MyPath Application System Documentation

Miami University (Oxford, Ohio) Senior Capstone

Creation: Spring 2021

Last Update: Spring 2021

Group Information

2019-2020

The MyPath (formerly known as Wheelshare) project was originally started in the 2019-2020 academic year by a CSE capstone group. The original group created an Android application in Java.

2020-2021

In the 2020-2021 academic year, the capstone group consisting of Andrew Grimes, Tooba Masood, Alex Chirokov, Devin Harris, and Lance Grengbondai continued the project. The initial project scope was to create an application for iOS and to add additional features to both the Android and iOS applications.

After some deliberation, it was determined the best course of action would be to create the app essentially from scratch using Xamarin. There were many advantages to this, including:

- The ability to create an Android and iOS application simultaneously
    - Cuts development time in half
    - Creating an iOS app in Swift would take approximately the same amount of time as creating an app for both platforms in Xamarin
- Achieve similar UIs
- Make updates and maintenance easier

There were two (2) teams working on this project: the application/frontend team, and the machine learning team.

Project Specifications

Purpose

MyPath's vision is to provide users with customized paths between destinations. The paths are ideally customized by taking the user's personal information into account. This allows the route provided to avoid certain barriers (i.e. fallen trees for users that may not be able to go over or around them), include specific facilities (i.e. curb cuts for wheelchair users), and avoid unsafe conditions (i.e. terrain that may be difficult for some users to physically navigate).

MyPath hopes to crowdsource information to allow for a better algorithm and more real-time updates, allowing users more independence than they may have had in the past.

Target Audience

The MyPath application's main target audience is people with disabilities that affect their physical movement (i.e. people who use wheelchairs, crutches, boots, etc). The app, however, can be used by anyone in the general public.

Main Features

While the app has many features that make it functional, there are a few that make it stand out from other navigation apps:

- Route is determined by an algorithm that takes the user's personal physical information as well as their ASIA score into account
- Users contribute directly to the algorithm by contributing information such as geolocation and sensors

Technical Information

Languages and Softwares Used

The app was coded for both Android and iOS using Xamarin, C#, and VisualStudio.

Xamarin Setup

Information on how to set up Xamarin in VisualStudio can be found in the Appendix

Xamarin Libraries

Note: Not all of the following libraries were used, this is just a list of all installed libraries.

- Xamarin.Essentials by Microsoft
- Xamarin.Forms by Microsoft
- Xamarin.Forms.Maps by Microsoft
- Microsoft.CSharp by Microsoft
- MySql.Data by Oracle
- MySqlConnector by Bradley Grainger,Caleb Lloyd
- NETStandard.Library by Microsoft
- Newtonsoft.Json by James Newton-King
- sqlite-net by Frank Krueger
- SQLitePCLRaw.core by Eric Sink
- System.ComponentModel by Microsoft
- System.ComponentModel.Annotations by Microsoft
- Xam.Plugin.Geolocator by James Montemagno

Xamarin Resources

Sensors

The app uses the phone’s geolocation & sensors during routing & data collection.

- Latitude, Longitude, Altitude
    - Denoted in database as latitude, longitude, altitude
- Accelerometer
    - Denoted in database as ax, ay, az
- Gyroscope
    - Denoted in database as gyx, gyy, gyz
- Gravity

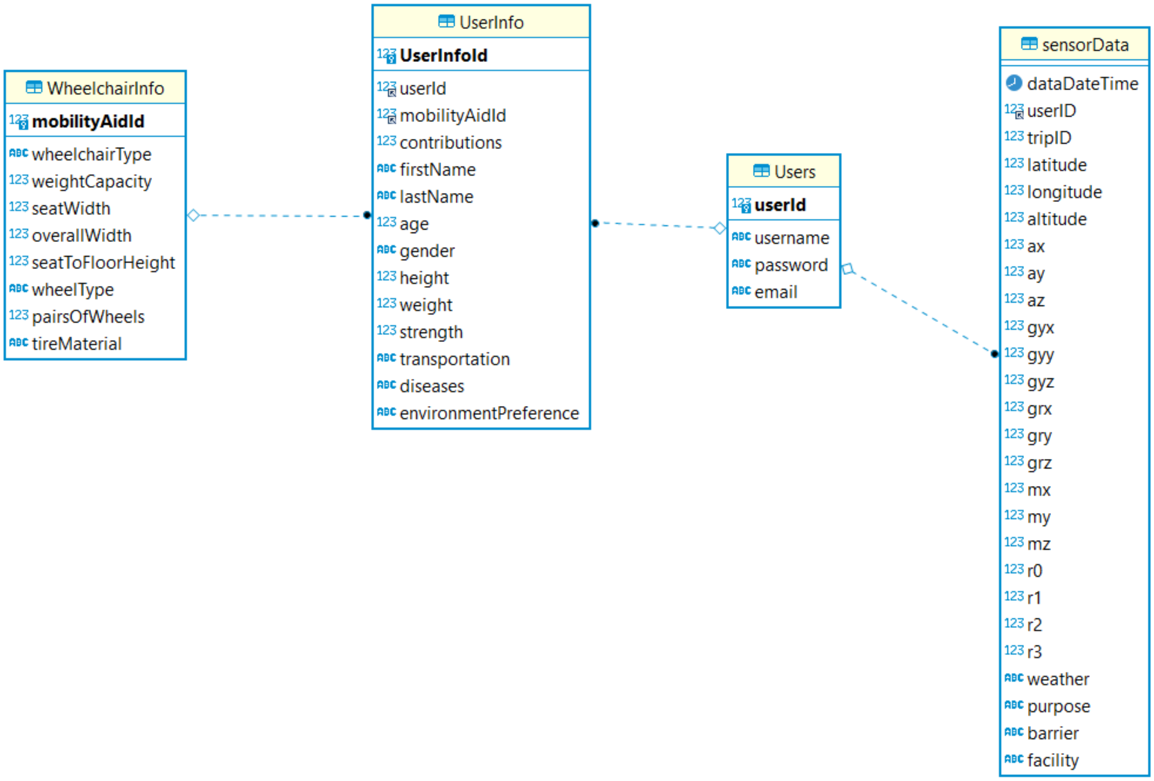
  - Denoted in database as grx, gry, grz
- Magnetometer
  - Denoted in database as mx, my, mz
- Orientation
  - Denoted in database as r0, r1, r2, r3

Database

Access information:

IP: 157.230.80.67
Port: 3306
Account: wheelchair
Password: PZYQ6xuG3CQTYEkG
Database: Wheelchair

Schema



WheelchairInfo
mobilityAidId
wheelchairType
weightCapacity
seatWidth
overallWidth
seatToFloorHeight
wheelType
pairsOfWheels
tireMaterial

UserInfo
UserInfoId
userId
mobilityAidId
contributions
firstName
lastName
age
gender
height
weight
strength
transportation
diseases
environmentPreference

Users
userId
username
password
email

sensorData
dataDateTime
userID
tripID
latitude
longitude
altitude
ax
ay
az
gyx
gyy
gyz
grx
gry
grz
mx
my
mz
r0
r1
r2
r3
weather
purpose
barrier
facility

APIs

We worked mainly with GraphHopper during development. Towards the end of the project we changed our code so it called the machine learning group's API instead of GraphHopper's API.

- GraphHopper Documentation
- GraphHopper examples
- Sample call to ML group API: http://157.230.80.67/route?point=39.51266%2C-84.741411&point=39.510442%2C-84.736633&locale=en-US&vehicle=wheelchair&weighting=fastest&elevation=false&turn_costs=undefined&ch.disable=true&use_miles=false&layer=OpenStreetMap

JSON

This is the agreed-upon design of the ML group's API both teams agreed upon for the map. We based our code on this. As of our last interaction, it appeared that the API was not fully complete. This is what the final API should look like.

