writing Yelp reviews of all the

restaurants he frequents, and if there is a particularly noteworthy one, he may take his girlfriend of 2 years to eat there.

Adam is recently engaged, and is using his free time to plan for his wedding. He often has to meet with wedding planners with his fiancé, and is apt to get lost when trying to find offices and venues. He also often needs to travel for work meetings. Since he goes to many new, unknown buildings to talk to clients, he is apt to get lost, and is embarrassed to ask for directions, even in the most confusing of places.

Adam currently uses Google Maps on his iPhone to find his way around when he drives to/from this office or buildings. He uses the voice-over that reads out directions to him as he is driving and finds that extremely helpful. Adam is almost always connected with his cell phone data plan. Since his work and free time takes him to exclusively urban areas, being without data is not something he worries about. Despite this, sometimes Adam ends up in a dead zone when lost in some of the older office buildings where his clientele work. When he is in this situation, he usually has to wonder around until he finds a signal, and then find his way back from there. Also, while on the golf course, Adam sometimes doesn't get a signal, but he isn't on his phone at these times.

Adam is only able to obtain directions to get to/from places but not within a building from Google Maps. He often struggles trying to find a particular meeting room inside the building, after he is able to find his way to the building.

Persona C:

Geoffrey is a 65 year old retired veteran. He is very technologically savvy for his age, and always likes to try out the newest apps on his iPhone, which his granddaughter gave him. Geoffrey is losing a lot of the vision in his left eye, but other than that, he is completely healthy. He lives alone, with his pet dog, Scooter, but makes an earnest effort to still go out with friends and to visit family often, even if he has to travel by plane.

Geoffrey doesn't drive because he doesn't like to drive. Instead his children or grandchildren either drive him places or he takes the bus into town to visit the shopping districts, especially around Christmas, when he can go shopping for his family. While at outlet malls, and sometimes when traveling, Geoffrey is prone to get lost. Fortunately, he is very good with maps, and can usually reason his way to someplace familiar, but with his failing vision, this is getting more difficult to do by the year. If there is no map, he often uses the OneBusAway application on his iPhone to figure out which bus he has

to take and what the wait time is. But this is getting harder for him to do as his vision is worsening and OneBusAway doesn't have a voice-over that reads out all of the buses and waiting times. OneBusAway also doesn't navigate someone inside of a building and Geoffrey struggles with finding his way around a building he isn't familiar with. A navigation application would be helpful for Geoffrey because as his vision is worsening, he is going to find it harder to read a map. He needs an application that will help him navigate within a building and that reads out the directions for him as they map won't do it all.

Use Cases:

Use Case A: *Finding Way Through an Unfamiliar Building*

- Actors
  - User navigating to unknown location within building with wayfinding app
- Pre Conditions
  - User has already downloaded wayfinding app on their iPhone
  - User has internet access
  - A map and directions based on crowd source documentation is available
- Main Success Case
  - User is asked their destination and given the choice to type or verbally respond. User says room 208.
  - App configures route and starts navigating. App audibly states the first direction along the route.
  - User starts going along the route.
  - App constantly updates with each completed step in the route. App then dictates the next direction.
- Postcondition
  - User arrives at location
- Extension Case: Amenities
  - If the user would like to detour, they indicate this using the touch input. When the user would like to continue to their pre-planned destination, the user indicates this with touch input and the app reconfigures the best route.
- Extension Case: Change Destination
  - If the user decides that they either no longer wish to go to their pre-planned destination and would instead like to be navigated elsewhere (ex. room change), they use touch input to enter a new destination. Again the

user has the option to type or verbally tell the app their destination, and the navigation process starts again.

- Extension Case: Get Lost and Call for Help
    - If the map is for any reason outdated or the user gets tired of trying to navigate themselves with the app, the user indicates with touch input that they would like to call for assistance, and the app automatically provides phone numbers for the front desk/main office of the building they are in for help.
- Minimum Guarantees
    - User finds desired location.

Use Case B: *Creating a New Note*

- Actors
    - User adding a new note
- Pre Conditions
    - The app is open
    - User is currently enroute to a destination
- Main Success Case
    - The user opens the notes menu from the map screen.
    - The user enters text into the text field on the notes screen.
    - The user submits the note which they have just typed.
- Postconditions
    - A note is added to the notes screen, reading the text that the user entered
    - The note is also pinned on the map of the search screen
- Extension Case: No GPS
    - The note is saved onto the note screen, and the app saves the location within the building that the note was taken.
    - When wifi is available, the note is matched to coordinates, based on the coordinates of the building
- Minimum Guarantees
    - A note is saved to the phone, with the current location.

Use Case C: *Removing a Route from Favorites*

- Actors
    - The user removing the route
- Pre Conditions
    - The app is open

    - User has a route favorited
- Main Success Case
    - The user opens the favorites menu
    - The user scrolls to the route they want to unfavorite
    - The user clicks the unfavorite button for that route
    - The user confirms their intentions to unfavorite the route
- Postconditions
    - The route is no longer favorited
- Extension Case: Unfavoriting from search screen
    - From the search screen, the user clicks the unfavorite button
    - The current route is unfavorited
- Minimum Guarantees
    - The route is unfavorited

The Conceptual Model:

We are designing a mobile app to assist with way finding indoors, for both visually impaired people, as well as non-disabled people. Currently, the app will be available for all iPhones, in order to incentivize common usage. This app will take input of a location inside of a building, and give directions to that location with the help of information given by users. The app will have a two search bars on its main screen where the user can input their current location and their destination. We are assuming that the user knows their current location. If not, they can use the GPS tool that will give them their location in standard address form. We will also assume that the user knows their final destination's full address. Once the user enters their location the screen will switch to a different screen that has a map and written direction to get to their desired location. The directions will be listed out in text and can be read out to the user using the voice over feature.

Users will also be able to take detours, in order to take them to the bathroom, or lounge area or any other amenity located within the building. If they are already on the path to some location, and want to use an amenity in the building, the user can detour to that particular amenity and then the application will reroute the direction to the amenity area. Once the user reaches the desired amenity, the application will revert to the user's last searched location. The list of amenities within a building will be crowd-sourced and will differ building to building. Directions can be verbally spoken to people who are visually or hearing impaired as well.

For the design of the application, we plan to use a minimalistic approach that is straightforward and gives the user simple access to all of the features we provide. Voice recognition and output will be one of the most emphasized features of the application as we feel verbal interactions is the best way to interact with the app when it comes to people with disabilities, although this will be an optional feature. We plan to make the sounds outputted by the application to be pleasant so it will not attract attention or make the user feel intimidated. Silencing and adjusting the volume of the application will also be easy to do and will not cause any confusion with just one scroll feature. Screen reading and spoken output will be a big part of the application. The buttons and features on the application will have information that if screen read, the user can understand what each button is. The user can also speak to the app in order to input their destination or input a detour such as a bathroom or lounge area. We will use crowdsourcing data in order to map out an indoor area and allow users to access information about any area, if the area is permitted for the public. The user can

The application will also allow users to add notes or warnings about an area, through typing or voice, that will help users be reminded of certain cautions within an area (ex. many people, steep steps, crosswalk, bicyclists, and so forth). In case of an emergency, the application will provide easy access to emergency services or, if user is in a public building, will provide the user the phone number to the area in order to ask for directions or ask for help. If the user likes a particular building/route and may want to re-visit it, they can add it to a list of favorite routes, which they can then revisit if they want to. If the user knows that they might be going into a building that doesn't have Wi-Fi, they can download the route they searched. Once downloaded, all buttons and items will look the same, but the GPS will not locate the users current location or follow the user as they walk. The user also has the ability to change the font color, font text and background color of the application. Although the buttons on the application can't change size, the map as a zoom feature that will allow the user to zoom into a specific location on the map. This will be a touch-sensitive feature.

Design Features: Decision, Rationale and Technical Details

General Technical Details:
The font is Arial
The background is color code 0x3399CC
Buttons are color code 0x76A9E0, except for 'Call for Help' and 'Amenities' screens
    The buttons on those screens are white (0xFFFFFF)
    The text on those buttons are color code 0x76A9E0
All buttons have a white, 3 pixel border
All buttons pictured with rounded edges have a border radius of 10 pixels
All text on blue backgrounds is white
All text on white backgrounds, or in text fields, is black.
All text on buttons is centered both horizontally and vertically.
All screens with a back button have the button on the bottom 40 pixels, and the button should take the user to the search screen, unless otherwise noted.

Start-Up Screen:

Details:
When our way finder app opens, it automatically goes to a screen that only displays the main menu and input boxes for the user to indicate where they should like the app to navigate them to. This is shown in Figure x. The two search tab have text placeholders: Current Location and Destination respectively. The user has to enter their current location. We are assuming that the user knows their current location. The location entered must be in this specific format: Name of Building, Room Name/Number, City, State, Zip Code. If the location doesn't have a room number (e.g. Mary Gates Hall Auditorium), the user can enter the room name. In this example, the room name will be Auditorium. If the user wants to visit an amenity (e.g. bathroom), they will not be able to enter that as a location, instead they will have to select it from the amenities options tab (discussed later). If the user doesn't know their current location, they can click on the GPS button to get their current location. This will populate the "Current Location:"

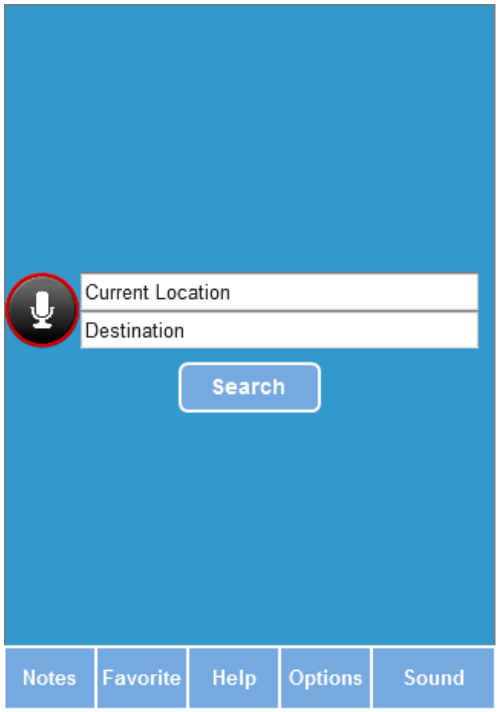


Figure 1: The App Home Screen

search bar with their current location in the same standard format. The user will then have to enter the destination they want to go to in the same format, in the "Destination" search bar. The application assumes that the user knows where they want to go. As the user is typing in their current destination or the destination they want to go to, options for locations will pop up underneath that are similar or match the query of the searched strings (characters/words). The application will use a database to store all of the locations that are crowd-sourced for this to be possible. The main screen also has a microphone icon on the left of the search bars. The microphone icon allows for user voice input. If the user doesn't want to type the address into the search bars, they can user input it through their voice by clicking on the microphone button.

Below is the script for user inputting locations after user clicks on the button:
Microphone: "Where is your current destination?"
User: "Mary Gates Hall, Room 420, Seattle, WA, 98195" (user must speak in the standard format)
Microphone: "Where is your desired destination?"
User: "Bloedel Hall, Room 070, Seattle, WA, 98195"

This will populate the search bars with the text that the user inputted using the voice input tool.

After the search bars have been populated either through keyboard text by the user or voice input, the user must click on the Search button in order to move to the next screen where their map and directions will be displayed.
On the bottom of this screen is our main navigation panel. This contains five tabs. If the user clicks on the Notes button, a box will pop-up on the middle of the screen (shown in Figure 5) with the message "message". Since the user is not currently navigating to any specific location, the Notes button on the home screen will not have notes. This is because Notes is specific to a location. The same rules apply if the user clicks on the Favorites button and the same message pops up. The Help, Options and Sound buttons have the same functionality across all screens (discussed later).

