College of Engineering
WEST TEXAS A&M UNIVERSITY

Walks - Outdoors GPS Tracking Software Application

Written by:

Austin R. Turner
Kerighan E. Wheeler
Lance G. Fletcher
Spencer R. Parton
Wade R. Carter

Supervised by:

Mohammad F. H. Siddiqui, Ph.D.

Version 8.0

Publish date:

April 26th, 2022

TABLE OF CONTENTS

Software Requirement Specifications

1. Introduction

1.1 Purpose

According to the National Park Service, approximately 2000 individuals get lost hiking every year. Additionally, hundreds of people participating in outdoor recreation activities are injured,or deal with complications while outside of the range of cell phone service. This oftentimes means individuals are unable to get help when they need it most. In response to the ever-present dangers that accompany many popular outdoor adventures, The Walks Outdoor GPS Tracking Software Application is a proof-of-concept project that will provide users with an additional layer of safety and support through continuous GPS tracking, and personalized alert systems outside of areas with cell phone reception.

1.2 Document Conventions

This document was created based on the template for IEEE SRS (System Requirement Specification Documents).

1.3 Intended Audience

Anyone potentially traveling outside of range of cellular service, and who would like to be able send notifications to others in case of distress. Any individual or organization desiring to monitor and track travel progress of people or equipment in or outside of range of cell phone service

1.4 Product Scope

The Walks outdoor adventure tracking software is a proof-of-concept tool intended to help mitigate the dangers associated with people and equipment traveling in remote locations away from cell phone service. It will allow users to integrate location data from different GPS tracking devices on one platform where they can track location, set geo-fencing points, manage notifications, as well as other data associated with planning a trip.

1.5 Definitions, Acronyms, and Conventions Used

- User - Anyone with a Walks account
- Spectator - User with viewing privileges of a trip
- Participant - User listed under a specific trip, automatically has spectator privileges

- Trip Manager - User and spectator of a trip, also has administrative privileges for trip (ie can rename, add/edit members, geo-fencing etc) Trip managers could also be participants and device holders, but not necessarily.
- Device - Tracking capable instrument which can send data to Walks for tracking purposes.
- Device holder - A trip participant handling the GPS device, automatically has spectator privileges
- Trip - Defined by a range of dates, group members, and at least one GPS device. Trips are managed by a defined trip manager, and can be viewed by any users added to the trip.
- Assets - Device attached to an object or equipment
- Marker - A specific point on the map which can be dynamically placed and deleted by users that have permission.
- Geofence - An area on the map which can be dynamically defined and deleted by users that have permission.

2. Overall Description

2.1 Product Perspective

Walks is intended for anyone that is wanting to track location data of an outdoor trip whether it goes beyond cell serviced areas or not.

2.2 Product Functions

2.2.1 Landing page

- Login in button - Will route the user to a login page
- Register (Create Account) button - User will be routed to an account creation page
- About information - This will be at the bottom of the webpage and will display information about the purpose of the application and the project team

2.2.2 Login page

- Login box - Will allow users to enter credentials (username and password) to sign in
- Don't Have an Account Link - Redirect users to the Register Page

2.2.3 Register page

- Registration box
  - Email
  - First Name
  - Last Name
  - UsernamePassword
  - Emergency Contact Name
  - Emergency Contact Number

2.2.4 Navigation Bar (Top)

- Create a trip - Routes users to Create a Trip Page.

- Walks logo (Map) - If not logged in this button will direct users to the landing page. Otherwise it will send users to the Dashboard Page.

- Profile - Routes users to their Profile Page.

- Add Device - Allows users to register a device to their account so it can be added to trips in the future.

- Notifications - Shows users friend requests or trip invites.

- Logout - Logs user out of application, redirects them to sign in page and blacklists access token.

2.2.5 Dashboard

- Active trips - Active trips will be a category of trips that are ongoing. Users can click into each active trip to be able to see tracking info

- Future trips - Displays information about upcoming trips. Information will include participant roster, route to be taken and date of trip

- Past trips - Displays information about previously taken trips. Information will include participant roster, route taken and date of trip

- Public Trips Button - Takes users to Public Trip Page.

- Clicking on any trip window will open up the trip in the Trip Management Page

2.2.5 Trip Management page

- Trip Name Drop Down Box - Clicking this will open a drop down box with 3 tabs

  - Device Tab - Shows all current devices on a trip

    - +/- Person Devices Button - Will open a popup display of all the users previously registered devices and let them check which ones to add to the trip.

        - Device Card List - Shows all devices that have been added to a trip. Clicking a Device Card will focus the map on the corresponding device marker location.
    - Markers Tab - Allows trip Managers to add/remove/view location of markers on a trip
        - + Button - Lets a user add and name a marker to the map
        - - Button - Lets a user remove a marker from the map
        - Checkbox - Toggles the makers being show on the map
    - Geofence Tab - Allow users to add and toggle viewing the Geofence
        - + Button - Allows users to add a polygonal geofence. User must select 4 points before further adjustment can be made.
        - - Button - Allows users to remove a Geofence
        - Checkbox - Toggles showing the geofence on the map
        - Target - Will focus the map on a users Geofence.
- Trip Information Tabs
    - Participant Roster Tab - Users can see a list of all the participants that are attending the trip and have the ability to delete participants from the trip if needed
    - Spectators Roster Tab - Shows a separate list of users who are only viewing the trip rather than participating.
    - Details Tab - Shows the temperature of the intended trip location, the date range for the trip, the trips description, as well as the total number of participants and spectators associated with the trip.
- Delete/Leave Trip Button - Users, depending on role, are able to delete the trip (if the user is a Trip Creator or Trip Manager) or leave the trip (if the user is a regular Participant or Spectator)

2.2.6 Create a Trip page

- Name - This is be the name of the trip. Validation restricts input to 15 characters or less
- Description - This will allow the Trip Creator to list a more detailed description of the trip

- Start Date - What is the start of the date range that this trip will take place. This will be used to sort the trip between future/active/past trip categories in the dashboard.

- End Date - What is the end of the date range that this trip will take place. This will be used to sort the trip between future/active/past trip categories in the dashboard.