WHAT WE (the app group) WILL SEND OVER (in JSON):

string[]: startCoordinate;
string: startingLocation;
string[]: endCoordinate;
string: endLocation;
User: userObj (includes age, gender, height, weight, strength, transportationType, diseases, environmentPreference, wheelchairType, weightCapacity, seatWidth, overallWidth, seatToFloorHeight, wheelType, pairsOfWheels, tireMaterial);
//userObj will have the wheelchair/personal information for ML's algorithm. If a
//user doesn't use a wheelchair, they will not have info for wheelchair related info

WHAT WE WANT (as a JSON response from the ML group):

string[]: coordinates; //we need a new coordinate pair for each change in direction/turn

Mockups and Logo

See Appendix (Mockups and Logo)

Features

Pages Implemented

The following are pages and features within the app:

- Sign In
- Sign Up
- Editing functionality for user (email, username, password), user information (personal information), and transportation information
- Trip settings
- Map routing
  - Visual route
  - Step-by-step directions
- Data collection
  - Reporting barriers
  - Reporting facilities

For the future

Features

- Overall App
  - Maybe some iOS editing
  - Uploading data only while on wifi
    - During periods of data collection, information should be stored locally and uploaded to the database only when the user is connected to wifi. In cases when the user has locally stored data but has not connected to wifi for three (3) consecutive days, the next time the user opens the app, they will be prompted to upload data by either 1- connecting to wifi or 2- uploading via data
  - API to talk between database and app
    - An API will need to be created to increase security and encryption. This API should contain GET and PUT calls that allow the app and the database to communicate with each other

- Login Page
    - Remember me setting
    - Detect and store the current user's userId from the database across the app
    - Check if username exists
    - Check if password does not match username
- Forgot Password/Username pages
    - Needs functionality
- Sign up pages:
    - If blank fields and click next, app breaks
        - Need checks
    - Data added to database should connect via mobilityAidId and userId (see Sign Up SQL in Appendix)
    - First sign up page (email, username, password): Add verification methods
        - Email field should be a properly formatted email address
        - Email must not already exist in the database
        - Username must not already exist in the database
        - Password should meet the Password Criterion (see Appendix)
        - Should provide relevant, descriptive error messages
    - Conditional third page
        - The wheelchair information page should only show when the selected transportation option is "wheelchair"
            - Wheelchair info will also only be added to the database if the "wheelchair" transportation option is selected
- Sign up and settings edit pages: Allow users to type a value when they select "Other" in a form field
    - Currently, most fields are Xamarin pickers. Future developers will need to add in a check that displays an entry field that users can type in when the "other" option is selected
- Settings pages
    - Logged in user's information from the database should show on the corresponding pages
    - Conditional Transportation Settings page:
        - Page will always show transportation and value. All other fields will only be shown if the selected transportation is "wheelchair"
- Settings edit pages
    - Save button should store changes in database
        - This can be done via a SQL update call to the database
    - If blank fields and click next, app breaks
        - If blank fields when button is pressed, move focus back to first field in error and show appropriate error messages.

    - Conditional Transportation Settings edit page:
        - Transportation field will always be shown. Other form fields will only be shown when the selected transportation is “wheelchair”
- Map page
    - Need step-by-step instructions
    - Bounding box needs to recenter to fit all of route
    - Ability to select a point via screen tap
    - Should display multiple routes
- Data Collection page (Route Data)
    - TripId needs to increment
        - Currently, the SQL code is there, however, the result of the SQL SELECT call is not stored. The tripId should increment for each trip recorded.
    - Date and time is not accurate
        - Currently, the date and time being recorded in the database is not accurate. We assume this is because a dateTime variable is created but the current date and time is not collected.
    - Barrier and facility information should be added to the database
        - A new SQL call will need to be made when a barrier or facility is submitted. The new SQL call will add in information for the optional database barrier or facility columns.
    - TripId value should be based on the current user
        - The first tripId for each userId will be 1. The userId of the currently logged in user will be used to determine the tripId that should be used.
        - Ex: user1 has recorded 5 trips and user2 has recorded 12 trips. User1 records another trip. UserId would be 1 and tripId will be 6.
    - Allow users to deny sensor access (if possible)
        - In the code’s logic there will need to be conditionals added that collect sensor data and include it as part of the database call only if the user has given access to the phone’s sensors
- Sign out
    - Sign out button should remove current user’s userId from app data and redirect to sign in page

Bugs

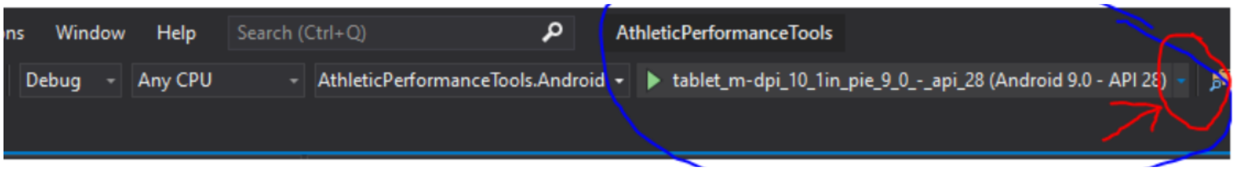
The following are bugs that were noticed that we did not have time to fix.

- The pointOnMap float[] variable in APICalls.cs currently has a set length. When large values for the length were attempted, the app overloaded.

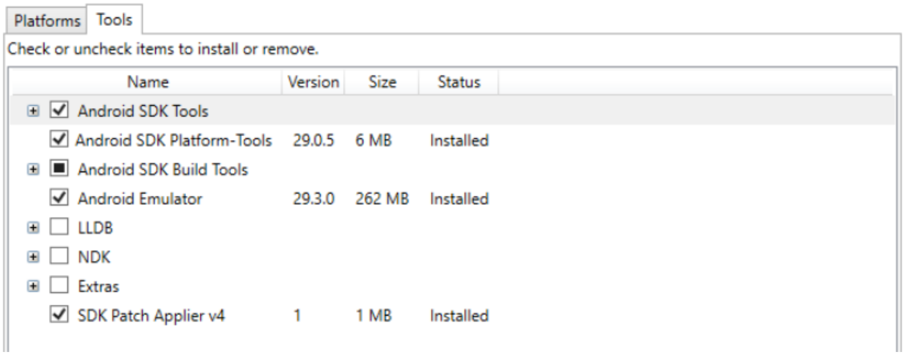
Appendix

Xamarin Installation

1. Install Visual Studio 2019
2. Install Mobile development with .NET (inside visual studio 2019 package installer)
3. Create a new Xamarin.forms project (forms is what we want since we are building on both ios and android)
4. To set up an emulator…
    1. Click the drop down arrow (see image below) and select Android Device Manager


ns
Window
Help
Search (Ctrl+Q)
AthleticPerformanceTools
Debug
Any CPU
AthleticPerformanceTools.Android
tablet_m-dpi_10_1in_pie_9_0_-_api_28 (Android 9.0 - API 28)

    2. Click New and make the base device Pixel 2 (or whatever you want your simulator phone to be)
    3. The OS should be Pie 9.0
5. Go to Tools -> Android -> Android SDK Manager and have all these (see image below) installed under tools (if you don't see Pie 9.0 installed under platforms make sure that gets installed)



Check or uncheck items to install or remove.

Name	Version	Size	Status
⊞ ☑ Android SDK Tools			
☑ Android SDK Platform-Tools	29.0.5	6 MB	Installed
⊞ ■ Android SDK Build Tools			
☑ Android Emulator	29.3.0	262 MB	Installed
⊞ ☐ LLDB			
⊞ ☐ NDK			
⊞ ☐ Extras			
☑ SDK Patch Applier v4	1	1 MB	Installed

6. Where we clicked the drop down menu, select your new emulator (should say something like Pie 9.0) then click run
    1. The first time you run it it can take up to 5 minutes to get the emulator up and going
        i. Every time after it it will take significantly less time
        ii. If you get a phone with a black screen that's fine it's just booting up

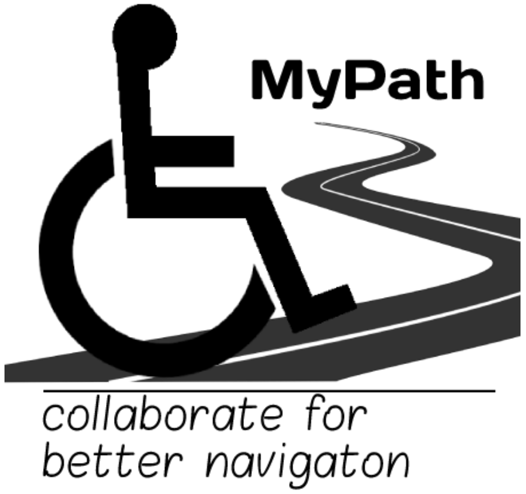
Sign Up SQL

```
// get last added userID
// the result of this should go in the Users.cs anywhere userID is called,
// so in UserInfo_Model, the 4 after "@param1" would be the result of this select
SELECT userId
FROM Users
ORDER BY userId DESC
LIMIT 1

// after mobility aid is added, we want to update the mobilityAidId for
// the entry we just put into UserInfo

// get the last added mobilityAidId
SELECT mobilityAidId
FROM WheelchairInfo
ORDER BY mobilityAidId DESC
LIMIT 1

// updates mobilityAidId in UserInfo table
UPDATE UserInfo
SET mobilityAidId = value1 //value1 is the result from the above select statement
WHERE userId = value2 //value2 is the result from the first select statement
```