Rationale:
We chose this design for its simplicity and functionality. We thought that having a main screen that opened to a logo, as is common for many apps. We thought that this would be especially bothersome to our users who were completely blind or were dependent on a screen reader to use their device, as they would waste more time in learning how to navigate our application before they actually get the app to help them navigate them around a building. We got positive feedback from both expert users

Dorene and Amanda who mentioned that they liked being able to immediately start navigating and control the different menu configurations. We have kept both the menu options and the navigation panel available in all primary screens so our application for additional flexibility, should the user wish to change their route or the way in which they use the app while they are in progress of using the application. We believe that this is important to our design approach, b - Designing for Social Acceptance- because it allows our user to change the way in which they experience the app according to their needs and their surroundings. Often the situations in which one typically uses their device vary and change rapidly (for example, it is not socially acceptable to use a phone with loud audio during a lecture or conference, however one may still feel the need to use the application at this time), and by keeping these options available, we are able to dynamically handle this use case.

Technical details:

Current Location and Destination search bar:
Size: 260 pixels wide, 25 pixels high.
Position: the 'Current Location' field should be 50 pixels from the left, 175 pixels from the top.
the 'Destination' field should be 50 pixels from the left, 200 pixels from the top.
Search button
Size: 90 pixels wide, 30 pixels high
Position: 115 pixels from the left, 235 pixels from the top.
Microphone button:
Size: 50 pixels wide, 50 pixels high
Position: 0 pixels from the left, 175 pixels from the top
Navigation panel:
Size: Each button is 60 pixels wide, except for sound (sound is 80 pixels wide), 40 pixels high
Position: Bottom 40 pixels of the screen
Text: The text is split between five buttons, in this order:

- Notes
- Favorite
- Help
- Options
- Sound

Search Screen:

Details:

This screen shows the map with current directions for the searched route. The map, which takes up the top half of the screen, is populated with a current location indicator, as well as a path drawn to the destination. The map is only able to display one floor at a time, so if the directions encompass multiple floors, the path will only lead as far as the staircase/elevator. Once a user changes floors, the map should update to the current floor, with the route drawn from their current location to the destination. The map will also display pins which represent notes that user has made (see page 19). If the user taps a pin, the note should be shown above the pin, within a blue box. The map should also be able to populate the 'Destination' field if a user clicks on a certain room on the map screen. If a user is using a screen reader, then the map should read off which room is currently being pointed at, so that a blind user can explore the current floor using audio, and touch.

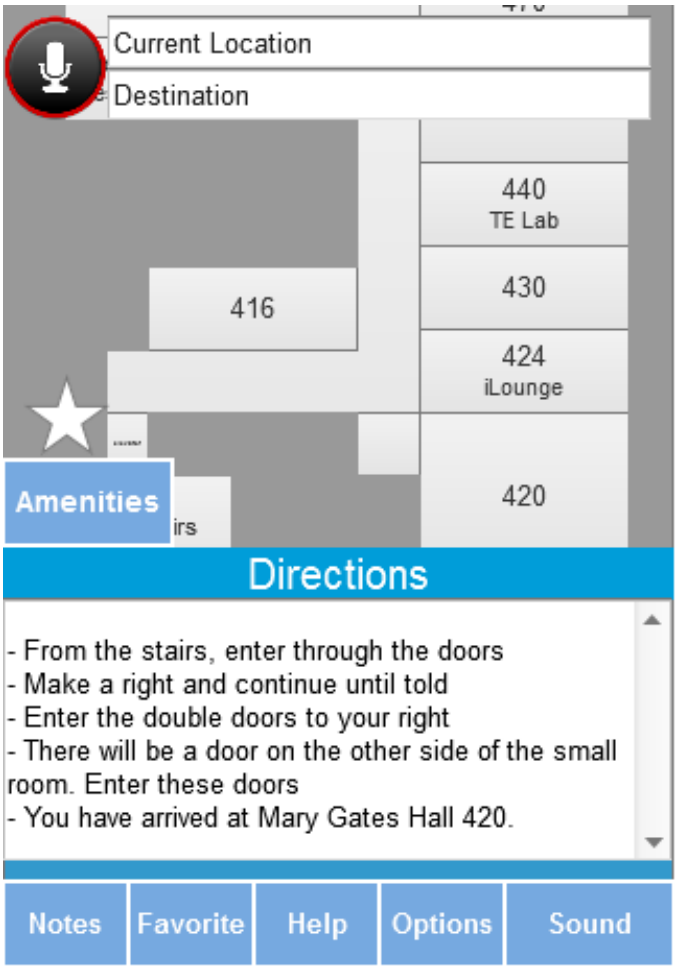


Figure 2: The Search Screen

Above the map is the current location, destination, and microphone input buttons, similar to the start screen. The destination and current location text fields should save their text values between screen navigation, and automatically populate when the screen loads. Since the user shouldn't be able to access the search screen without searching for a destination, the destination text field should never be empty upon loading the page.

On the lower left corner of the map is a star shaped button, and an 'Amenities' button. The star shaped button should, when pressed, add the current searched route to the saved list of favorite routes. If the current route is already favorited, the star button should be colored gold, as opposed to white, and if pressed should remove the

current route from the favorites list. The 'Amenities' button will take the user to the 'Amenities' page.

Beneath the map is the list of written directions. The current direction should be read off if the user has the 'voice' option under the sound menu checked (see page 27). Directions will be removed as a user completes them. If the user has 'verbose' mode under the sound menu checked (see page 27), then directions, as well as user created notes will be read off as the user passes them. Notes should not be removed once read off from the directions. If the user taps on the directions text, then all of the directions should be read off in sequence. At the bottom of the screen is the main navigation panel, which is described on page 13.

Rationale:
We needed to design a map which would be compatible with a screen reader, so that blind users could navigate with relative ease. We also wanted to allow blind users to follow the instructions at multiple speeds, both one at a time, when needed, or all at once, so that they can get an idea of where they're going. We also decided on allowing users to add routes to their favorites list from this screen. We thought this would be easier than forcing them to go to the favorites screen, and choosing to add the route there.

Technical Details:
The microphone/current location/destination text fields are located 5 pixels from the top of the screen, and have the same dimensions as on the start screen (page 14)

Map Section:
The map portion of the screen should encompass the top 260 pixels, and be the width of the entire screen. It will overlap the search bars, as well as the amenities and add to favorites buttons. It should be behind all of these elements. This map should show the users current location, as well as a route drawn to the destination. Users should be able to zoom and scroll along the map using two fingers.

Favorites Button (star shaped button):
Size: 40 pixels wide, 40 pixels high
Position: 10 pixels from the left, 178 pixels from the top
This button should read 'Add/Remove from Favorites' when a user with a screen reader highlights it.

Amenities Button
Size: 80 pixels wide, 40 pixels high
Position: 0 pixels from the left, 218 pixels from the top

Directions Section:
The 'Directions' label should be 24 pixels high, and 260 pixels from the top of the screen.
The directions should then be in a list beneath that, 285 pixels from the top of the screen, and 135 pixels high. Directions should be scrollable if not all of them fit within this area.

The navigation panel is at the bottom of the screen, and should be the same as described on page 13.

Amenities Screen:

Details:
When the user presses the 'Amenities' button on the search screen, a menu comes up listing out the different general areas that are located in that particular building they are in. It also has a button for voice input if the desire area is not listed. When the user selects one of the options, the application will then detour from the current destination and find the nearest location of one of these areas. If the user selects the 'Bathroom' option, a pop-up alert will be initiated asking the user if they desire to go to a men's or women's bathroom. Once the user has reached the desired area, the application reverts back to the user's original destination. The layout of the menu shows the title of the screen followed by wide buttons all titled with the areas. If the building has more areas they will be listed and if there are more than four then the user will be able to scroll through them all with the microphone icon still appearing at the bottom, not being part of the

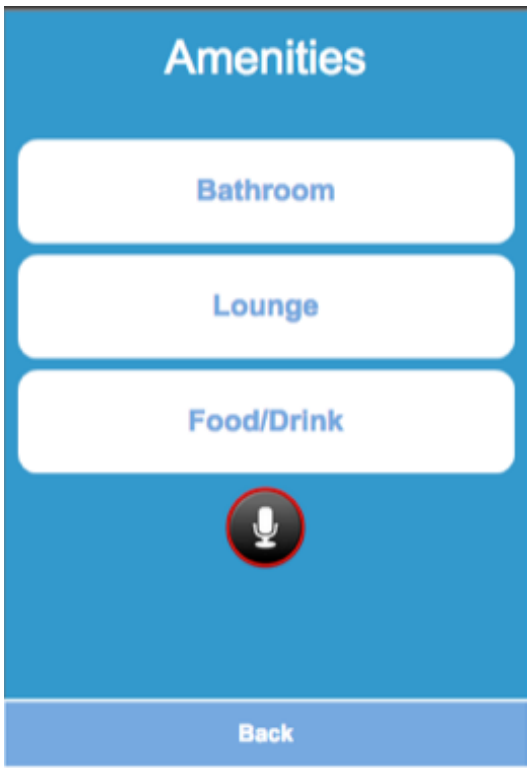


Figure 3: The Amenities Screen

scrolling. The voice input button is black with a red border and has a microphone symbol on it. At the bottom, there is a button to go back to the last screen, which should go to the map screen when pressed.

Rationale:

We decided to include this feature because our expert user claimed that she wanted to know if certain buildings had a bathroom or a café that she might want to explore. This way we included all of the necessities (bathroom and food) as well as any other major landmarks that might be interesting to the use, which is determined by the crowd-sourcing information. For this option, we did not want to make it too cluttered with different areas, so we narrowed it down to three areas that we figured were the most sought after. If the user wanted to find another area, they would have to use the voice input button in order to locate it. The overall feature is quick and easy to use. The button to access it is easily be found on the map portion of the search screen.

Technical Details:

Amenities label:

28 pt font, 14 pixels from the top of the screen, and centered.

Amenities buttons:

The screen is populated with buttons for amenities within the current building.
Each button is 46 pixels high, and 300 pixels wide
The first button is located 10 pixels from the left, and 80 pixels from the top.
Each following button is 70 pixels lower than the previous. If there is not enough room for the buttons, the page should be scrollable.

Mic button:

The mic button should be located statically on the bottom of the page
Size: 50 pixels wide, 50 pixels high
Position: 135 pixels from the left, 290 pixels from the top.

Back button on the bottom of the page

Favorites Screen:

Details:

While on the search screen, there is a button located 2nd from the left called 'Favorites'. This button is used to add the current location to a list of favorites that can be recalled and searched again in the future. When the button is pressed, a new menu will surface showing a list of favorites you have saved and a button used to save the current location. On the saved locations, there is a symbol on the far right that deletes the saved favorite from the list. Once the user is done saving the current location as a favorite, the star on the map on the search screen will be lit up, symbolizing that the current location is a favorite. The rectangle that describes the saved area has a white border encasing it on a blue background. The symbol to remove and the button to add is also white with a blue background. If the user hasn't yet searched for a route, the 'Add Current Route' button should not be visible. A user can tap on any of the favorite-ed routes, and the app will route them to that location, from their current location.