- Invite Users - Here the Trip Creator will add a member to invite other users to be added to the trip. Users must be friends first to add each other to trips.

- Add devices - The Trip Creator will select which, if any, of their previously registered devices will be taken on the trip.

- Make Public Checkbox - By checking the "*Do you want to make the trip public to friends*" checkbox the trip will be added to the public trip page making it available for all users friends to add themselves to the trip

- Permissions - Upon clicking submit, if the users added members to the trip, the user will be able to assign permissions/roles for the attending participants. Permissions for a manager will include: Altering route, adding/removing participants, adding/removing devices, setting geofence zone, manage notifications, and change device nicknames.

2.2.7 Profile page

- Personal account info - User's account profile picture, first name, last name, short biography will be displayed.

- Edit Account Button - Users can alter their first name, last name and bio here.

- Device Management - Clicking on in a device box will allow users to edit device name or remove the device from their account. Clicking this button will open a popup box.

2.2.9 Device Registration page

- GPS Name - User will specify GPS name they want to show up on account

- GPS Brand - User will specify GPS Brand

- GPS Serial Number - User will input GPS Serial Number

- GPS API Key - Users will input what their device API key is

- GPS Api Secret - Users will input what the device API secret is if needed

- Need Help? Button - Opens a new tab and sends users to the help page for adding a device.

2.2.10 Notifications Page

- Message - Invite will display a default invitation message inviting the user to Trip Name or a default Friend Request invitation
- Accept/Decline button - Two buttons will be shown at the side of the invitation box with options to either accept or decline the trip invitation. If accepted the user will then be added to the trip roster or the users friends list depending on the notification.

2.2.11 404 Page Not Found

Simple 404 page telling the user the desired page cannot be found

2.3 User Class and Characteristics

All permissions are trip specific and do not carry over from one trip to the other.

- Trip Creator - Defines the user who initially creates the trip. This user will by default have all available privileges and will have the ability to grant other users privileges.
- Trip Manager - Defines the user who will be primarily in charge of managing the trip details. A Trip Creator and a Trip Manager could be different users. Trip Manager can be assigned by Trip Creator. Trip Managers will be granted full privileges the same as the Trip Creator
- Spectator - Defines a user who is not going on the trip but will be observing the trip's progress. A Spectator will have no inherently granted permissions. Additionally, GPS Devices cannot be assigned to this member
- Participant - Defines a regular trip member. No permissions will be automatically granted to participants.
- The permissions for all of User classes mentioned above can be altered by the Trip Manager/Creator. Trip Creator, Manager, Spectator, and Participant are only default user roles to define default user permissions.

2.4 Operating Environment

2.4.1 Support for Multiple Browsers

The user will be able to use the webpage on multiple platforms: Firefox, Internet Explorer, Chrome, and Safari.

2.4.2 Devices

There is a mobile app that will allow users only to designate their device as a tracking device with Walks

2.5 User Class and Characteristics

The backend of the web application has been developed using python and will utilize the web development framework Django. The frontend has been developed using JavaScript by deploying the React.js framework. The database portion of the application will make use of SQLite to store all necessary data. The current plan is to host the web server with a cloud service such as Azure or AWS. Additionally, there are multiple third-party API's implemented throughout the application such as Trakopolis, Google Maps, and a weather API. The overall functionality of the application is dependent on these various frameworks, services, and API constraints.

2.6 Operating Environment

User's Should:

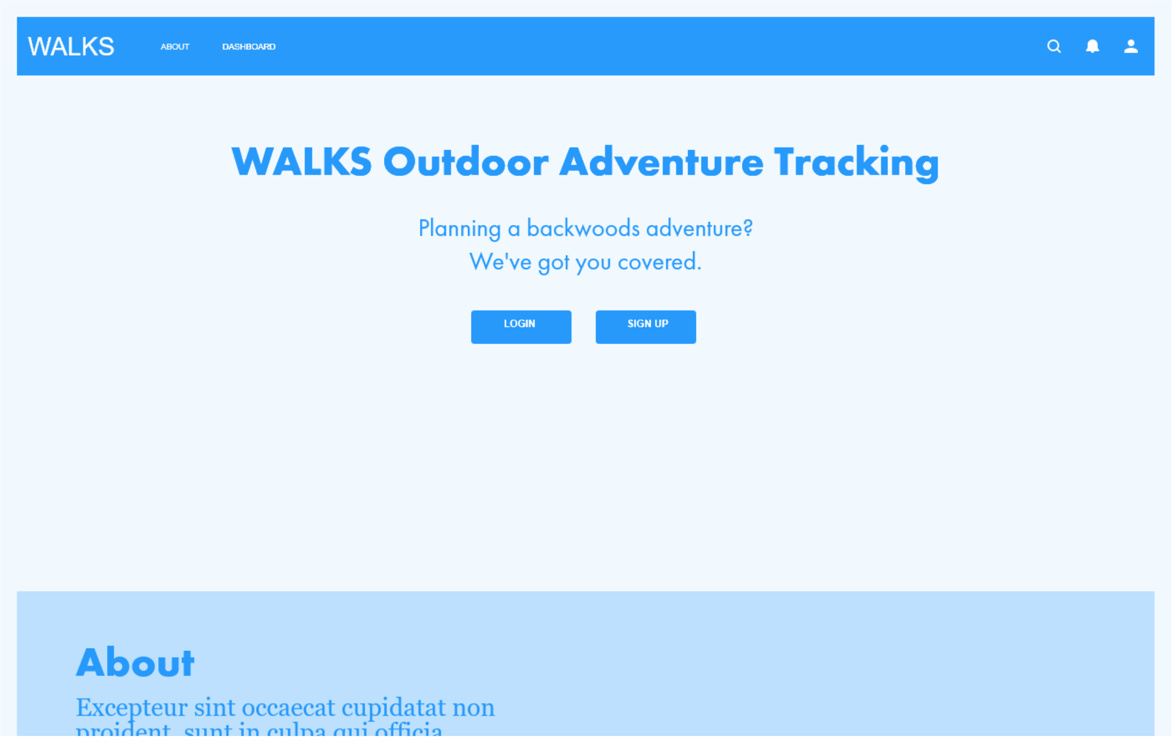
- Be familiar with the use of digital maps.
- Establish and maintain an internet connection.
- Use a compatible web browser.
- Have or be in a group with someone who has an active GPS device registered.