Password Criterion

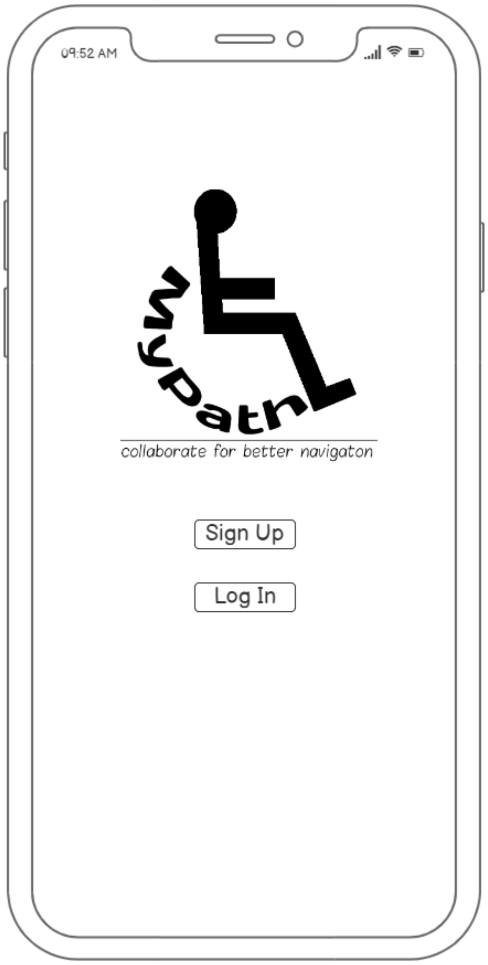
- Contains at least of each of the following: uppercase letter, lowercase letter, number
- At least 8 characters long
- Does not contain the username or parts of the username that are >= 3 characters long

Logo

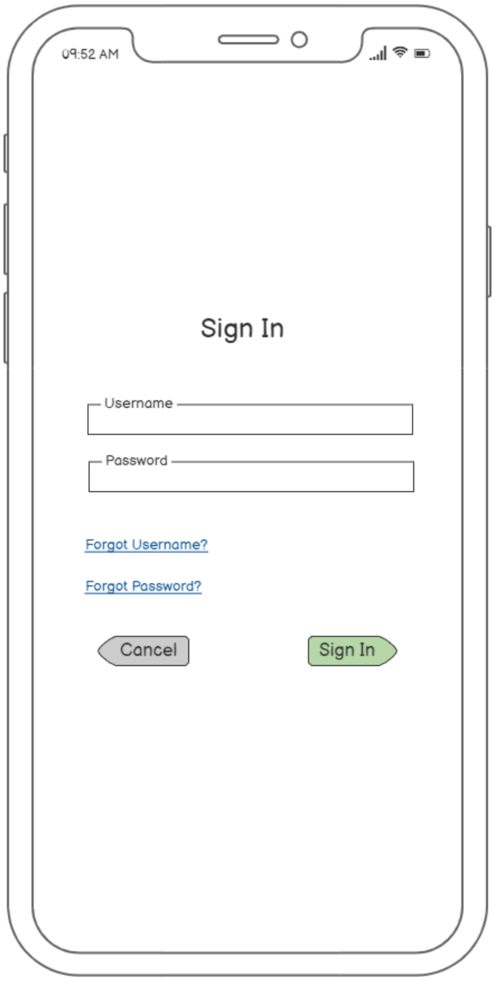


Mockups

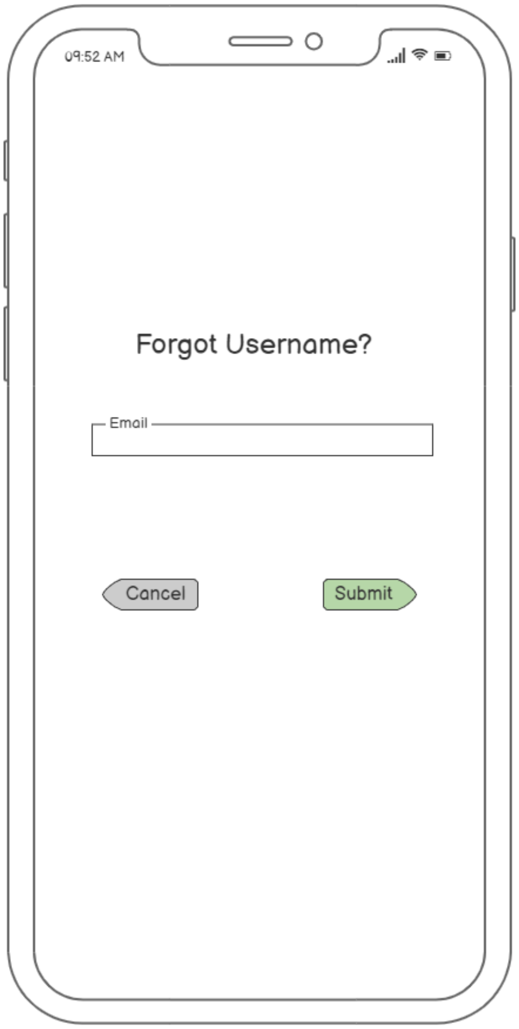
Landing page



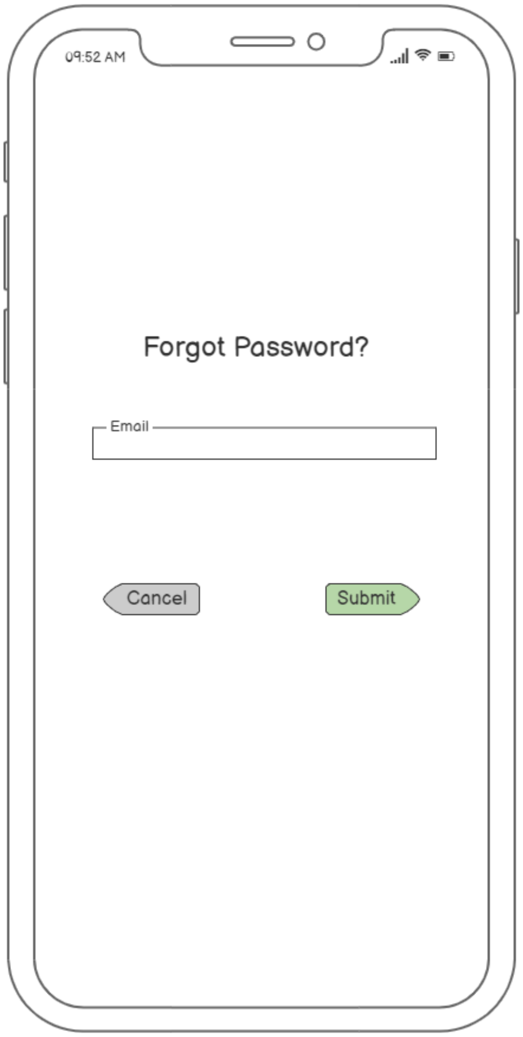
Sign In

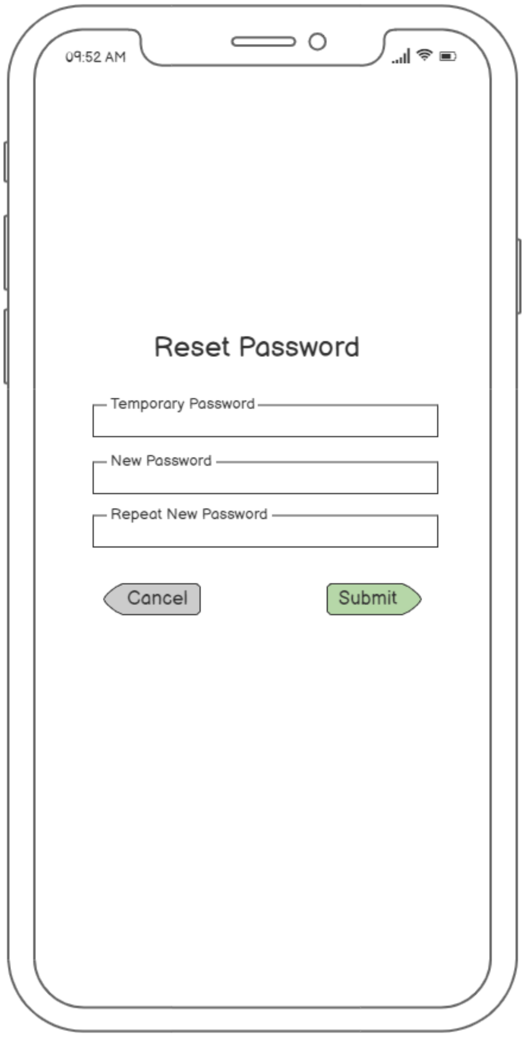

09:52 AM
Sign In
Username
Password
Forgot Username?
Forgot Password?
Cancel
Sign In