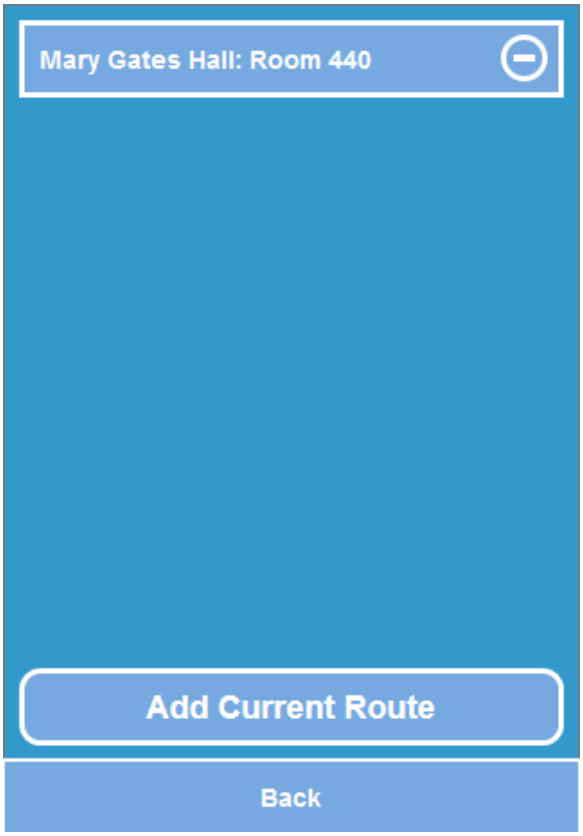


Figure 4: The Favorites Screen

Rationale:

We decided to implement a favorites feature so that our users would not have to repeatedly search for locations that they often have to go to or find themselves searching for. For the visually impaired, this feature will become extra handy because typing is already difficult enough and we made it so impaired users can have one less action they have to worry about.

Technical Details:

Add Current Route:

Size: 300 pixels wide, 40 pixels high

Position: 10 pixels from the left, 370 pixels from the top

Favorited Routes:
Favorited routes should be in a list starting with the most recently added route. The first route should be 10 pixels from the top, and each route after that should be 50 pixels below the previous route.

Back button on the bottom of the page

Notes Screen:

Details:
The notes screen is initially only a single text field at the bottom of the screen, with a plus button to the right of it, and the text 'Add Note' above it. This section is segmented off by two horizontal lines. Beneath this section is the back button, which should take the user to the search screen. If the user types text into the text field, and then hits the plus button, a note should be added at the top of the screen. If there are existing notes, they should all be moved down, so that the newest note is always at the top of the screen. All notes should be saved by the app, even after closing the app, or turning off the phone. Notes should not only save their text and the order in which they will appear on the notes screen, but also the users coordinates at the time of saving the note, and the current floor of whatever building they are on. These coordinates (and floor) will be used to populate the search screen's map with all of the notes that the user has taken, placing them on the map as pins. Each note on the notes screen is a box with text, and a subtract button on the right side. If a user presses the subtract button, there should be a confirmation popup asking if they want to delete the note. If they choose yes, the note should be removed from the list of notes (shifting all lower

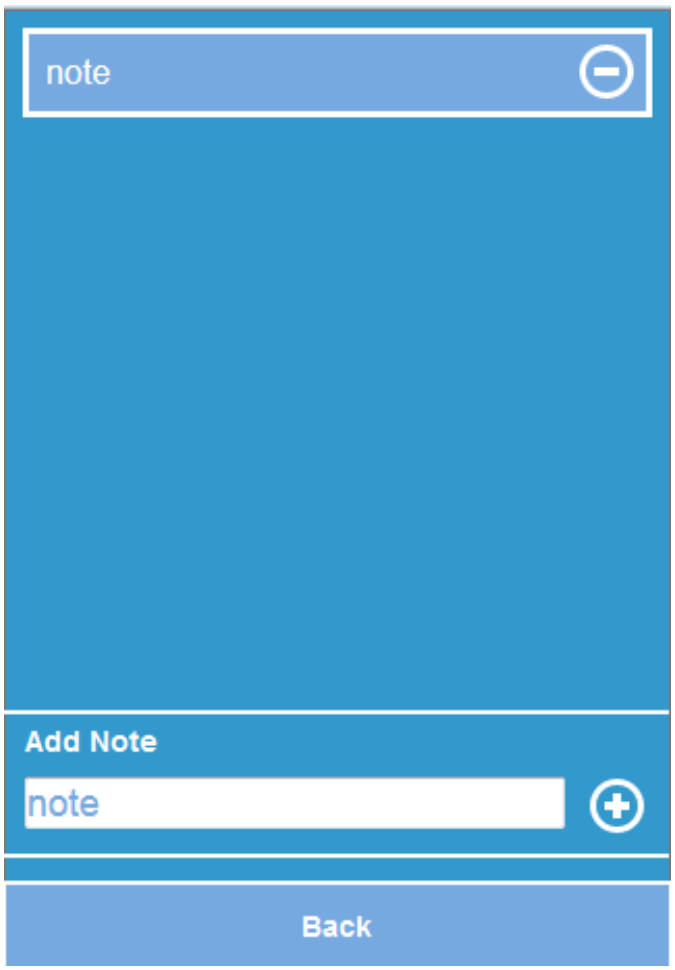


Figure 5: The Notes Screen

notes up), as well as from the map on the search screen. If the number of notes exceeds what can fit on the screen, then the notes should be scrollable, with a scrollbar on the left side of the screen.

Rationale:
We decided to have the notes function for users who discovered something along a certain route which they would like to remember. Notes could be used for remembering the location of a certain bench or water fountain, or for making sure you remember to stop by a certain room during your route. We also wanted this function to allow blind users to remember more detail which they may discover about a route, and have this detail read back to them with verbose mode (see page 29).

Technical Details:

Add New Note Section:
There are two horizontal lines at 333 pixels from the top, and 402 pixels from the top.
The "Add Note" label is 10 pixels from the left, and 343 pixels from the top, 14 pt font
The text field is 10 pixels from the left, 369 pixels from the top. It is 260 pixels wide, and 25 pixels high.
The add note button is 30 pixels wide and tall, and 280 pixels from the left, 368 pixels from the top.

Notes:
Notes should be in a list starting with the most recently added note. The first note should be 10 pixels from the top, and each note after that should be 50 pixels below the previous note.

Back button on the bottom of the page

Options Screen:

Details:
The Options button (second button from the right) on the search screen takes the user to a menu for extra options or features they may like. This screen allows the user to customize their settings by turning the accessibility mode on/off, change the font options, call for help or download the map they are currently looking at. The back button located at the bottom of the screen will take the user back to the search screen. The font options button should link to the font options screen. The call for

help button should link to the call for help screen. The Download Map button should bring up a loading icon while the map downloads (see page 28 for more info)

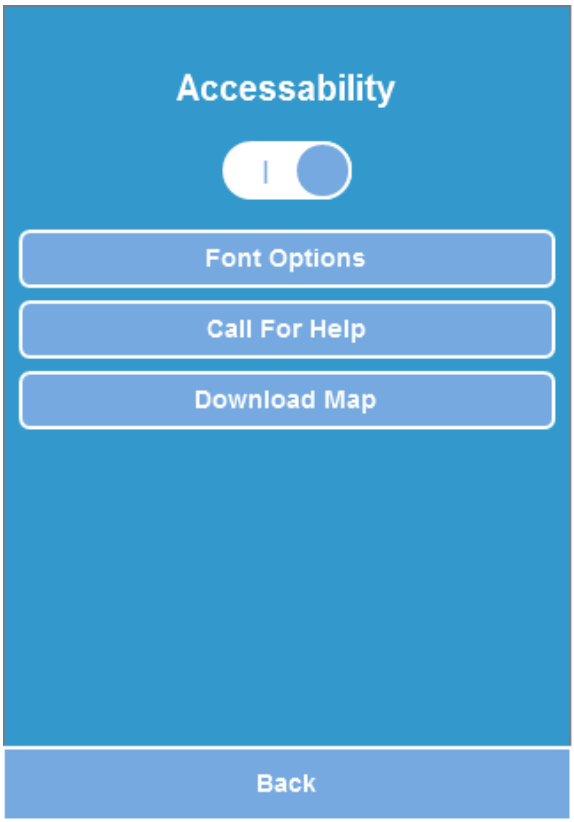


Figure 6: The Options Screen

Rationale:
The options button is meant to keep all other features that we wanted to include in a separate screen to reduce clutter on the main screen. We decided to include other features and settings that users might not want to be able to access right away in the options screen, but still want to access. Reducing clutter would make it easier for our users to navigate around the main screen and this is especially useful for the visually impaired users. We didn't want to have a main screen that was too complicated to interact with. This also adheres to our design approach as we want to make the interface of our design aesthetically pleasing to the users.

Technical details:

'Accessibility' label:
24 pixels from the top, 20pt font, centered

Accessibility slider:
Size: 90 pixels wide, 50 pixels high
Position: 115 pixels from the left, 68 pixels from the top
When the app is first opened, this option should be turned on.

Back button on the bottom

The following buttons are 300 pixels wide, 30 pixels high, 10 pixels from the left:

Font Options button
128 pixels from the top

Call for help button

168 pixels from the top

Download map button
208 pixels from the top

Accessibility Screen

Details:
The first feature on the options menu is the accessibility slider that lets the application know that the user wants it to run in accessibility mode. This is seen in Figure 6.The user can either swipe the slider right to turn it on or left to turn it off. When this mode is on, the map will choose directions that are easier to follow for wheelchair options. It will collect crowd-sourced directions that are specifically for wheelchair accessibility. For example, it will give directions that involve taking the elevator versus taking the stairs. Directions that ask the user to use the stairs will be available if the accessibility option was turned off. The default setting for the accessibility option is "on" (blue circle to the right). The user can switch it depending on their preference and will remain as their preference even if they close the application.

Rationale:
We decided that the easiest way to design an on/off button would be to use an on/off slider so the user doesn't have to click into a new screen each time they wanted to switch it on/off. This is efficient and faster. We decided to include this feature because we wanted to serve both our target audiences equally. If we didn't have an accessibility option and maps always showed directions that involved taking the stairs, then this would not be conducive to our visually impaired users. The same would apply if we only had wheelchair routes on all the time. It would not be conducive to our non-visually impaired users.

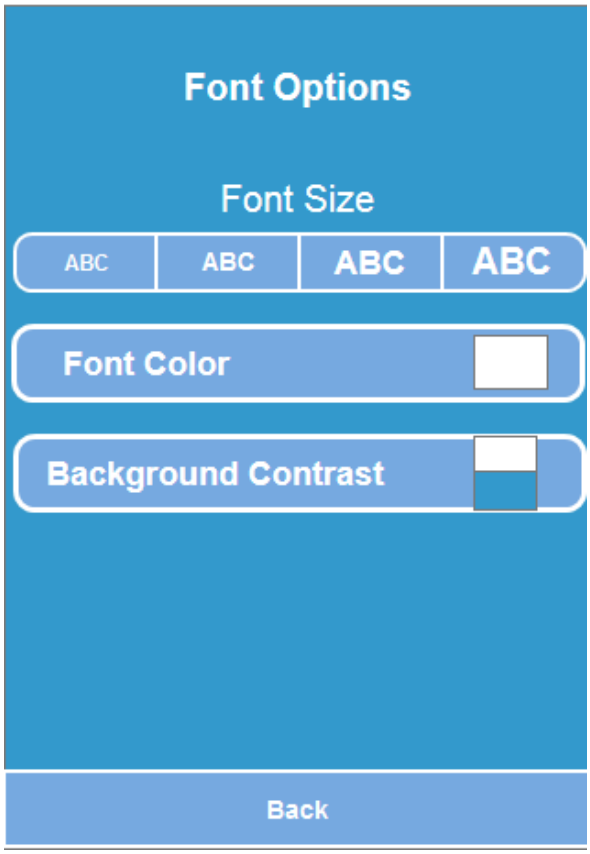


Figure 7: The Font Options Screen

Font Options Screen:

Details:
The second feature on the options menu is the font options. Font options give the user a choice of options they can choose from that helps them read better. This includes changing the size of the font, color of the font and the contrast between the font color and the background color. This can be seen in Figure 7. The size of the font can be changed by selecting from the three different font size options. The user can select the size they want by clicking on the different size buttons that are increasing in order of size from left to right. The user also has the option to change the color of the font. They would have to click on the box where the currently selected color appears. In Figure 7 this would be the white box. This will display a drop down menu of all the colors in the format of a standard font color dialog (shown in Figure 7). The background contrast will also display a drop down menu of the list of high contrast color options with a set of predetermined pairs of contrasting colors. The back button located at the bottom of the screen will take the user back to the options screen.