3. External Interface Requirements

3.1 Use Cases

The following are wireframe images that give a general idea of additional functions of the Walks application that will be implemented this semester.

3.1.1 Wireframe Main Landing Page:



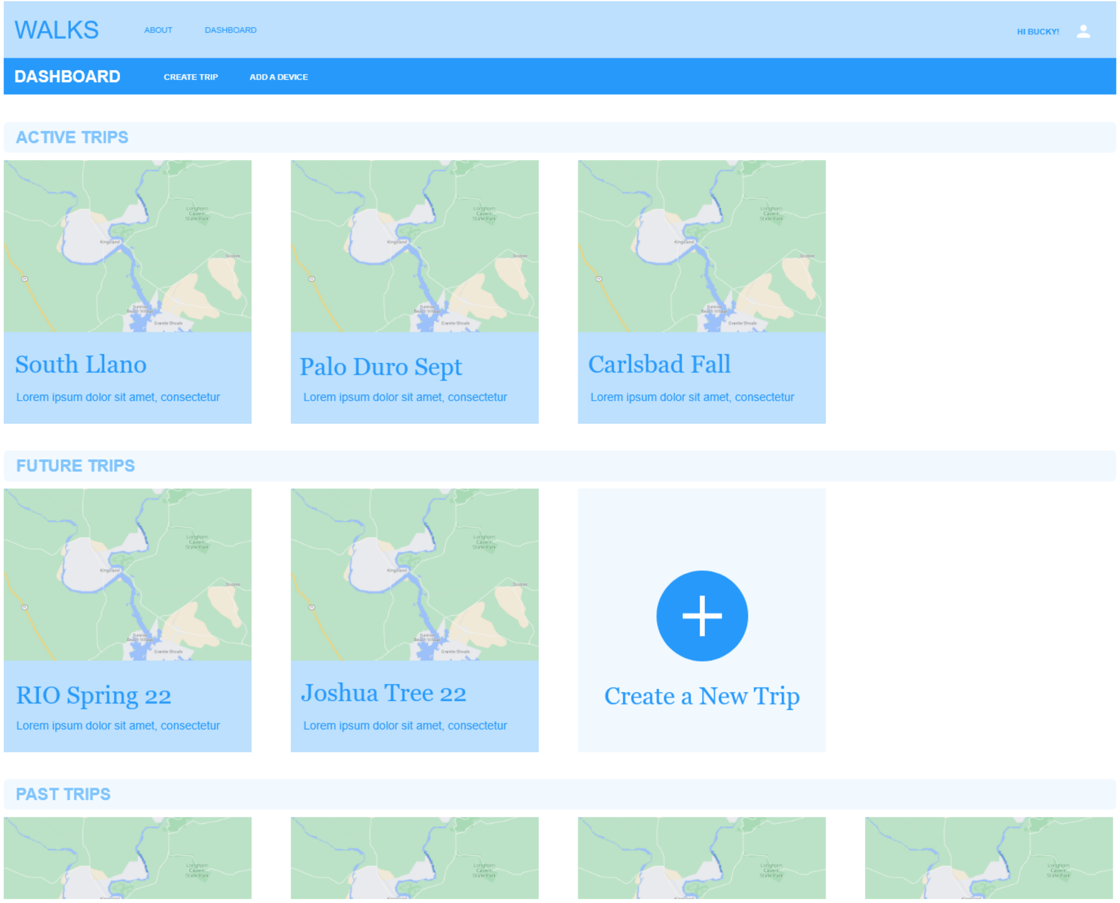
WALKS Outdoor Adventure Tracking

Planning a backwoods adventure?
We've got you covered.