Forgot Username

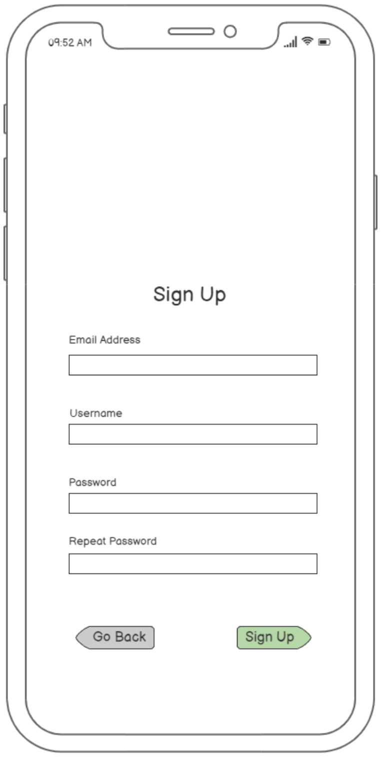


Forgot Password





Sign Up



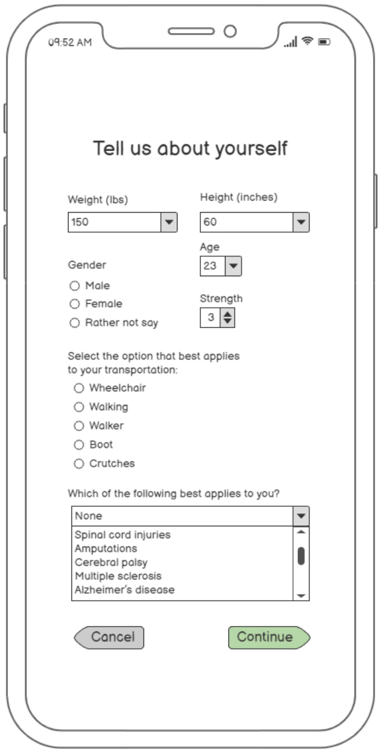
Sign Up

Email Address

Username

Password

Repeat Password



Tell us about yourself

Weight (lbs) 150

Height (inches) 60

Age 23

Gender

○ Male
○ Female
○ Rather not say

Strength 3

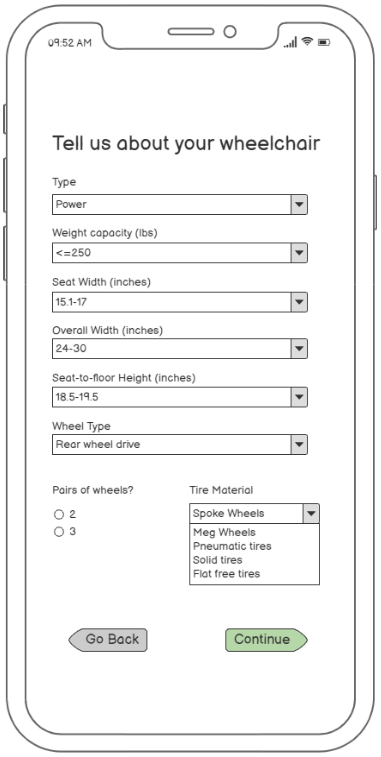
Select the option that best applies to your transportation:

○ Wheelchair
○ Walking
○ Walker
○ Boot
○ Crutches

Which of the following best applies to you?

None

Spinal cord injuries
Amputations
Cerebral palsy
Multiple sclerosis
Alzheimer's disease



Tell us about your wheelchair

Type: Power

Weight capacity (lbs): <=250

Seat Width (inches): 15.1-17

Overall Width (inches): 24-30

Seat-to-floor Height (inches): 18.5-19.5

Wheel Type: Rear wheel drive

Pairs of wheels?