Rationale:
Our expert users suggested that we include a setting that allows the user to adjust the font size and color as some visually impaired users might want to increase the size of the font or adjust the color to a color they can see better. This function allows them to adjust the font settings to better fit their needs and makes navigating a lot easier. This also ties in with our design approach of social accessibility. We don't want our users looking too closely at their phone trying to read text as this might make them feel conscious and stand out. We can avoid this by allowing them to change the font size and color so they can better read the screen. This feature was added to our list of options instead of the main screen because it is something a user might only do occasionally or only once. The color of the text has a drop down menu, which allows them to see all the possible colors they can pick from. We chose the standard font color dialogue because it allows the user to choose from a variety of colors. If we predetermined the colors, the user might not like the colors and would not be conducive to their experience with the application. Additionally, after our expert user, Dorene, tested our prototype, she suggested that some visually impaired users might want to also change the color of the background to create a contrast of colors. For example, some visually impaired users can see better when the text is yellow and the background is purple. So we included another feature on the same page that allows the user to change the color of the background.

Technical Details:

"Font Options" label
34 pixels from the top, 20pt font, bold, centered

"Font Size" label
95 pixels from the top, 20pt font, centered

Font size bar:
Size: 300 pixels wide, 30 pixels high
Position: 10 pixels from the left, 127 pixels from the top
This element is split into 4 sections, each with 'ABC' written in a different font size, as follows
12, 14, 18, 20

Font color button:
Size: 310 pixels wide, 40 pixels high
Position: 5 pixels from the left, 177 pixels from the top
Color icon size: 40 pixels wide, 29 pixels high
Color icon position: 256 pixels from the left, 182 pixels from the top
color icon default color: white (0xFFFFFF)

Background color button:
Size: 310 pixels wide, 40 pixels high
Position: 5 pixels from the left, 237 pixels from the top
Color icon size: 34 pixels wide, 40 pixels high
Color icon position: 256 pixels from the left, 237 pixels from the top
color icon default colors: white (0xFFFFFF) and light blue (0x3399CC)

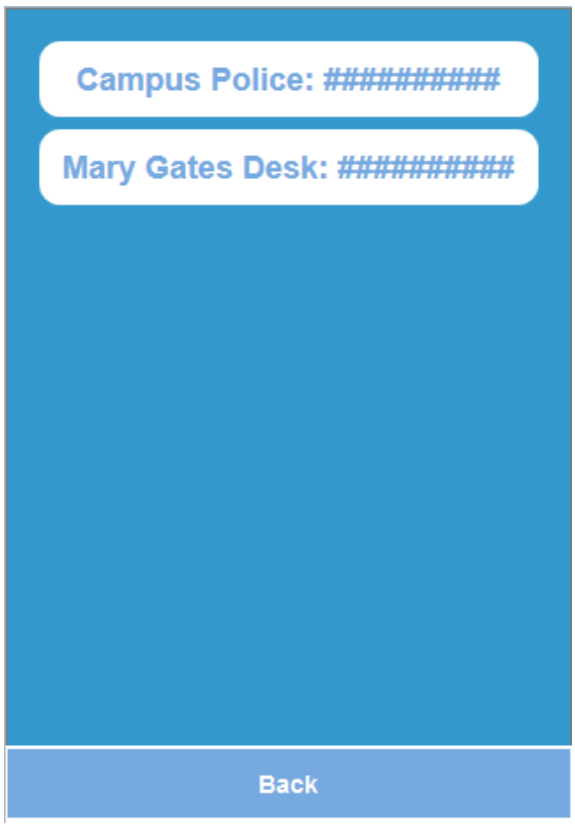


Figure 8: The Call for Help Screen

Call For Help Screen:

Details:
The Call for Help button on the options menu will take the user to a different screen, shown on Figure 8. This feature populates a list of contact numbers that are relevant to the area or building that the user is in. For example, if the user was in Mary Gates Hall at the University of Washington, the screen will populate with contact numbers for the Mary Gates Front Desk and Campus Police. This allows users to quickly find numbers relevant to the building they are in if they are lost or have any questions/concerns. Clicking on any of the buttons with the numbers will cause a small screen to pop up asking the user if they want to make the call. It will say "Are you sure you want to call Mary Gates Desk?" The user will then click on the Call button located on the bottom of the pop-up box, to call. This will automatically dial the number on the user's iPhone. Clicking the back button will take the user back to the Options Screen.

Rationale:
After speaking with Dorene at our interview session, she suggested that she would like a feature where she is able to know or call someone that could help her find her way if she was lost. Using this idea we decided to include a call for help option. We initially had another option to call 911 and other contacts that were already in their iPhone phone book as well. However, we removed that since these were redundant functions, which the user could easily do by exiting out of the app. We instead decided to have some helpful, area specific phone numbers which could be called instead, such as building front desk, or campus police. Another reason we considered removing the "Call 911" option, was due to feedback from Amanda, who noted that a blind user may accidentally call 911 on accident, if such a button exists. We also considered whether other numbers could be dialed accidently which is why we have the pop up that double checks with the user if they want to make that call. We chose to include the call for help feature in the options to avoid accidental clicking if it was part of the main navigation panel.

Technical Details:

<u>Phone number buttons:</u>
280 pixels wide, 40 pixels high, 20 pixels from the left
The first phone number is 20 pixels from the top
All subsequent phone numbers are 50 pixels below the previous phone number.

Back button on the bottom of the screen

Download Screen:

Details:
As seen in Figure 9, we also have a Download Map feature that allows the user to download the current map they are viewing, in order to be able to use it offline. If the user knows that they will be going to a building that doesn't have Wifi or service for data, they can download the map after they've searched it. Once the route has been downloaded, a message will pop up on the screen saying, "The route has been downloaded" to notify the user. The user will be able to exit out of the pop up by clicking on the "OK" button to ensure that they have read the message. Now, if the phone loses connection, the app will automatically switch to a different screen that we called the "Offline" mode screen, and notify the user with a popup saying "You are now in Offline mode". This will pull up that route that was downloaded and the user can still navigate with a screen that looks very much like the original screen. Once the phone is connected again, the phone will switch back to it's "Online" mode. The back button on this screen will take the user back to the searched screen.

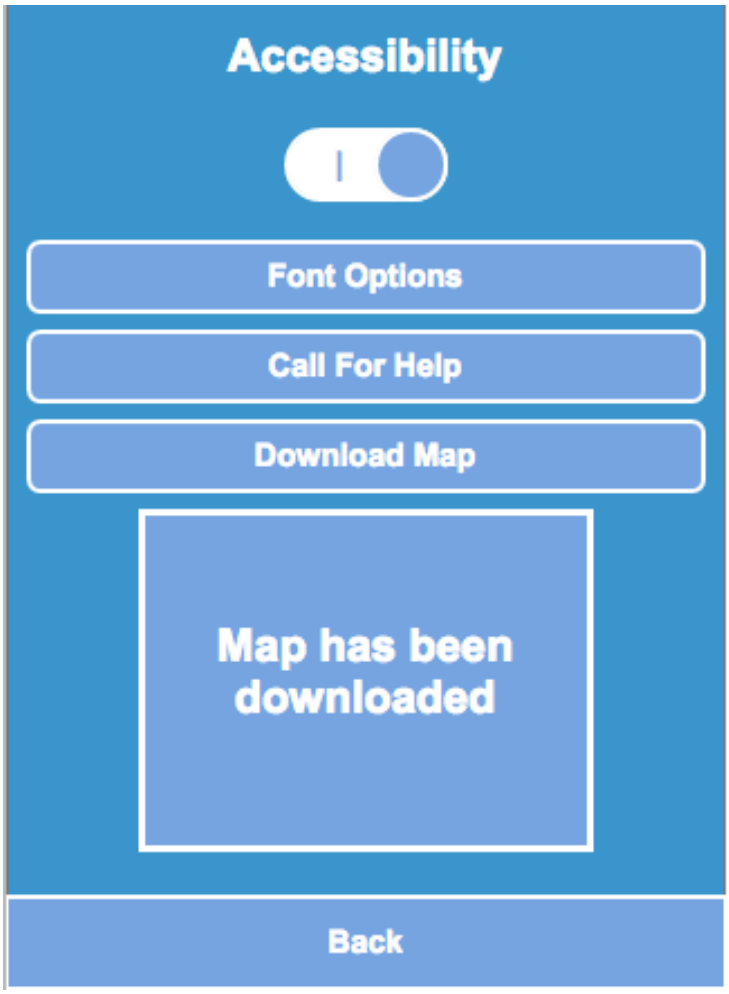


Figure 9: Download confirmed

Rationale:
We want to include a download feature because the user might be in a building where there is no service. In this case, we want to give the user the ability to download the map before they go into a building so they still have a list of directions to follow. This is an important feature for using the app in the offline mode and when there is no service or internet. We don't want our user to feel completely lost if they lose service. We still want

to provide directions to the best of our ability. The offline mode map does just that. We still have an image of the Map and Directions listed that they can follow.

Sound and Volume Bar:

Details:

The sound and volume bar has the same functionality across all screens and appears on the bottom of every screen as part of the navigation bar, along with the other buttons (Notes, Favorites, Help and Options). The user must click on the Sound button for the Sound bar to appear. It will be pop-up bar on the main screen, instead of pulling up a new screen. This makes it easier for the user to click in and out of it. As seen in Figure 10, the slider adjusts the volume of the application. Adjusting this volume does not affect the volume of other applications or the phone it self. It only adjusts the volume of the application and the sounds that the application makes. Shifting slider down will decrease the volume and shifting it up will increase the volume. If the slider is all the way at the bottom this will indicate no volume at all and the application will be in silent mode. There are also two checkboxes just above the slider. The user has the option to check or uncheck on the boxes depending on weather they want it on/off. Checking a box will turn that feature on. The two features are Voice and Verbose.

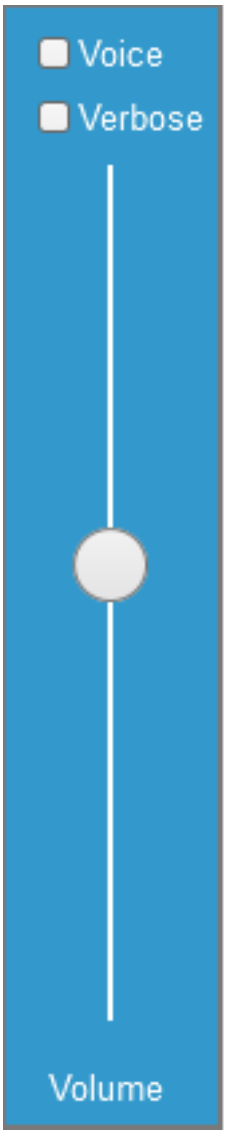


Figure 10: The Sound Bar

When a user is using a screen reader and interacting with the volume slider, the reader will output the volume percentage. For example, if the user were to slide all the way up to max, the screen reader would tell the user that it is at 100%. For others that do not need the help of a screen reader, the slider will not produce any voice output, as the application will assume the user can see the slider. It will also not show any text of the percentage of the volume, assuming that the user can gauge by just looking at the slider.