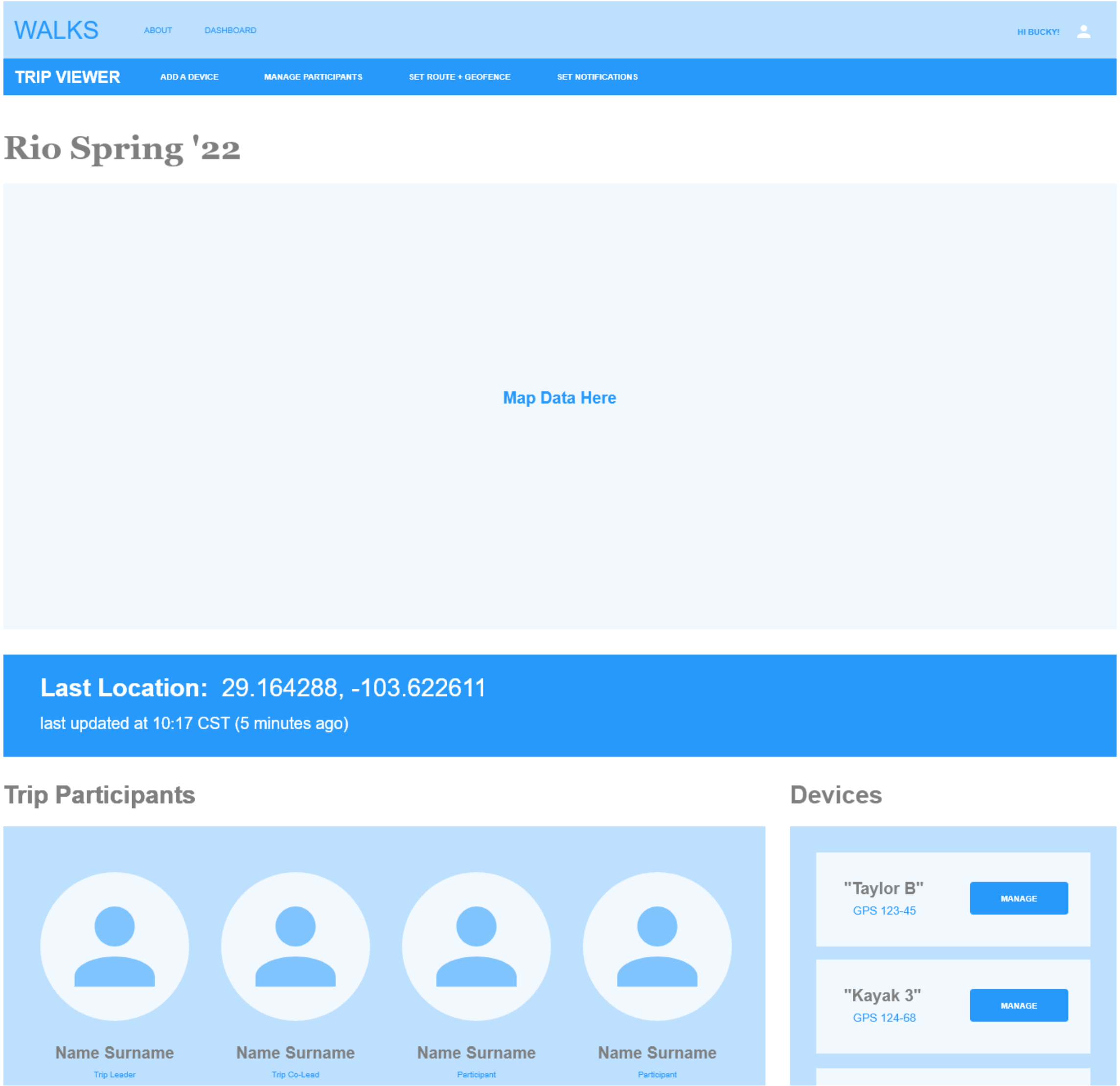
About

Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia

3.1.2 Wireframe Main User Dashboard:


WALKS
ABOUT
DASHBOARD
HI BUCKY!
DASHBOARD
CREATE TRIP
ADD A DEVICE
ACTIVE TRIPS
South Llano
Lorem ipsum dolor sit amet, consectetur
Palo Duro Sept
Lorem ipsum dolor sit amet, consectetur
Carlsbad Fall
Lorem ipsum dolor sit amet, consectetur
FUTURE TRIPS
RIO Spring 22
Lorem ipsum dolor sit amet, consectetur
Joshua Tree 22
Lorem ipsum dolor sit amet, consectetur
+
Create a New Trip
PAST TRIPS

3.1.3 Wireframe Main User Dashboard:



Rio Spring '22

Last Location: 29.164288, -103.622611

last updated at 10:17 CST (5 minutes ago)

Trip Participants

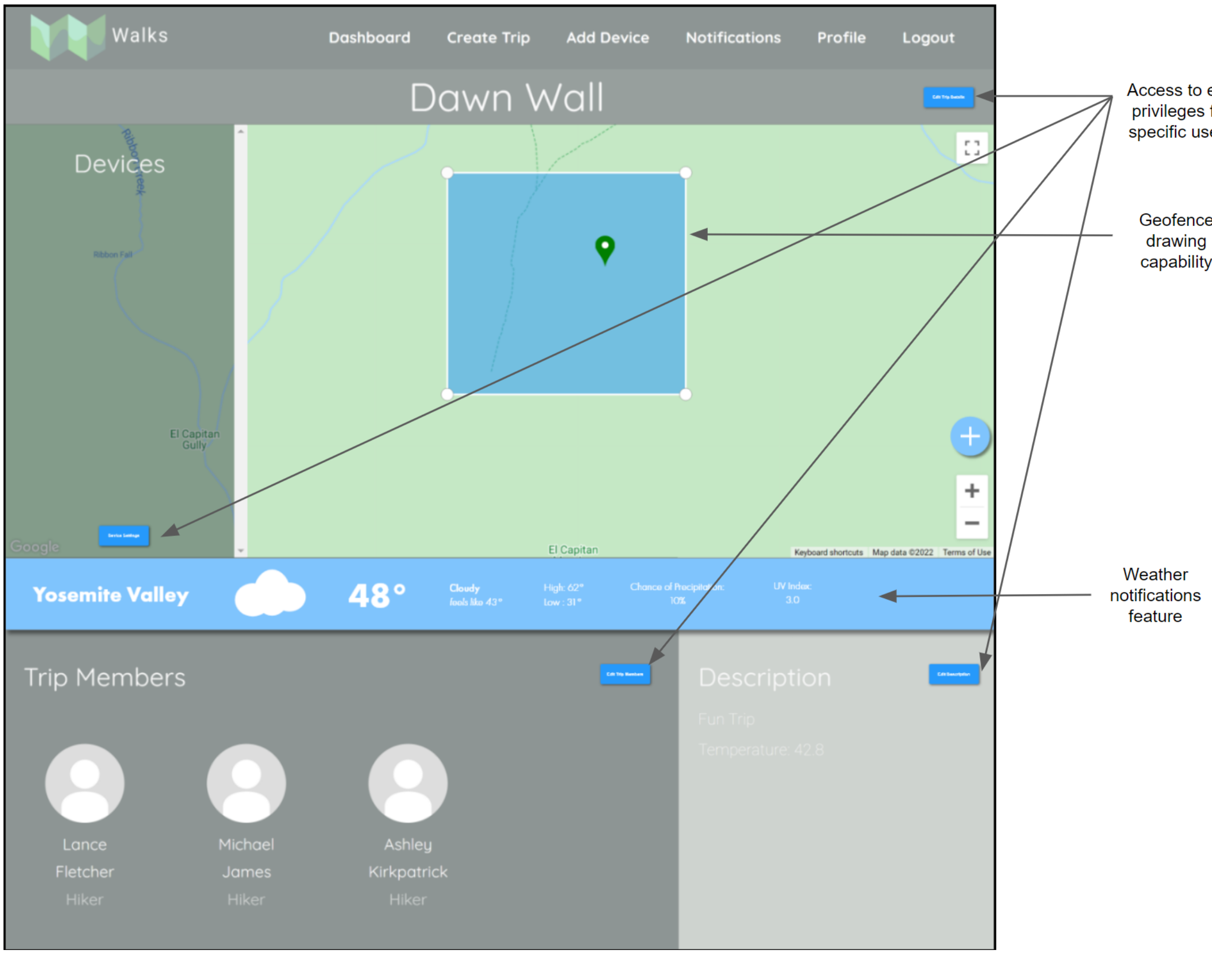
Name	Role
Name Surname	Trip Leader
Name Surname	Trip Co-Lead
Name Surname	Participant
Name Surname	Participant