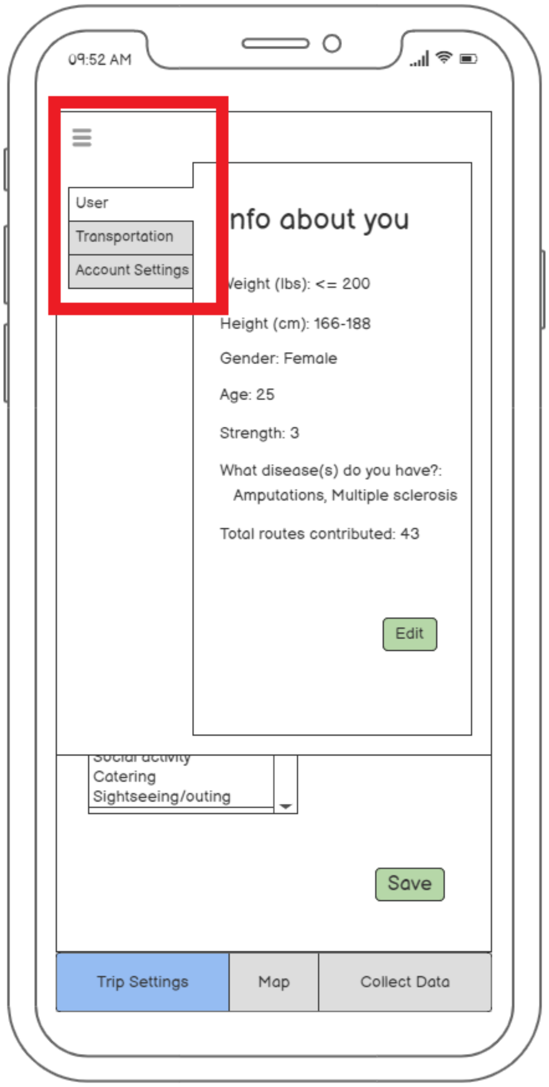
○ 2
○ 3

Tire Material

Spoke Wheels

Meg Wheels
Pneumatic tires
Solid tires
Flat free tires

Trigger Menu



nfo about you

Veight (lbs): <= 200

Height (cm): 166-188

Gender: Female

Age: 25

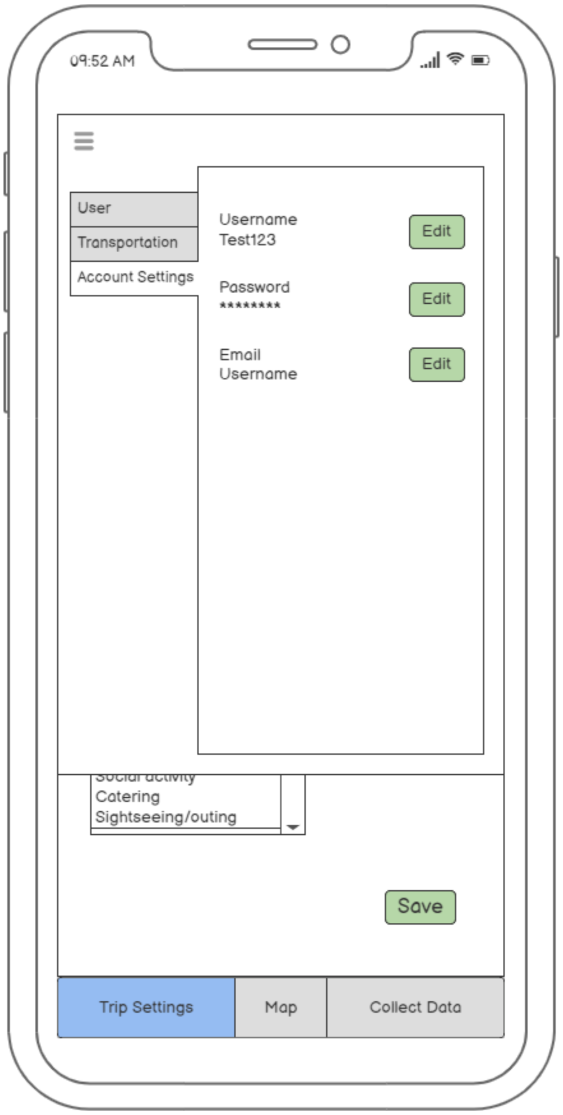
Strength: 3

What disease(s) do you have?:
Amputations, Multiple sclerosis

Total routes contributed: 43

Social activity
Catering
Sightseeing/outing

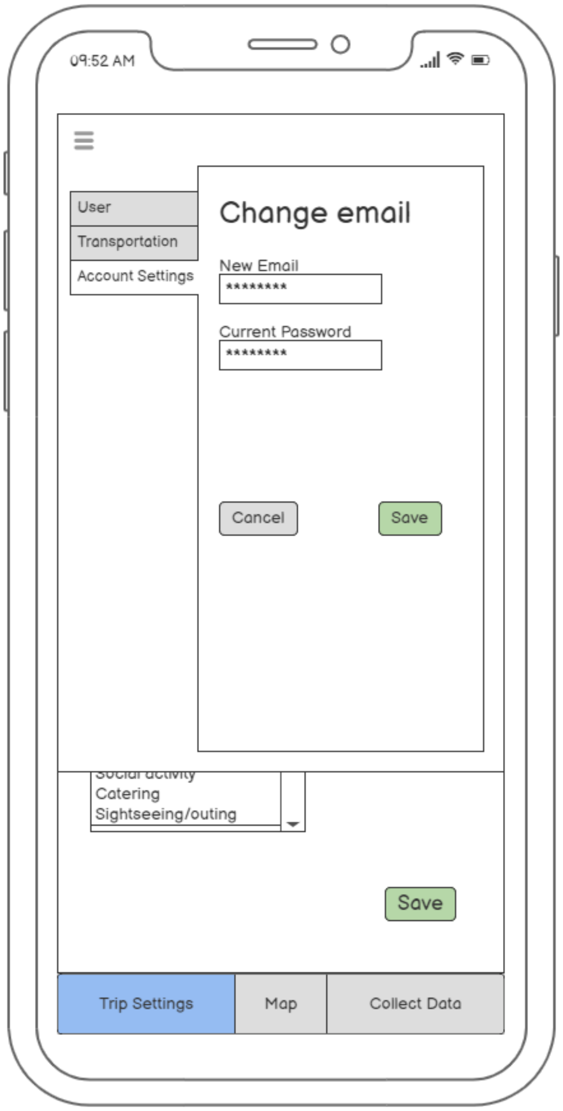
Account Settings



Username
Test123

Password
****

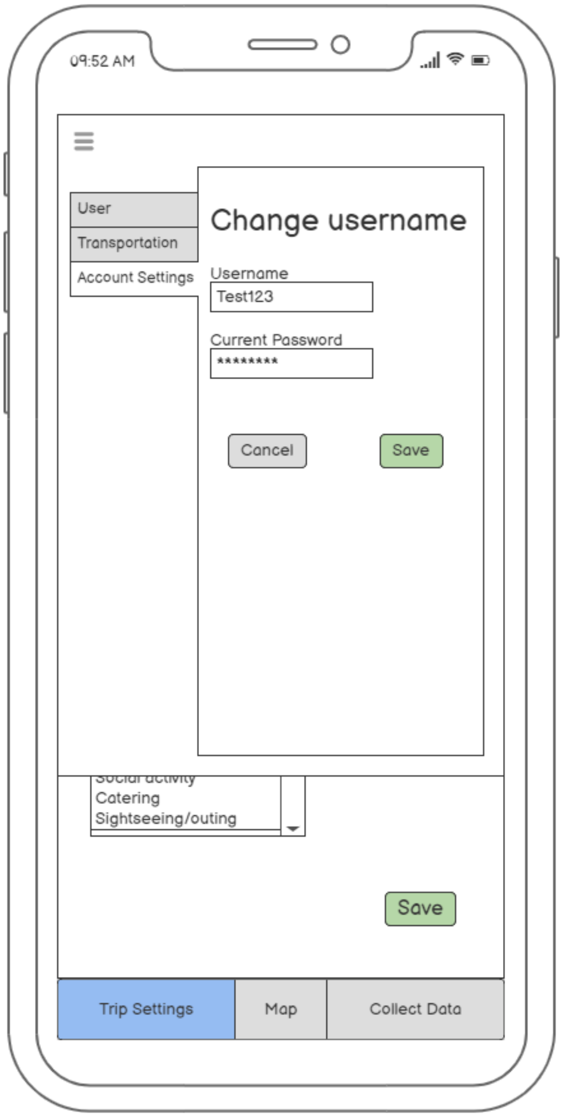
Email
Username



Change email

New Email
****

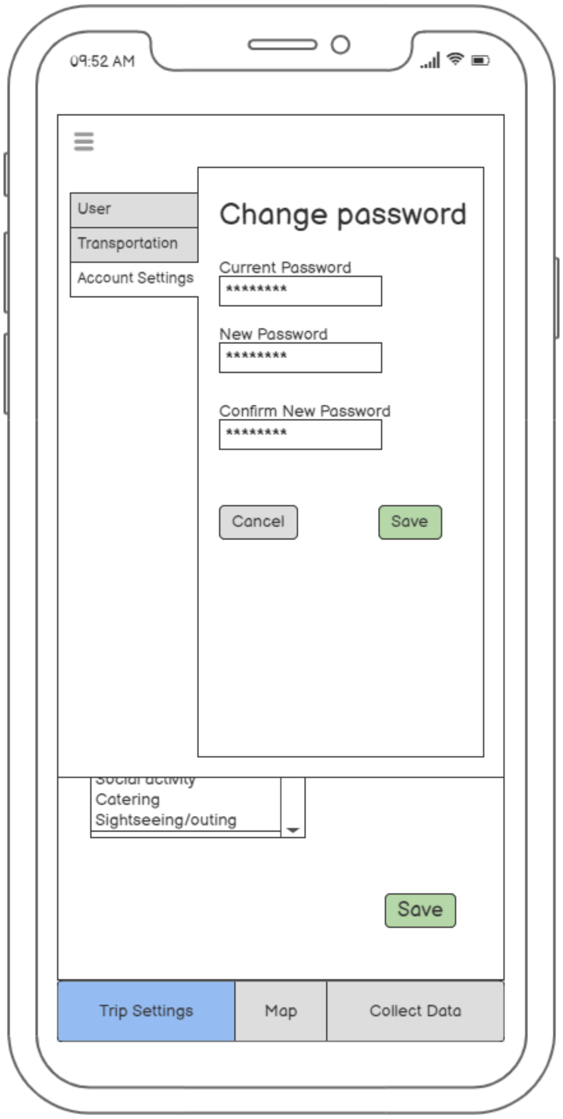
Current Password
****



Change username

Username: Test123

Current Password: ********



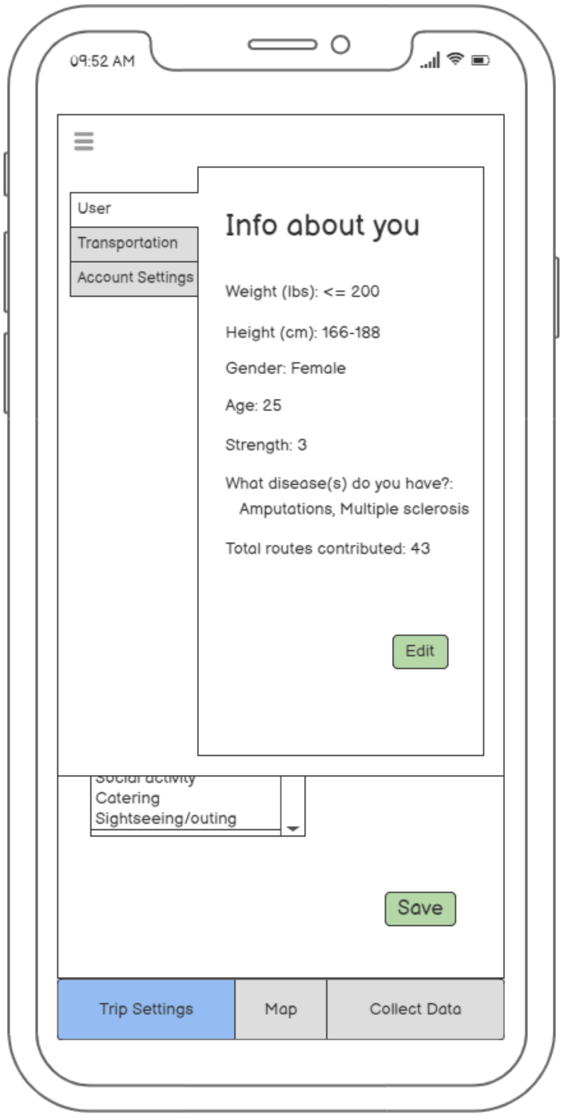
Change password

Current Password: ********

New Password: ********

Confirm New Password: ********

User Settings