The color of the slider is a slightly darker blue (0x3399CC) than the rest of the background. This is so that the slider stands out against the background. All font color

for the text is white. The slider circle, bar and check boxes are also white. This creates an obvious contrast.

Rationale:

For our design approach, we were given the task to make our application more socially accessible and acceptable. In order to improve how users act in social situations while using our app, we narrowed down the factors that cause social self-consciousness through interviewing our expert user. Our expert user emphasized how difficult it was to adjust the volume and sounds of certain applications and the overall phone. For example, the user may have used the volume button on the side to lower the volume, but instead of silencing all of the sounds, it only silenced the ringtone and not sounds from applications. Through this, we decided that sound played a big factor as it tends to attract a lot of attention.

In order to improve this, we implemented a button that when pressed slides up a narrow panel that allows the user to adjust the volume, which ranges from silent to max. The panel also has an option for voice output that when checked allows the app to do just that when interacting with the interface. We wanted to keep the overall volume interface simple and straightforward in order to reduce confusion.

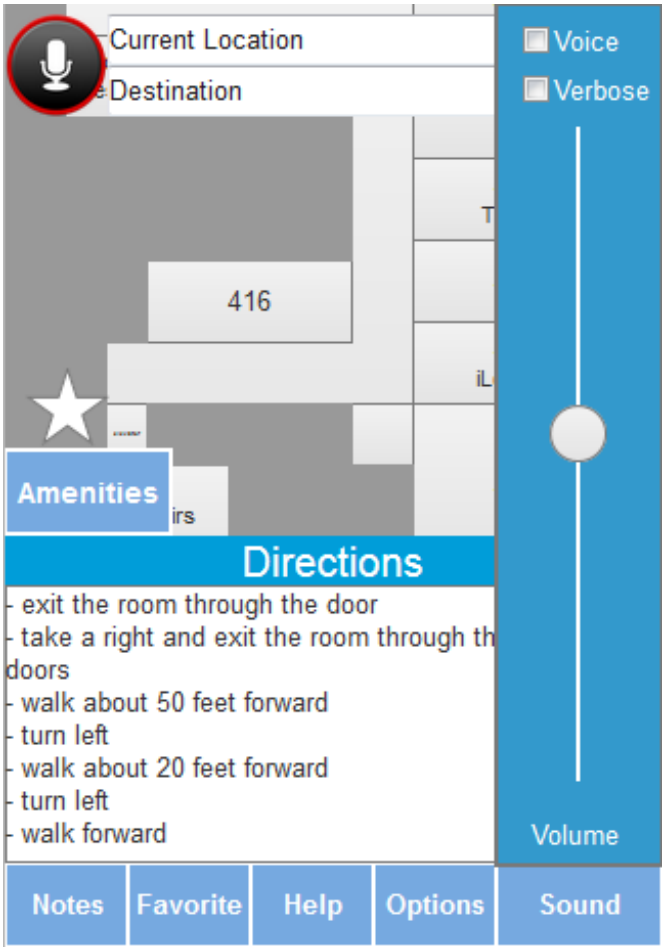


Figure 11: The Sound Bar overlayed on the Search Screen

After showing our expert user the volume feature, they were appreciative that we took into consideration of sounds and volume of the application itself, as this was an emphasized feature that our user consistently talked about. It was very convenient for them to be able to silence the application while using the application instead of having to go through different menus just to get to adjust the volume. It also gave them reassurance that the application was at a certain volume level, which allowed them to not be self conscious or worry that it will make a loud noise when out in public. It was very easy for them to understand how to use it, as the interaction for it was very simple and straightforward.

Technical Details:

The entire sound bar is located 240 pixels from the left of the screen.
It is 80 pixels wide, and 420 pixels high.

Relative to the sound bar
The 'Voice' checkbox is 12 pixels to the left, and 10 pixels from the top
The text is 13pt

The 'Verbose' checkbox is 12 pixels from the left, and 34 pixels from the top
The text is 13pt

The horizontal bar which measures volume is 320 pixels high, located 35 pixels from the left, and 60 pixels from the top

The sound slider is 27 pixels wide, and 27 pixels high
The 'Volume' label is, 398 pixels from the top, 13pt font, centered

Help and Tutorial Screen:

Design:
When the user clicks on the Help button in the navigation panel of any screen, it will take them to the screen shown in Figure 12. This screen includes the text that gives a summary of each of the navigation panel features and the amenities feature. (just as shown in Figure 12). If the user wants a tutorial animation for each of the features for more detail on how to use it they can click on the "Start Tutorial" button. This will take the user to the screen shown in Figure 13. This screen has four buttons listed for each of the features. Clicking on any of these buttons will take the user to a different screen where there will be an animated tutorial of how the user can use this feature. This video is not an interactive video like many other applications currently have where the user must follow the tutorial and click to be able to move forward and see what it does. Instead, this video will just show an example of a user playing around with a particular feature. This is a minor feature in our design and we are assuming that these videos will be recorded after the application has been implemented. All these videos will be recorded with sound where there will be a voice explaining each move and what the purpose of each button or click is. Below is a breif summary of the different actions in

the video tutorials but because this isn't our major feature, we did not include a script or storyboard for each video.

Brief details for each video tutorial:
General Wayfinding:

- *User enters current location and destination*
- *User clicks on the search button which takes them to the search map screen*
- *User follows the directions and the blue dot moves along with the user*
- *User clicks on the amenities button and chooses to go to the bathroom*
- *The map re-routes the user to the bathroom and then re-routes the user back to their destination*

Notes:

- *User clicks on notes and this pulls up their existing notes for that particular location*
- *User enters a new note for that location*
- *User deletes an existing note for that location*
- *User goes back to the search screen*
- *User clicks on a an existing note on the map to view the note*

Favorites:

- *User clicks on favorites and this pulls up their existing favorites*
- *User enters a new favorites for the location they are currently searching for*
- *User deletes an existing favorite for that location*
- *User goes back to the search screen*
- *User clicks on the star to add the route to their favorites as an alternative method.*

Options:

- *User clicks on the options button on the navigation panel*
- *User changes size of text, color of text and background color*
- *User makes a call for help*
- *User downloads the map*
- *User goes into offline mode and back to online mode*

Sound:

- *User adjusts the volume by clicking on the sound button in the navigation panel*
- *The user checks the Voice and Verbose boxes and unchecks them - explaining what the difference is.*

The back button on Figure 13 will take the user back to the screen on Figure 12. Additionally, the back button on Figure 12 will take the user back to the main search screen.

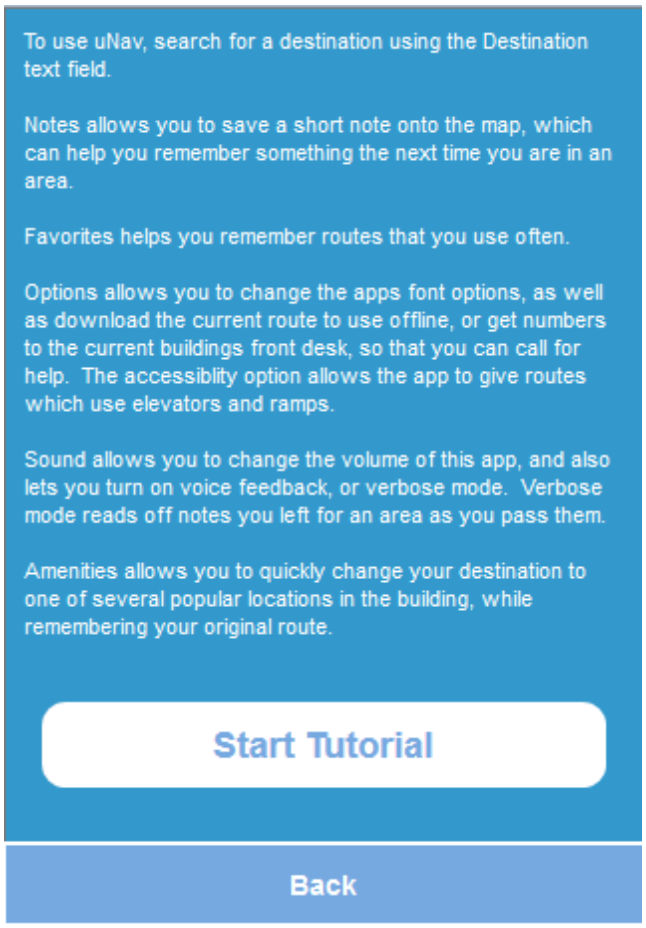


Figure 12: The Help Screen

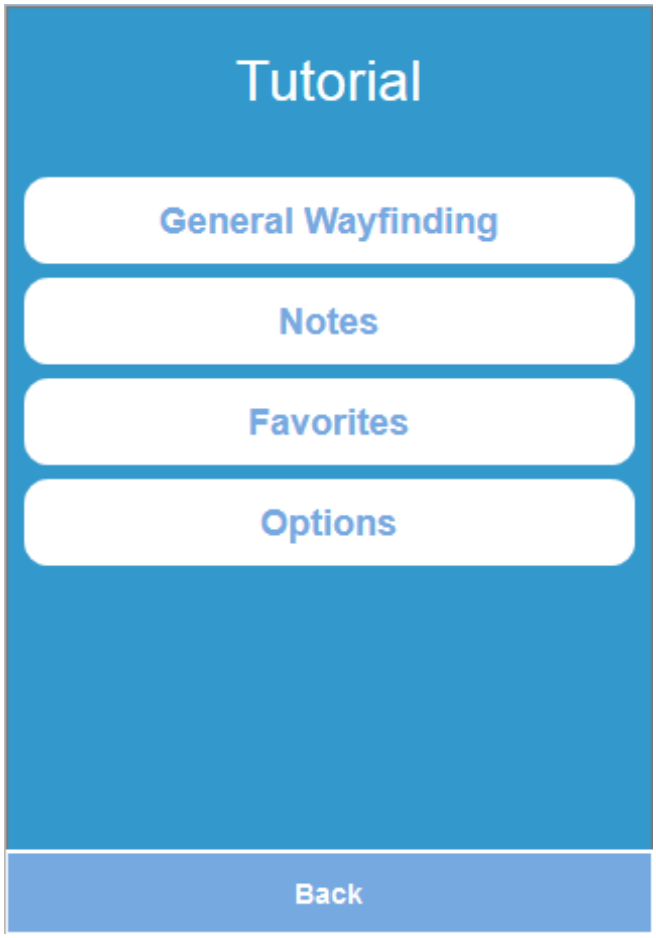


Figure 13: The Tutorial Screen

Rationale:

After doing user heuristics and accessibility tests, one of the comments we received was that the user didn't know the purpose of particular features and buttons. i4Designers suggested that we have some way of notifying the user what each feature is about and what the navigation panel buttons do. Therefore we decided to include a button on the navigation panel that would take the user to a different screen and provide all that information. We decided to include this as part of the navigation panel because we realized that there is a lot to learn about the application and we want to give the user easy access to the tutorials. We chose to have a text summary just incase the users want to be reminded of what each feature does instead of having to re-watch the entire

tutorial. We also have animated tutorials for new users to make sure they have detailed information about each feature.