Devices

Device	GPS
"Taylor B"	GPS 123-45
"Kayak 3"	GPS 124-68

3.1.4 Wireframe overlaying current trip manager page:

This wireframe overlay (blue), highlights the key features that will be implemented this semester:



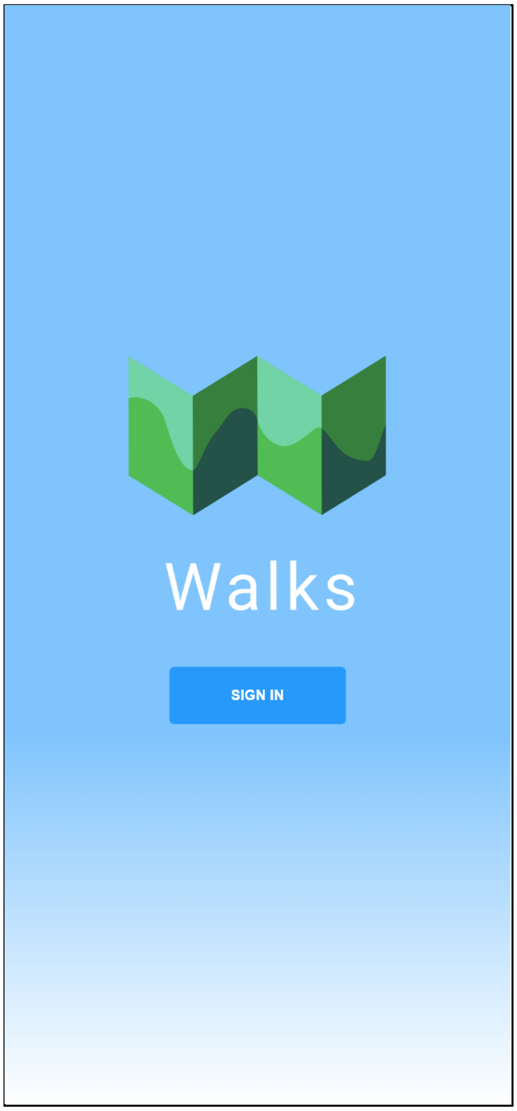
Access to edit privileges for specific users

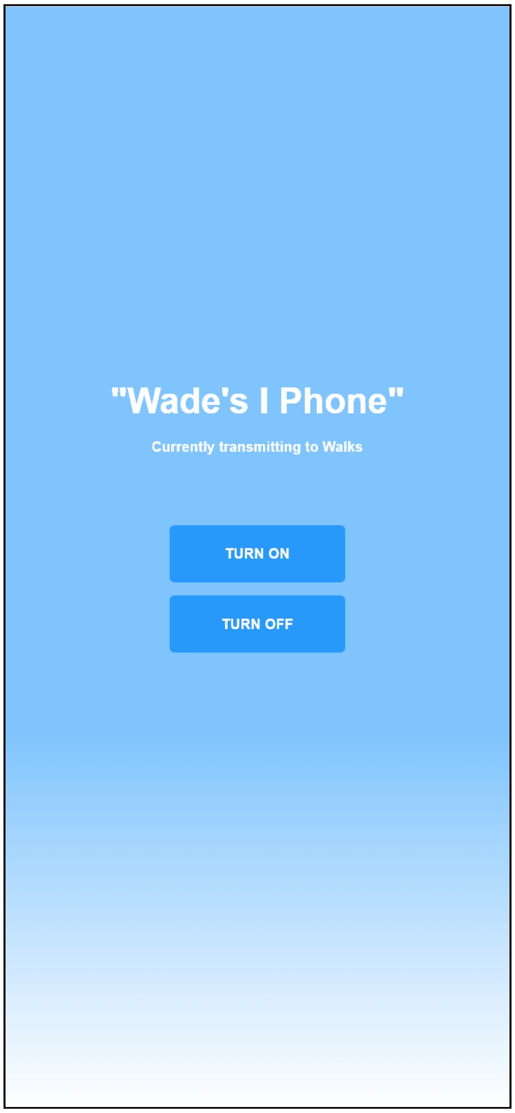
Geofence drawing capability

Weather notifications feature

3.1.5 Wireframe for mobile access:

This will allow users to connect their phone to their Walks account, allowing their phone to transmit data to the Walks application when within range of cell phone service.





3.2 Hardware Interfaces

To be able to track an individual or group of people on a trip, it is necessary for at least one person on the trip to have a fully functioning GPS device. The device should be upright and facing towards the sky and must not be obstructed by any objects or materials.

3.3 Software Interfaces

Walks requires the user to have access to an up-to-date internet browser such as Firefox, Safari, Chrome, etc. If a user is wanting to use a GPS location device such as a Garmin inReach they will need to obtain the proper subscription to allow Walks to properly register the device with the application. Many devices require paid subscription plans for the user to be able to retrieve GPS data.

3.4 Communications Interfaces

Walks will require an internet connection for the user who is spectating the trip. The individuals on the trip must have GPS devices which are capable of satellite connection to properly triangulate the devices position.

4. System Features

4.1 Create Account and Manage Devices

4.1.1 Creating Accounts

Users of the Walks application are able to set up an individual account within the application. This account stores personal information such as email address, phone number, as well as data about their GPS devices that will be associated directly with their profile. Users need to create a unique username and password with a unique email in order to set up an account. Once users have created an account with Walks, they are then able to sign in to the application to access other features of the application such as the personal profile page where users can change their profile picture, bio, and devices linked to their account.

4.1.2 Managing Devices

Users with an activated account within the Walks application can add GPS devices to their profile. GPS devices can only be associated with a single profile, but users can have multiple devices from different manufacturers associated with their account. Users will be able to add and manage their GPS devices using the toolbar at the top of the dashboard and trip pages, as well as from their user profile and settings pages. The specific setup process is unique to the type of GPS device the user would like to interface.