Info about you

Weight (lbs): <= 200

Height (cm): 166-188

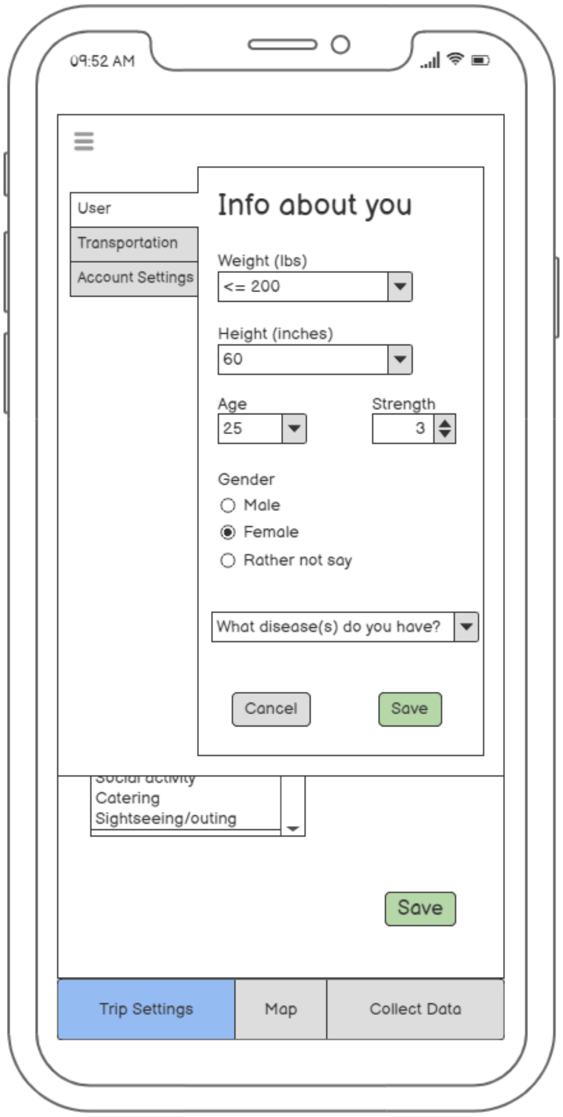
Gender: Female

Age: 25

Strength: 3

What disease(s) do you have?:
Amputations, Multiple sclerosis

Total routes contributed: 43



Info about you

Weight (lbs): <= 200

Height (inches): 60

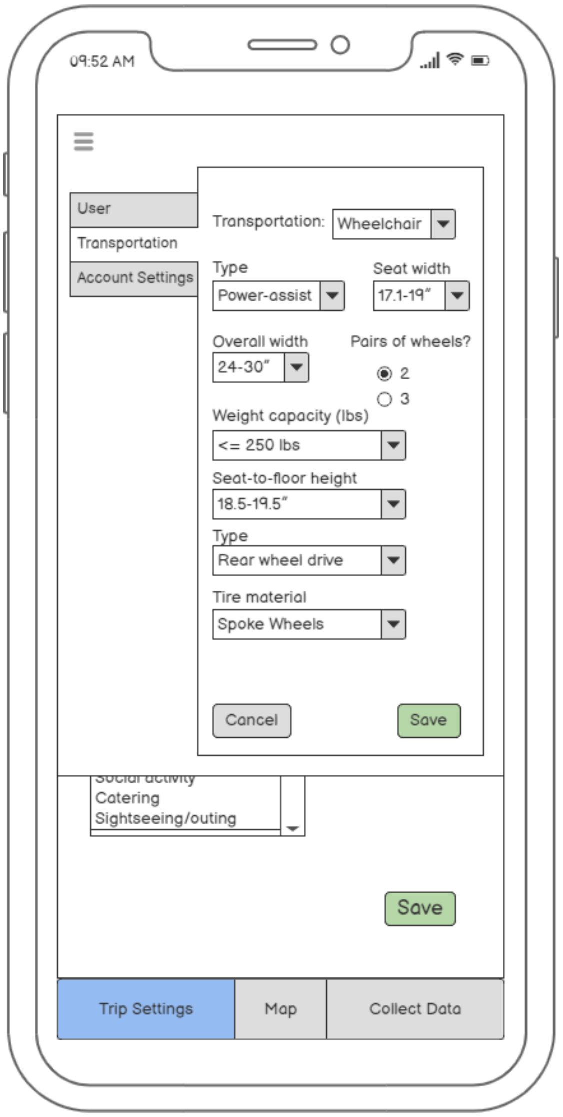
Age: 25

Strength: 3

Gender
○ Male
● Female
○ Rather not say

What disease(s) do you have?

Transportation Settings



Transportation: Wheelchair

Type: Power-assist

Seat width: 17.1-19"

Overall width: 24-30"

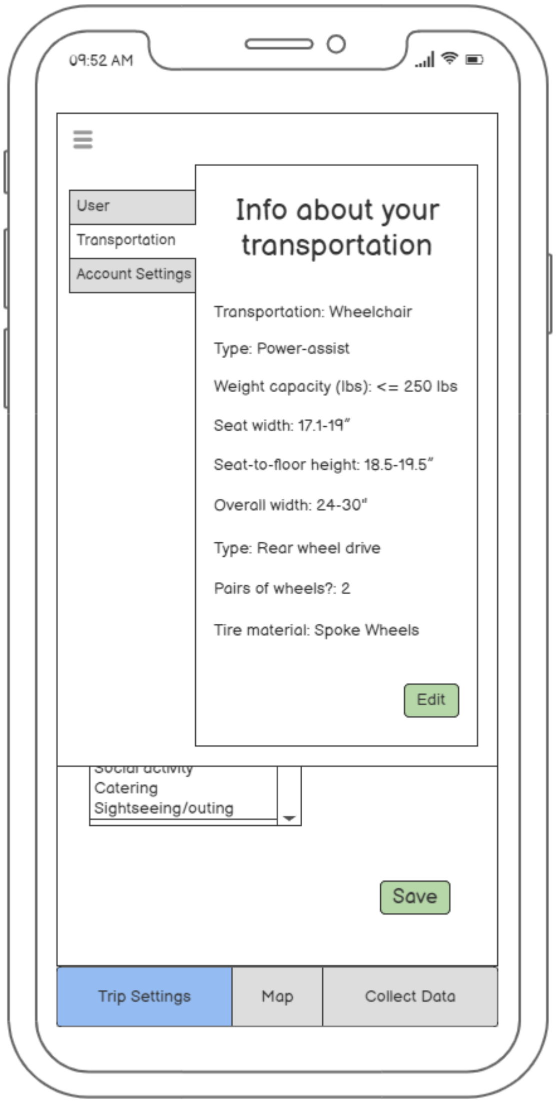
Pairs of wheels?
● 2
○ 3

Weight capacity (lbs): <= 250 lbs

Seat-to-floor height: 18.5-19.5"

Type: Rear wheel drive

Tire material: Spoke Wheels



Info about your transportation

Transportation: Wheelchair

Type: Power-assist

Weight capacity (lbs): <= 250 lbs

Seat width: 17.1-19"

Seat-to-floor height: 18.5-19.5"