Technical Details:

Help text:
Wording should be the same as what is on the screenshot 11pt font
Position: 10 pixels from the left, 10 pixels from the top
Size: 300 pixels wide, 400 pixels high

Start Tutorial button:
Position: 20 pixels from the left, 350 pixels from the top
Size: 280 pixels wide, 40 pixels high

'Tutorial' label is 20 pixels from the top, 28pt font, centered

General Wayfinding button
Position: 10 pixels from the left, 86 pixels from the top
Size: 300 pixels wide, 40 pixels high

Notes
Position: 10 pixels from the left, 136 pixels from the top
Size: 300 pixels wide, 40 pixels high

Favorites
Position: 10 pixels from the left, 186 pixels from the top
Size: 300 pixels wide, 40 pixels high

Options
Position: 10 pixels from the left, 236 pixels from the top
Size: 300 pixels wide, 40 pixels high

Brainstorming:

Early on in our design process, we had an idea of the direction we wanted to go towards and knew that we needed to make our application simple to use but packed with features and details any user might need. Designing for users with visual

impairments brought upon new characteristics and factors that we had to consider in order to accomodate. Initially, we were stumped on designing for disabilities because we had little to no design experience to begin with. However, that soon changed and we started thinking of different ways we could implement certain features that were more beneficial than if we just made it aesthetically pleasing.

Before interviewing our expert user, we were given the task to design for social accessibility, meaning that we were designing to allow users to feel comfortable using our application in any setting, without feeling self-conscious or troublesome. By having this approach in our minds, we wanted to apply it to our application as much as possible. Initially, our ideas on how to use this approach was very limited as we did not know how to improve upon certain features in order to make it socially accessible. We brainstormed many interesting features such as bluetooth sync to a digital cane, voice input and output, different modes of alert, font sizes and styles, and so forth. Many of the features we initially thought of were an attempt to think about how to make the application for user friendly to users with visual impairments.

When interviewing our expert user for the first time, we wanted to get to know her history with technology and how she interacted with them. It was also essential for us to learn how she felt about using technology in a social setting and if there were certain features that were detractors or beneficial. Through this first interview, we got to see the relationship between a visually impaired user and technology that none of us have not really been able to observe. Our user also provided us with tips on how to implement certain features and what features would make her experience with the application more simple, as well as make it socially accessible. A big part of the interview is that she emphasized sounds and noises, which made her feel self-conscious if her phone made a loud noise in public. Their main concern was to be able to easily change the application sounds and be sure to know it is at a certain level.
After, the interview, we now had a better understanding and what were the important things to implements and the features we needed to improve in order to accommodate. Although we were thinking on the right track, we started to narrow down our brainstormed ideas to a few features we know would work best for our users. These features, included easily adding favorites, a "detour" feature, being able to download maps to use offline, usage of haptic feedback throughout, and ,most important, being able to adjust application sounds in the application itself. We felt these were the best features as these were the best to implement in terms of what our user wanted in an application and our design approach.

Developing and Prototyping

In order to design the functionality and user experience, we developed in depth use cases, storyboards, created and user tested a paper prototype and then drafted our high fidelity prototype, based on the use cases and personas above.

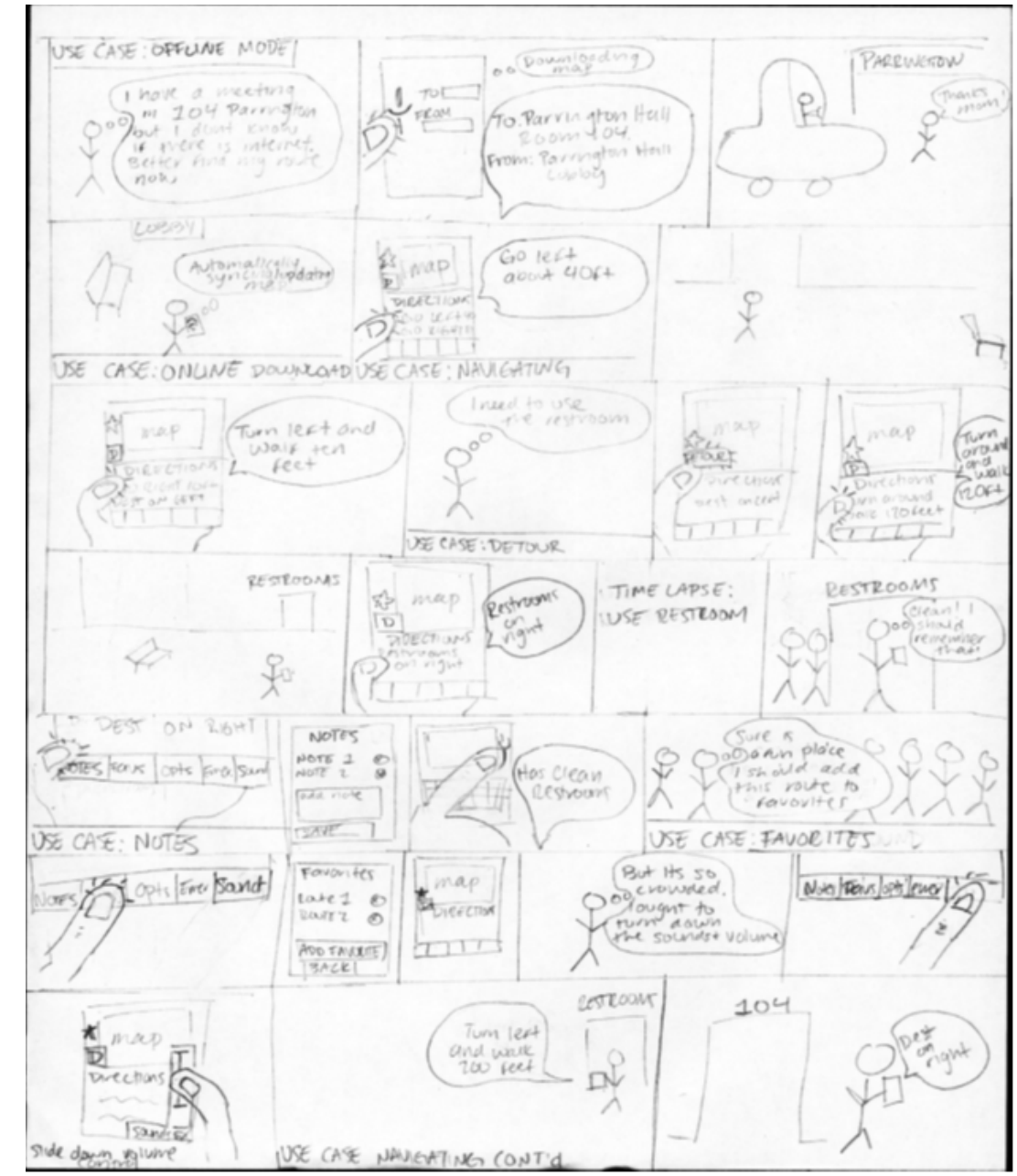
Storyboard:

The primary objective of the storyboard was to determine in what ways we wanted our user to communicate with our application. We were most concerned with the following questions: How would the application communicate with the user to navigate them through a building? How would the user communicate back with the device? How can we best shape the overall user experience? We drafted this in one fluid storyboard, shown on the following page.

The main problem we ran into was one of the constraints we were given at the beginning of the quarter - our application could not rely on access to wifi or GPS while navigating. Immediately, it became obvious that the information that the application would require to find a route would then need to be more specific and standardized to be interpreted by a computer. Therefore, we had to figure out how the application would know when to start and when the user had completed certain direction, if we could not track the users location and how we would be able to guarantee a successful end case of the user having reached their destination. Our solution was to have our user directly interact with the screen of their mobile device to let it know their progress along the route.

Our storyboard (shown in Figure 14) illustrates a user navigating our application, and demonstrates the online mode, offline mode, adding notes, adding favorites, detouring, and as an alternate ending, also getting lost and calling for help, the entire spectrum of wayfinding functionality that our application provides.

Surprisingly, there has not been any significant change to the way we currently expect our users to interact with our device since we first drew up the storyboard. In fact, wherein our prototypes did not produce the expected user interaction from this storyboard (ex. The Help feature at this point also linked to emergency contacts and allowed the user to call 911), all implemented solutions resulted in the exact same user interaction that is displayed in the storyboard.



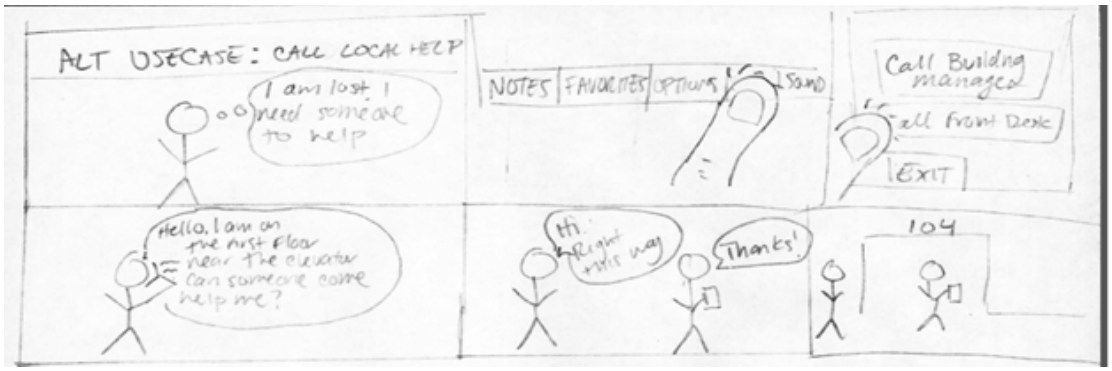


Figure 14: Storyboard Sketches

Paper Prototype and User Session:

With our first user testing session on the horizon, we prepared a low fidelity paper prototype so we could better understand how our expert users would expect to use and interact with our application. This prototype was built entirely out of notecards and sticky notes, and was based on the Storyboard. We chose to use sticky notes to demonstrate small updates to the screen, rather than entire screen changes, to allow for additional flexibility during our tests and demonstrate the usability of each template. This was particularly useful for the notes and favorites features, where only small changes are made to the display after each user interaction.

To the left, we have the two opening screens of our application and the detour menu. The first, the welcome screen, is what we initially presented to our expert user.