4.2 Create and Edit Trips

4.2.1 Creating Trips

Creating Trips - with an account, users are able to add trips to their dashboard. When creating a trip, the application prompts the user to input information about the members of the trip, the GPS devices the user would like to track for this trip. The application will allow users to specify markers and geofences for the trip on a map.

Users who create trips are defaulted to have "trip manager" privileges within that trip. Trip managers then can change permission levels for other users who are invited and accept invitations to the trip. The base level of viewer permissions for a trip allows users to view location data and members on the trip. They are not given permission to edit any trip information without first being given permission by the trip creator or another trip member who has been given manager permissions.

4.2.2 Editing Trips

After the initial creation of a trip, members of a trip with the correct permissions (either the creator of the trip, or another trip member who has been given editing permissions by the trip creator), can edit or add information to a trip using the toolbar on the page for the specific trip they are trying to edit. This trip page will be accessible from the user's dashboard page. Depending on their permissions, they would be able to see some or all the following options on the toolbar: "manage participants", "add a device", "set route + geo-fence", "set notifications".

4.2.3 Viewing Trips and Sending Invites

If a user with an account is sent an invitation to view a trip, they will be given permissions to view the basic data for that trip. This data includes GPS location data, members associated with the trip, and the planned route and geo-fence. Viewers of the trip do not necessarily have to be listed as members on the trip. The trip creator and trip members has the capability to send invites to anyone with an account to view the trip progress. Additionally, those with the correct permissions can also quickly add anyone on their friends list.

4.3 Set Routes, Geo-fencing, and Notifications

4.3.1 Setting Geofences

Users will have the ability to set geofences by specifying a point on the map. The geofence can then be defined from this point, either by a default radius from that point, or by a user specified radius from their point. The fence formed by this point will then be generated on the map.

4.3.2 Notifications

If a GPS device registered with a specific trip travel outside the bounds of the geo-fence for a predetermined amount of time, users have the option to specify how notifications are sent to some users. Users can specify who gets notifications and under what time and location circumstances they would get them. This can be changed and set within the trip viewer page by users with the correct permissions. Also, when sending a friend request a notification will be sent to both users notification center one of which says sending the other which has the option of accept or deny.

4.4 Wish List Features

- **Trip Dashboard Search -** Users will gain the ability to search for trips by name of the trip.
- **Search for Public Trips -** Users will have the option of making a trip public to their friends. This will give friends the opportunity to possibly join their friends trips without being explicitly invited.
- **Navbar Search Bar -** Will give users the ability to search for trips or locations
- **Add Device Instructions** - Web pages which will walk the user through adding various devcies.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

5.1.1 Creating Accounts

Server will dynamically allocate resources depending on throughput of data from GPS devices in use.

5.1.2 User Device Specification

At minimum a device would require:

- Processor of 1.9GHz

- Memory of 2Gb
- Bandwidth of 400kbps
- Latency under 150ms

5.2 Safety Requirements

In the event of a catastrophic failure in the database, a timed backup will be restored, and operation will be reapplied to achieve the closest to the previous current state.

5.3 Security Requirements

5.3.1 Lock Requirement

Server will dynamically allocate resources depending on throughput of data from GPS devices in use.

5.3.2 Access to Data

- Logging Access to Data
  - When location or personal data is viewed, the viewing of the data will be logged.
- Restricting Access to Data
  - When location or personal data is called to be viewed, the viewing of the data will only be given to allowed users. Access to certain data is restricted to other users.
- This functionality of restricting data protects users from their data being viewed by anyone.

5.3.4 Device Ownership and Viewability

A device owner has all rights to pair, view, and unpair devices. When a device is loaned to a different person, that owner still has all rights to view and limit rights. For security purposes, this allows random users to not be able to claim ownership over an already claimed device.

5.4 Software Quality Attributes

5.4.1 Maintainability

When new updates, bug fixes, or added features are made after development, the database will still maintain all existing tables and data and will not lose this data. This means that the database will be designed, created, and tested appropriately.
In the implementation of various methods and functions, no method or function should exceed 500 lines for understanding and bug fixing.

5.4.2 Availability

System should be up 24/7 unless a scheduled downtime is announced.

5.4.3 Reliability

Database notifications such as insertions, updates, and deletions should not fail at any point without being logged. With this database backups should be performed regularly to prevent data loss.

5.4.4 Performance

- App and Web Page Loading Time
  - When location or personal data is viewed, the viewing of the data will be logged in temporary access logs.
- Restricting Access to Data
  - The webpage should be optimized to load the full webpage within 10 seconds given the normal broadband high-speed internet of 25 Mbps.
- User Capacity
  - The server should be able to scale for user capacity depending on the demand of the server.

Design Specifications

6. Design Overview

6.1 Description of the Problem

This section of the document discusses the design specifications for the project and how it was implemented during the Spring and Fall semesters. The goal of the document is to give a clear description on how the application was designed and implemented while also discussing the reasons for the design choices. Many factors were kept in mind when designing the architecture and functionality of the project's design. These factors were:

6.1.1 Ease of Implementation

Our goal for this project was to choose frameworks that would facilitate ease of implementation. React.js and Django have very powerful features that allow for creation without having to work from scratch.

6.1.2 Security

Like any other well built application, the Walks web application has been implemented in a way which will not compromise users data privacy. Since the Walks application will be tracking users locations via GPS devices, it is extremely important that user's information be secured.

6.1.3 User functionality

Another major factor in the design of the application is what layouts will allow users to easily navigate throughout the application. Additionally, the types of features needed to give users a powerful and useful tool to track and manage hiking trips were considered. The application is a multi-page application which is built through implementing custom made components.

6.1.4 Performance

The application must be able to perform its needed tasks to ensure the user has a smooth experience. How the data is managed, stored, and transferred was heavily influenced by the performance factor.

6.1.5 Scalability

Although the application is not being built for real world usage, many of the technologies and architecture choices enable the application to scale for heavier usage. The employment of an AWS EC2 virtual server allows for load balancing and scheduling to handle high amounts of traffic.