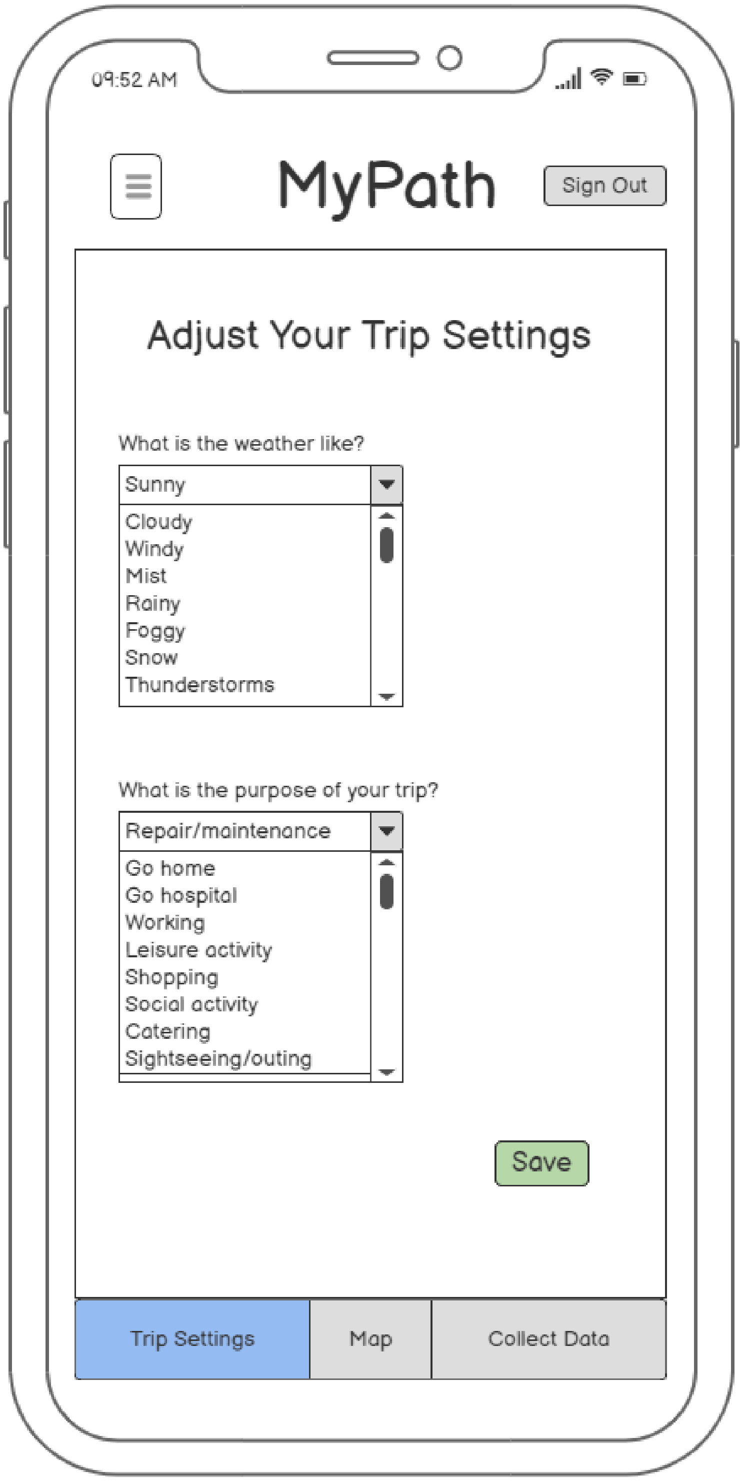
Overall width: 24-30"

Type: Rear wheel drive

Pairs of wheels?: 2

Tire material: Spoke Wheels

Trip Settings



Adjust Your Trip Settings

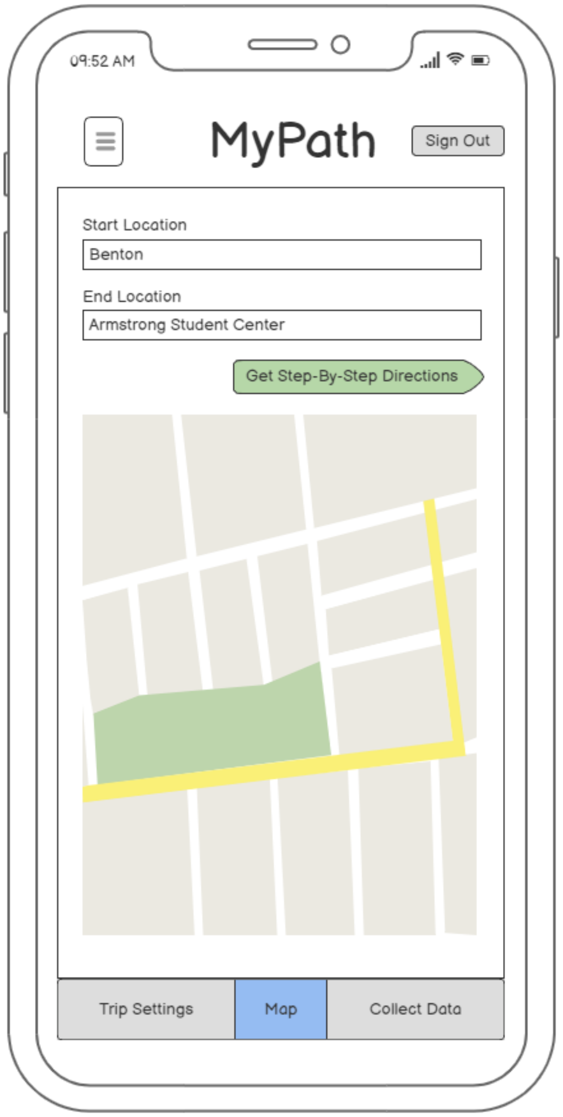
What is the weather like?

Sunny

What is the purpose of your trip?