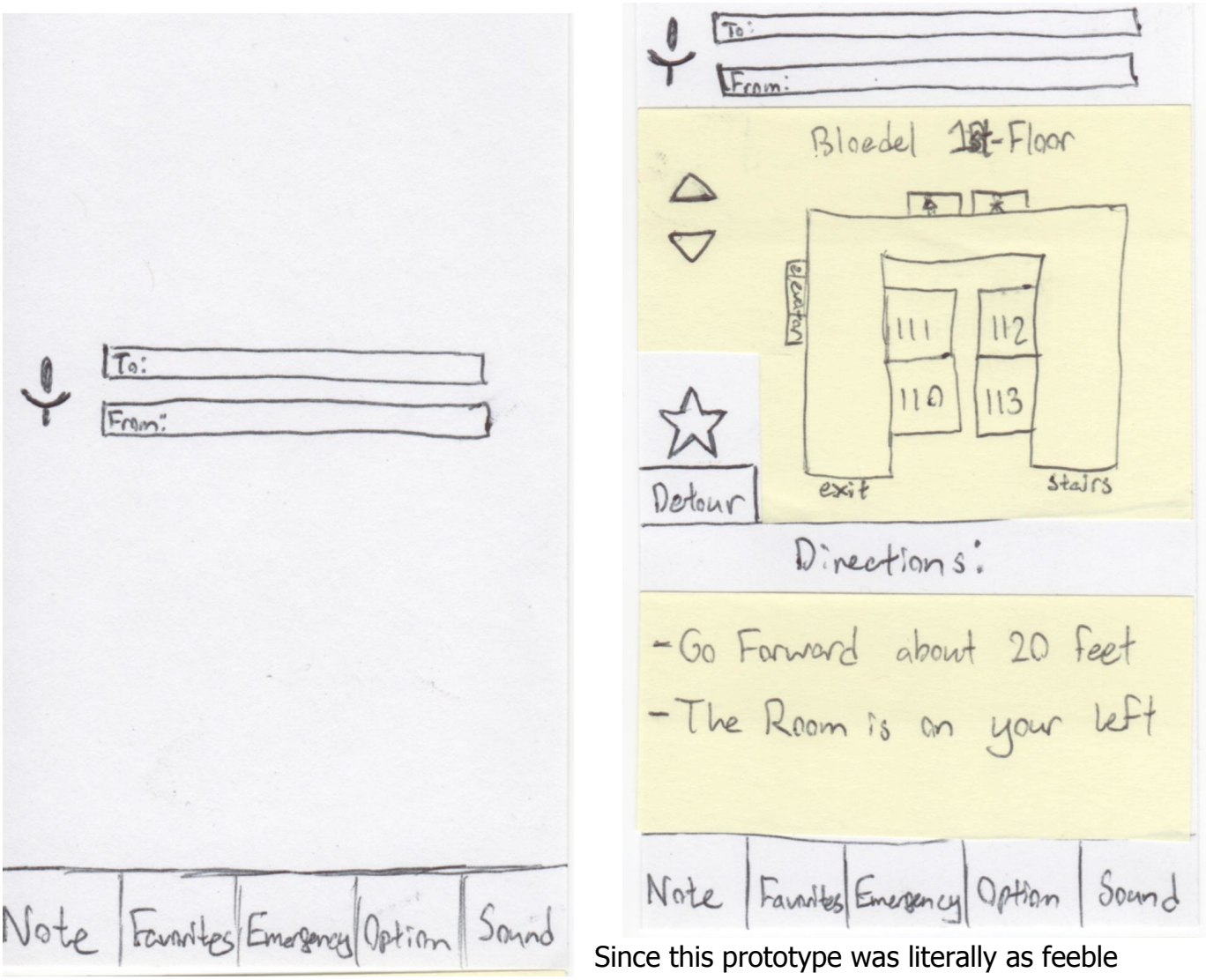


Figure 15 The Home Screen (left) and Search Screen (right) of the paper prototype

Since this prototype was literally as feeble

as paper, Russell and David shared the role of "Wizard of Oz", by pretending working as a screen reader and switching between, as well as, updating the notecards with the appropriate pre-prepared sticky note as our expert user selected each option. This would give the illusion our application was working itself, regardless of the fact that the tool itself had yet to be developed. Dorene gave us very positive feedback on these two screens, however, she noted that "Emergency" in the bottom menu along every screen was ambiguous. For this reason, we changed the label of this option to "Help", otherwise these index cards do reflect our final prototype.

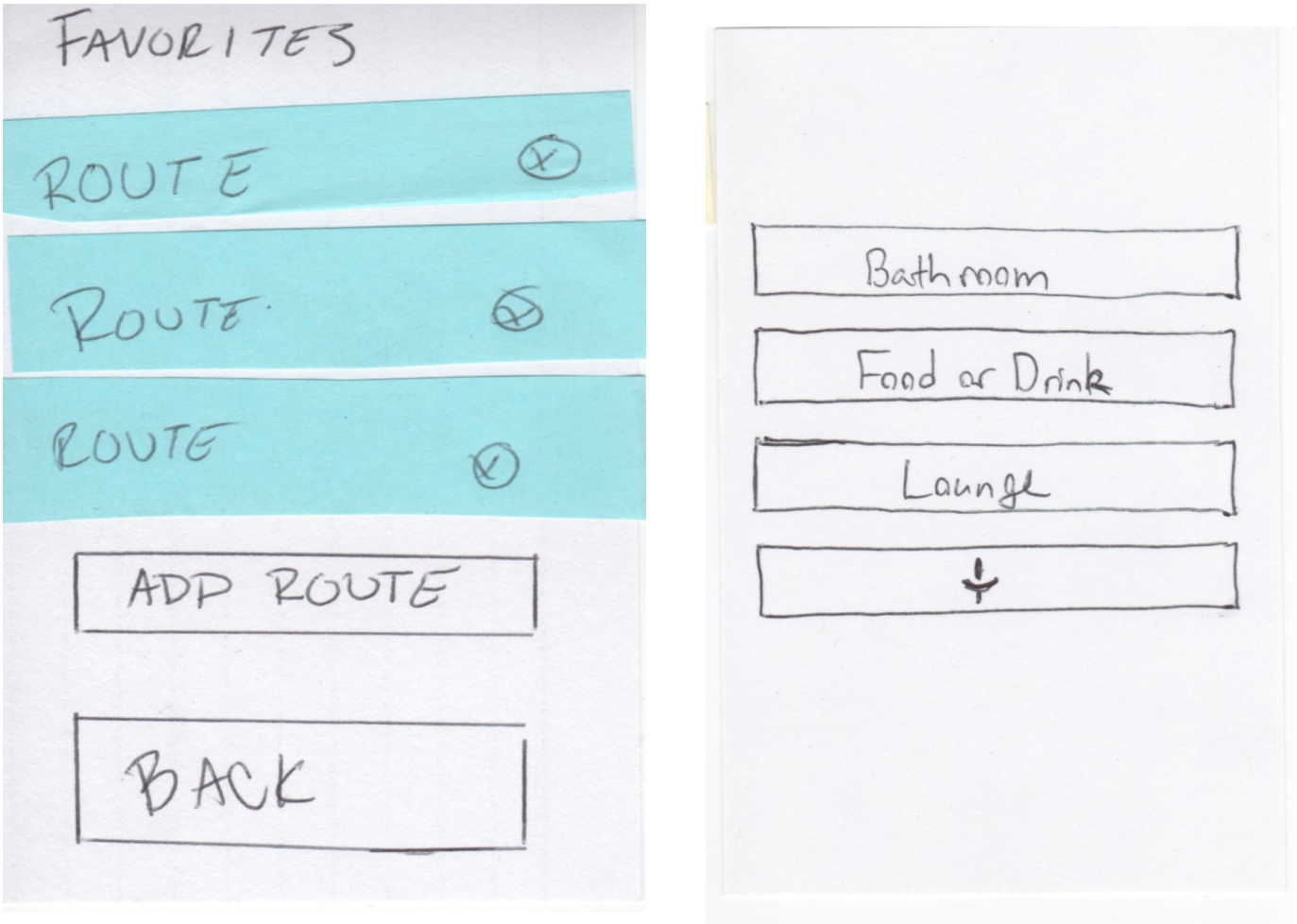
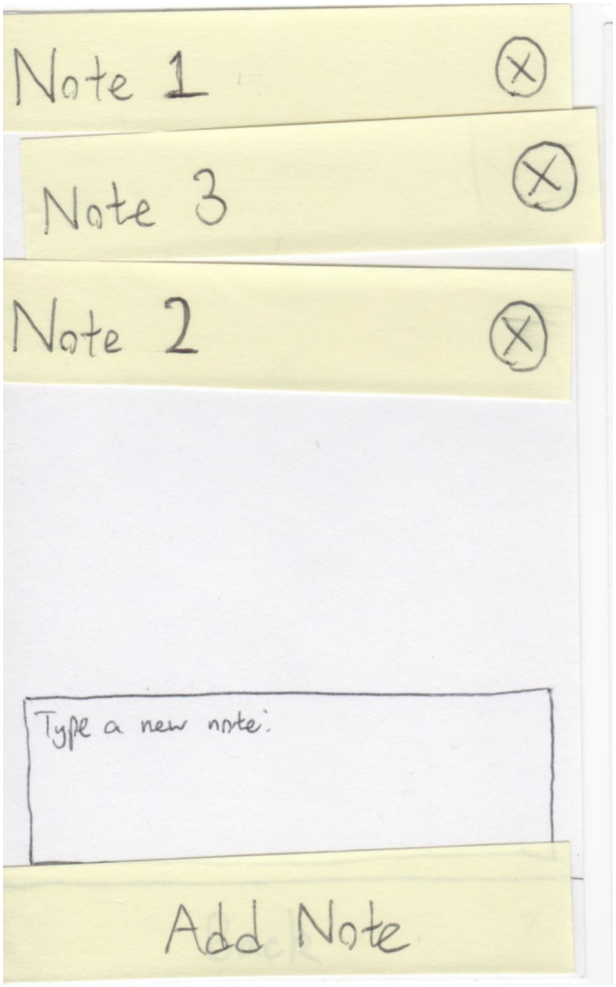


Figure 16: The Notes, Favorites, and Detour screen of the paper prototype



After exploring these screens, Dorene explored our application going along that bottom menu from left to right. The first of these being the "Notes" option, followed by "Favorites", which work almost identically to each other in this prototype. At this point in our design process, we had envisioned the "Notes" feature being just a local storage for anything the user wanted to be automatically reminded of while navigating the route, and again, the process of adding and removing a note or favorite implemented here is identical to that added in the final project.

The feature that changed the most bases on the feedback obtained during this user session was the "Emergency" menu. At this point, the user could use this feature to call for help should an emergency arise, in any situation, such as calling 911 or a predetermined emergency contact. This is also how a user might access relevant local numbers that might be called should the user experience any trouble navigating or maneuvering around a building.

During user testing, it was presented to us, that an emergency option in this manner is not only designed poorly for users with visual imparities, but also is redundant for an application that already has this capability. When we were performing user tests with Amanda, she pointed out that the size and orientation of the big "Call 911" button would most likely only result in people accidently calling 911 while trying to use the application with a screen reader. While she liked the idea of having relevant numbers available, she also noted that it would be redundant to have emergency contacts programmed into the application, separate from her main phone application. She also noted that using these menus would not likely save the user any time in an emergency and explained that should she encounter an unfortunate situation, the fastest way to get help, would be to exit out of any app currently in use, go do to her phone's main screen, and then use the preprogrammed contacts. To the right, we have presented the screen that was part of our paper prototype that allowed the user to create a new emergency contact. This stressed just how redundant this functionality would be be,

especially as the application would most likely be running on a iPhone. For these reasons the "Call/Add Emergency Contact" were removed completely from our application and was retained as "Help" which only provides the user with relevant number, such as a building manager, that a typical user would not likely have preprogrammed into their phone, or know off the top of their head.

For our paper prototype, our options button opened up a menu screen that allowed the user to change the way they interact and experience the application. Accessibility would direct to another screen that allowed the user to search for handicapped accessible routes, fonts would open up another page that have the user the choice to change the font size and color scheme of the application, and Download allowed the user to download a map to use while they are offline.