6.2 Technologies to be used

6.2.1 Development Tools

- **Git -** enables collaboration among team members while keeping development separated using version control. Git will keep track of all additions, updates, and deletions.
- **Docker -** Containers are a standardized unit of software that allows developers to isolate their app from its environment, solving the "it works on my machine" headache.
- **VS Code -** enables team members to utilize Git and Docker within one IDE. This IDE is well documented and capable of bringing in various other extensions.

6.2.2 Frameworks

- **React.js -** is a front-end JavaScript library for building user interfaces or UI components. React.js will be utilized to create a dynamic website and will be responsible for making API calls and HTTP requests to the server.
- **Django -** is a python framework for full stack web development. This will primarily be

used as the back-end framework to create API’s which will interact with the front-end framework.

6.2.3 Languages

- **HTML/CSS -** HTML is the most basic building block of the Web. It defines the meaning and structure of web content. CSS is used to describe a web page's appearance and presentation. Both have been utilized to implement the UI/UX of the web application.
- **Python -** is an interpreted high-level general-purpose programming language. It was used mainly for the back-end development by employing the Django framework.
- **JavaScript -** is a high-level, often just-in-time compiled, and multi-paradigm programming language. It was mainly used for the front-end development by employing the React.js framework.
- **Bash** - is a scripting language which is highly utilized in many systems. Within the project bash is being utilized to automate the deployment of the application.
- **Swift** - is a high level programming language made for Apple devices specifically iOS. We will be using Swift to write the functionality of our Mobile Application.

6.2.4 External API’s

- **Google Maps -** integrates Google's Place details, search, and converts coordinates into addresses into an app. This has been used in the front end UI/UX of the web application.
- **Visual Crossing Weather -** integrates Weather data of trips displayed on the trip manager pages. This data uses the coordinates of the trip to display its current weather as a viewable display.
- **Geoforce -** allows for the retrieval of GPS data of all GeoForce Devices. This will be needed to properly integrate the location of the assets or participants in a trip using this type of device.
- **SPOT -** allows for the retrieval of GPS data of all Spot GPS Devices. This will be needed to properly integrate the location of the assets or participants in a trip using this type of device.

6.2.5 Database

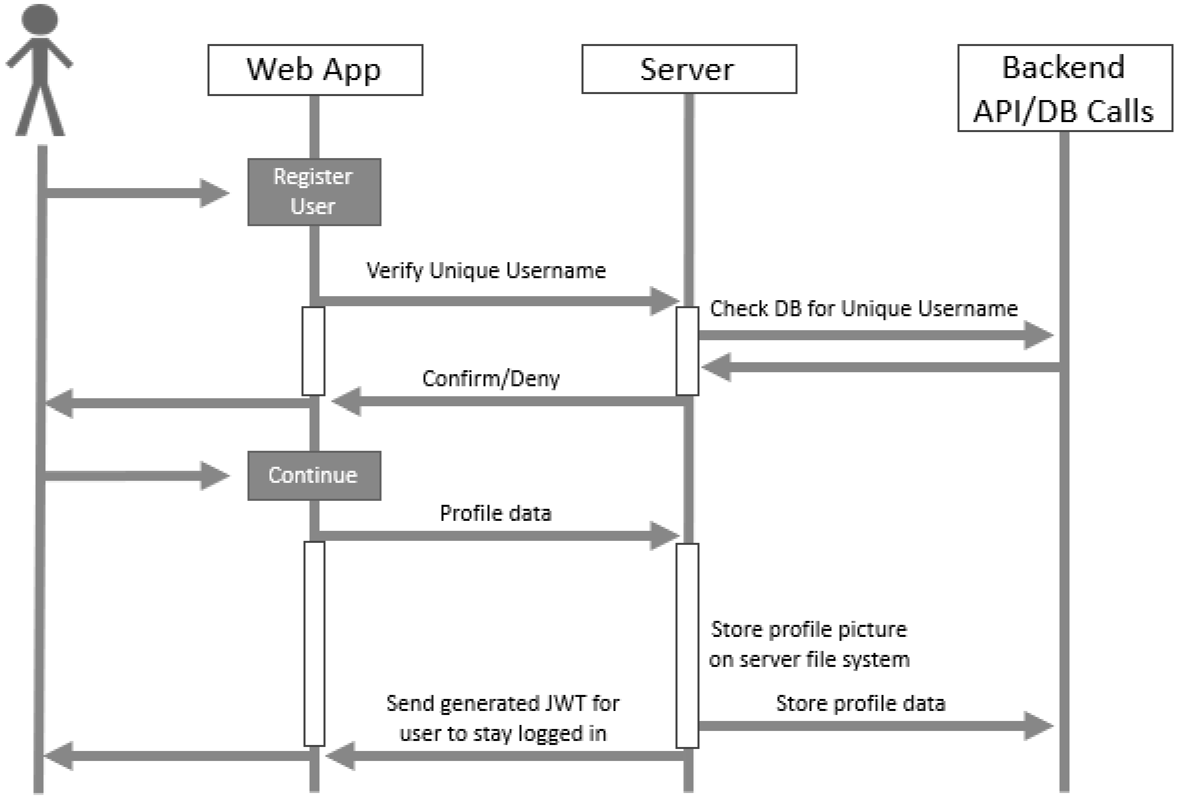
- **SQLite -** is a relational database management system. SQLite is not a client/server database engine, but rather used in end programs