Repair/maintenance

Map Page

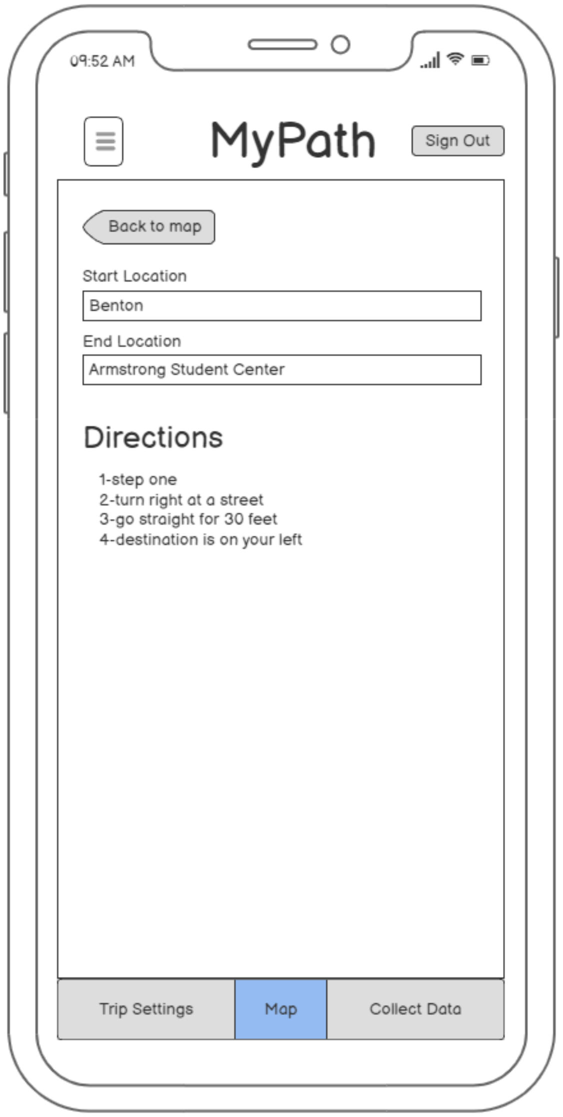


Start Location

Benton

End Location

Armstrong Student Center



Start Location

Benton

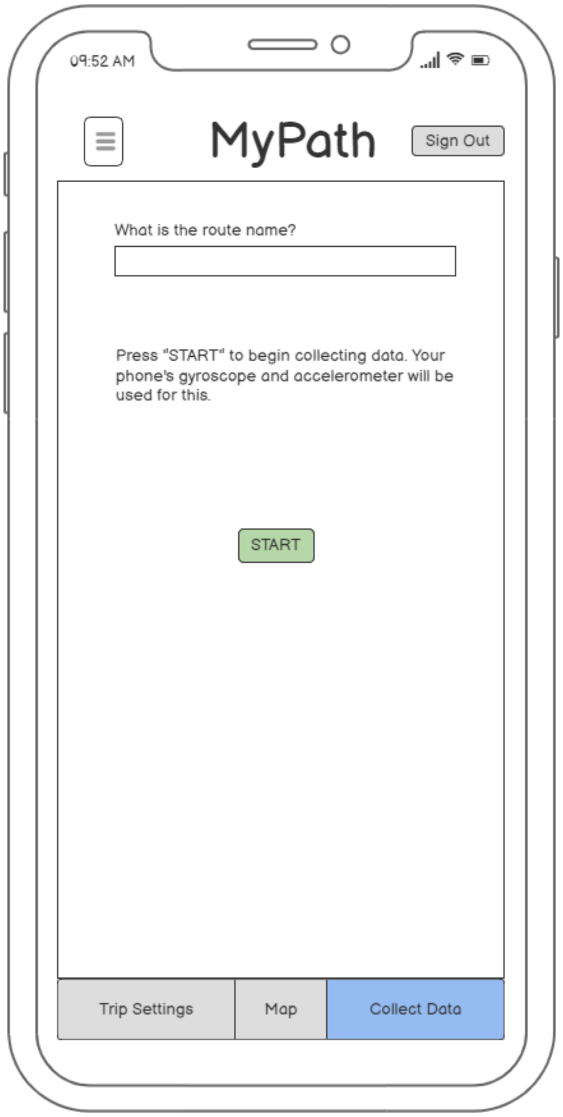
End Location

Armstrong Student Center

Directions

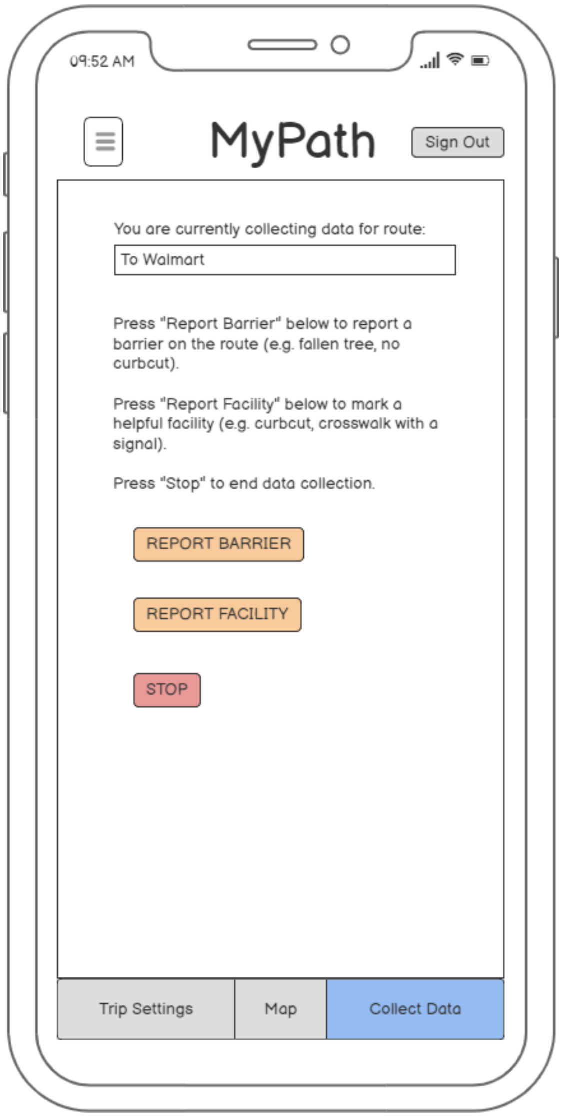
1-step one
2-turn right at a street
3-go straight for 30 feet
4-destination is on your left

Data Collection Page



What is the route name?

Press "START" to begin collecting data. Your phone's gyroscope and accelerometer will be used for this.



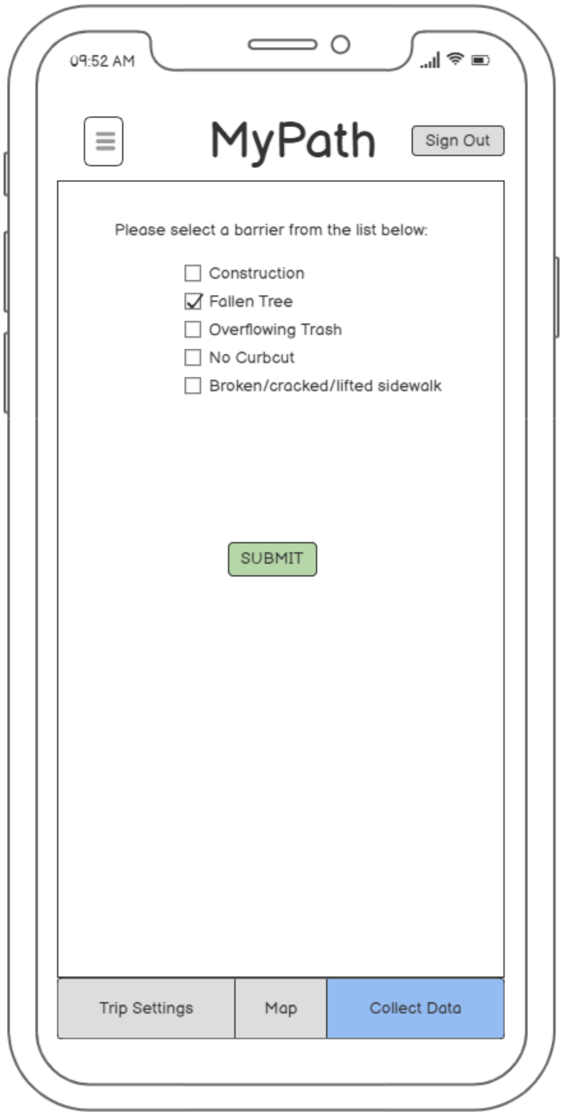
You are currently collecting data for route:

To Walmart

Press "Report Barrier" below to report a barrier on the route (e.g. fallen tree, no curbcut).

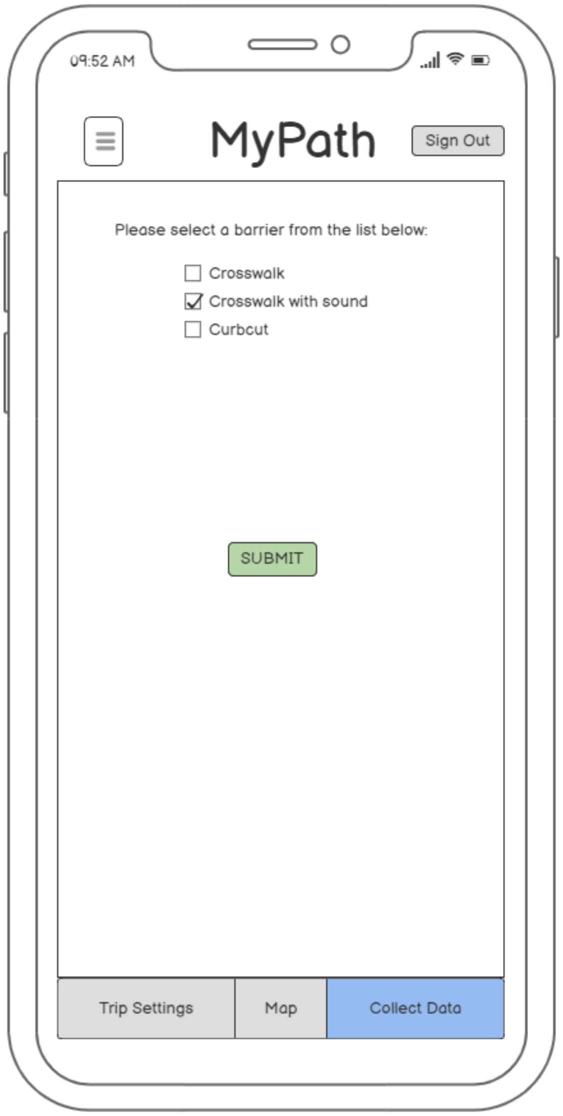
Press "Report Facility" below to mark a helpful facility (e.g. curbcut, crosswalk with a signal).

Press "Stop" to end data collection.



Please select a barrier from the list below:

- ☐ Construction
- ☑ Fallen Tree
- ☐ Overflowing Trash
- ☐ No Curbcut
- ☐ Broken/cracked/lifted sidewalk



Please select a barrier from the list below:

- ☐ Crosswalk
- ☑ Crosswalk with sound
- ☐ Curbcut