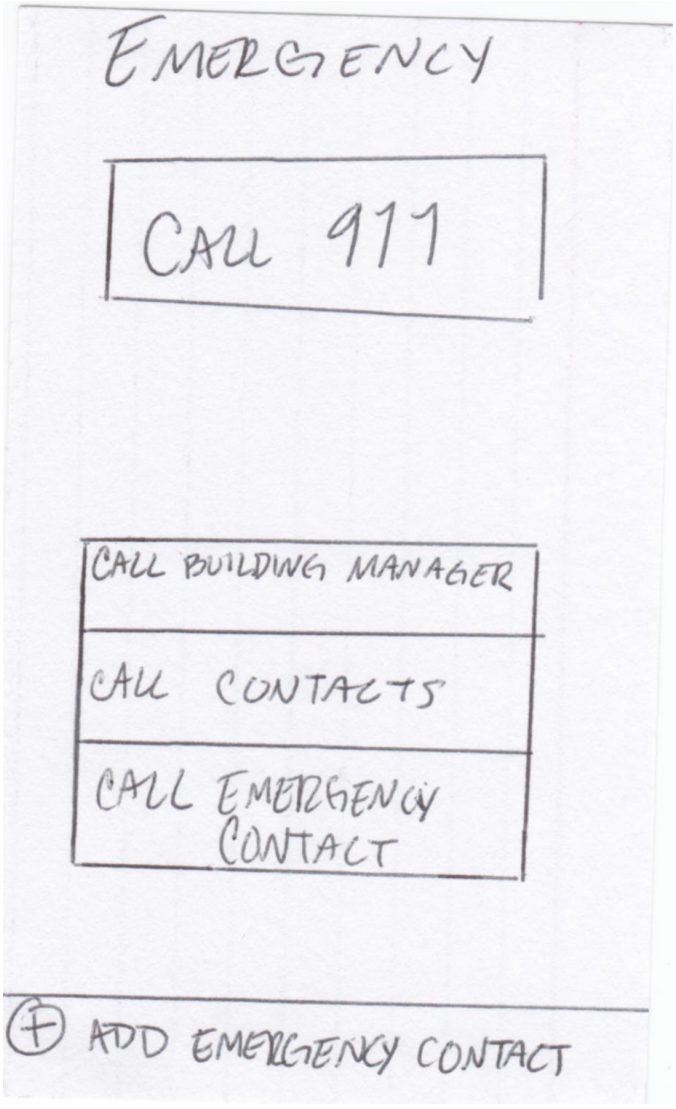


Figure 17: The Emergency Screen of the paper prototype

During the user session, it became apparent that we had these different menus and submenus were redundant and overly nested within each other. After this user testing session, we decided to streamline the customization process by merging the main options menu with the accessibility submenu, by allowing the user to change it directly from this screen. We decided that it would not be appropriate to do the same with the Font submenu, because it was much more complex and to merge the two screens made the Options menu far too cluttered and complicated.

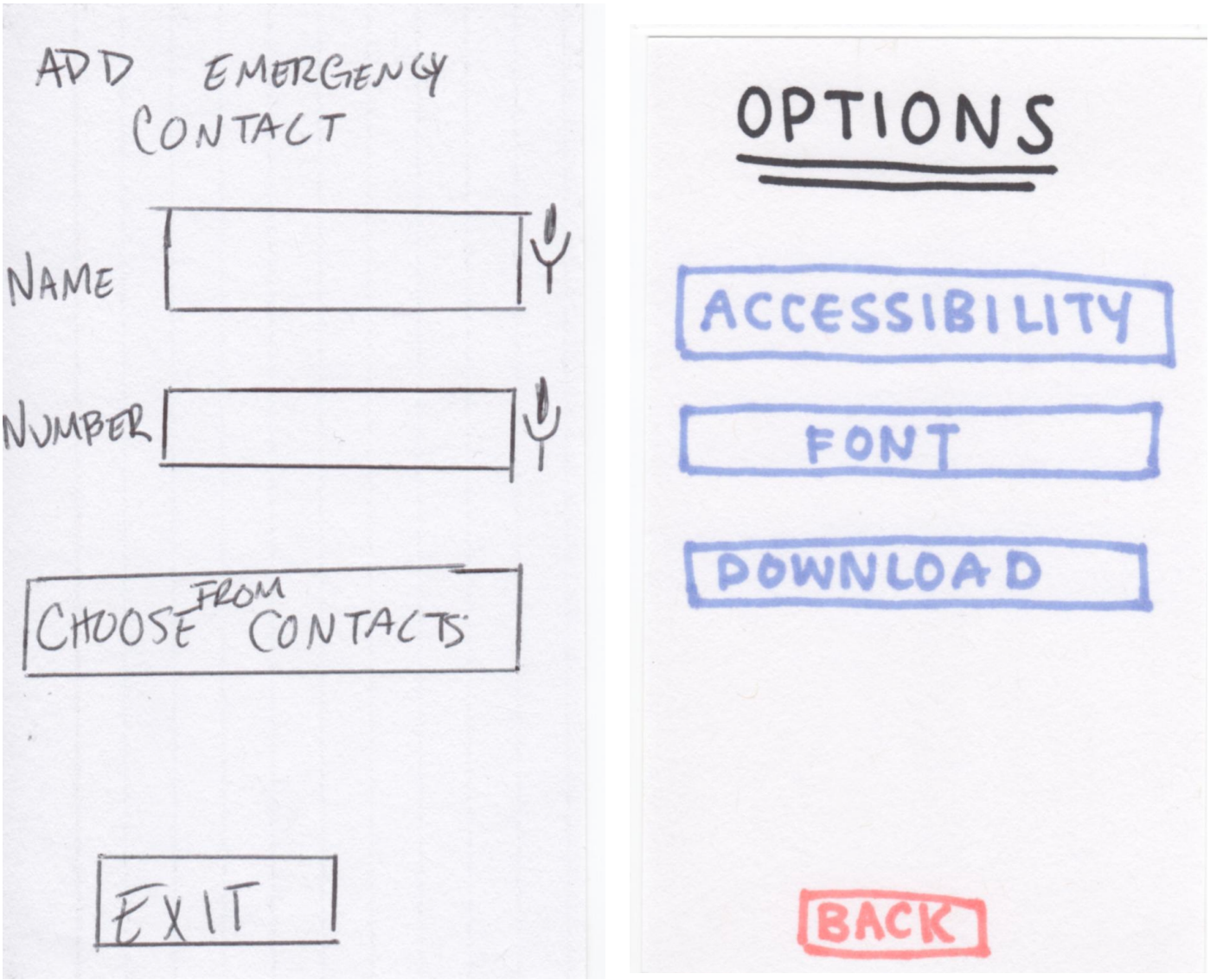


Figure 18: The Add Contact Screen (left) and Option Screen (right) of the paper prototype

The Final Prototype

Below is a working demo, available for testing. Instructions for viewing demo:
Click Here!

High Fidelity Prototyping Constraints

- Our prototype has several constraints limiting its overall functionality. The most obvious constraint is the lack of map data needed to route users. The final product will have a dot that will show the users current location and direction.
- The map will also contain a line from the current location to the entered destination. The 'Current Location' text field will be populated by default with the users current location, if it is available. We couldn't prototype this functionality in Axure, since the program isn't able to generate pages that access the users GPS.
- Our prototype also doesn't include the correct step-by-step directions that a user would take. In the actual design, these directions would be unique to each route being searched, and even unique based off of certain settings, such as verbosity, or accessibility modes.
- Our prototype doesn't account for these. Our prototype isn't actually able to save user data to the phone, such as notes or favorite routes, which the final design should be capable of.
- The prototype also doesn't have working font options, so changing font size, color, or contrast will not affect any of the text within the prototype.
- The prototype is only capable of saving a single note and a single favorite route at a time.
- The number of notes or routes should be effectively unlimited in the final design.
- The tutorial videos are not in our prototype, but should be located in the 'Tutorial' page of the final design.
- Amenities should be relevant to the current building, and not the same three static options which our prototype has.
- When the sound bar is visible, interaction with elements on the map should be disabled, and clicking anywhere outside of the sound bar should close the sound bar. Also, the volume slider of the sound bar should change the volume of the app.

High Fidelity User Session:

During our third meeting with our expert user, we had our hi-fidelity prototype up and running on our laptops and phones and everything started to feel as if it was coming together. Although there were still some bugs we had to fix, the overall application was good enough to be tested by our user and to be able to get feedback from. Our user was excited to see that we were able to get our application on to our phone and was also delighted to see that we have implemented the features she highlighted to us. After giving a brief introduction to our application and an overview of the layout, we asked our user to interact with the application.

Our first tests was to see how well our user could navigate through the application. We asked her to do many things such as add a new note, add the current location to favorites, adjust the volume, and so forth. A detractor, however, is that our voice output was still buggy and did not read out the proper things. Although we had mentioned this to our user, it still caused some confusion due to all of the noise. After having her explore the application a bit, we then asked her to try to use the application to navigate through the 4th floor of Mary Gates Hall. Due to our sounds not working properly, we had to act as screen readers for our user. During this time, we were able to learn about more details that we had to consider and implement into our application. We also learned that our user liked using her phone in landscape mode which is something we did not consider when designing our application.

From this session, we learned that there were still some things to be fixed and improved upon. We were on the right track and were very close to getting down all of the details we needed. We learned about smarter features that we could implement through this user session and saw how a user may use our application in the real world. As mentioned, seeing our application on a phone and being used was really sight to see. It was the first time we had seen our culminated work go to action.

User Testing & Heuristics:

Paper Prototype Expert User Testing

How we tested it:
While using our paper prototype, we would test out certain features by asking our user to do a certain task, such as add a note, add to favorites, take a detour, and so forth.

We also acted as screen readers to our user as an attempt to generate a more mobile-like feel. Our user did have some trouble finding certain features and doing some actions, but in the end was able to do what we asked. It also gave our user a chance to start exploring what features we had implemented so far and gave them an idea of what we have been thinking about.

Information gathered:
This was the first time we got to sort of see our user interact with our application. We learned from observing our user that our application had some language barriers and certain features that could have been worded more accurate. Our prototype also used a good mixture of colors, which we figured would be a good thing to implement in our future designs so that certain things throughout the application can be more visible and a bit more helpful for the visually impaired.

Feedback from users:
Our expert user mainly gave us tips on how to improve details throughout our application, such as wording, overall layout of buttons, and how they would interact with certain features. They praised the features that we implemented in order to satisfy what they wanted most out of the application, such as sound. They also mentioned that the application was easy to navigate due to the simple interface and layout. They did however give us tips on how to change certain features, such as emergencies, as they did not know if the feature was going to be used much at all. Another thing they mentioned we can improve on is the overall design of the map of the area. The map should be very detailed with its labels and should not cause any confusion. Also, the map being able to output certain details about an area automatically would be a great feature as well.

High Fidelity Prototype i4Designers Testing

How we tested it:
This testing session was different because we weren't having our expert users test it. Instead, we had another team test it. We handed them the high-fidelity prototype and allowed them to explore the application themselves. They conducted a heuristic evaluation on our prototype. They didn't have a chance to ask questions and we didn't have the chance to clarify or point things out at all during this testing session.

Feedback from users:

I4Designers commented on our use of names and labels for buttons. They mentioned that some were not specific enough and didn't convey enough information about what the feature did. For example, they didn't understand what the "detour" button was supposed to do. But after we explained the purpose of the detour button, they suggested be change the name of it to something that was more true to the functionality, such as "amenities".

They liked our notes function but also questioned the necessity of it. They asked us if users would really want to add notes. But as a team we decided that we are going to keep the notes feature because that reduces the amount of information that the user needs to remember about each route. They also suggested that we include a confirmation for deleting things as a user might accidentally click on the delete button and can't undo it.

They also suggested that we include a help feature that gives the user information about what the different buttons and features do which will help the user navigate the app easily. I4Designers felt that we should move the download map button to the main screen instead of the options tab because users might not want to go through the effort to click many buttons, just to download a route. They would want the download feature to be easily accessible.