7. UML Design Diagrams

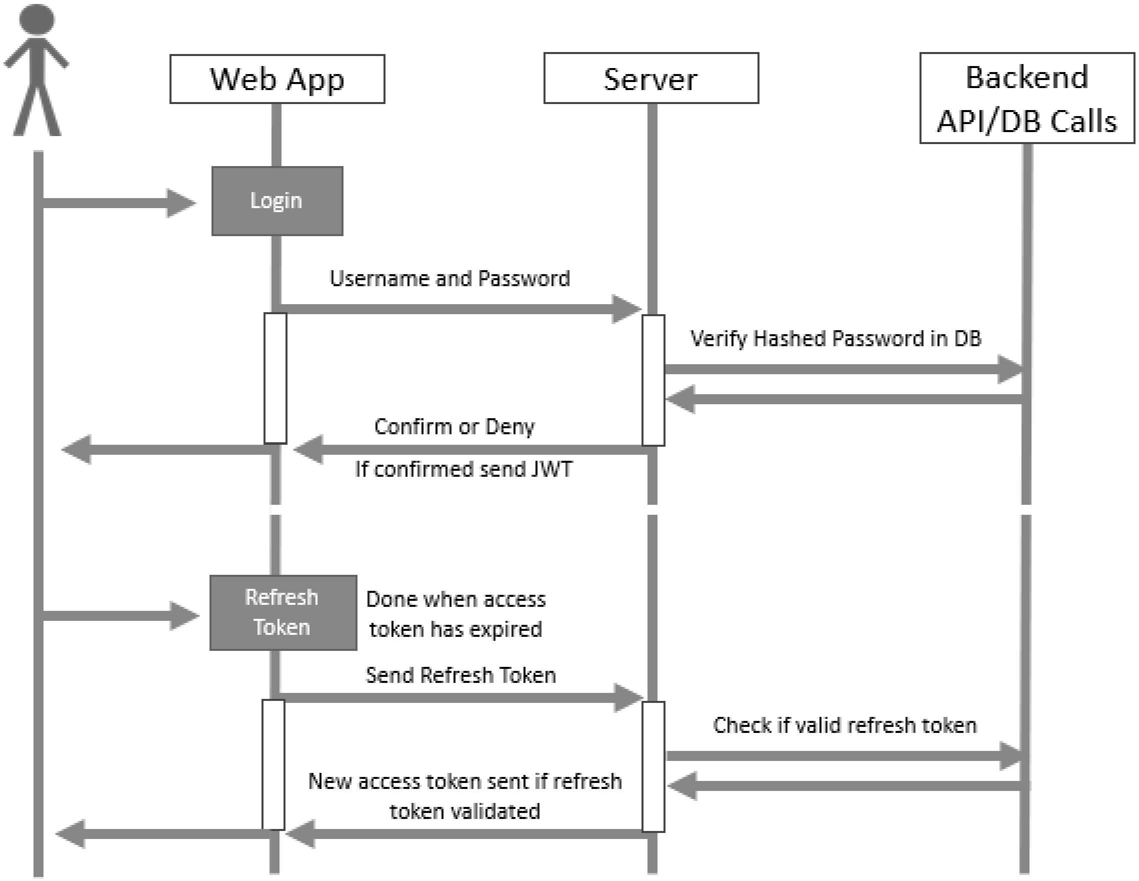
7.1 Sequence Diagrams

The following diagrams describe specific interactions between the user, web application, server, and database/API calls. The goal of these diagrams is to give a visual representation of how the data will be flowing to and from the various parts of the application architecture. These diagrams will help in implementing many of the backend API endpoints as well as the frontend API calls.

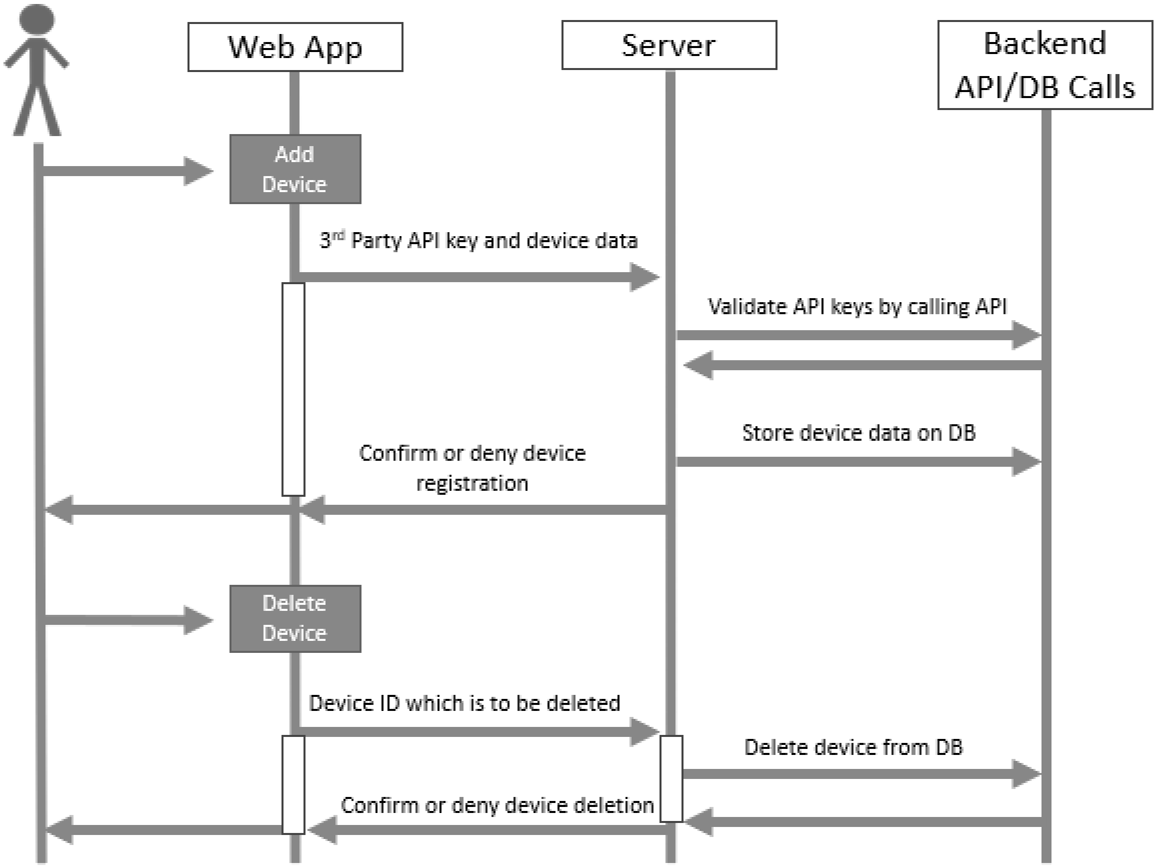
7.1.1 User Registration



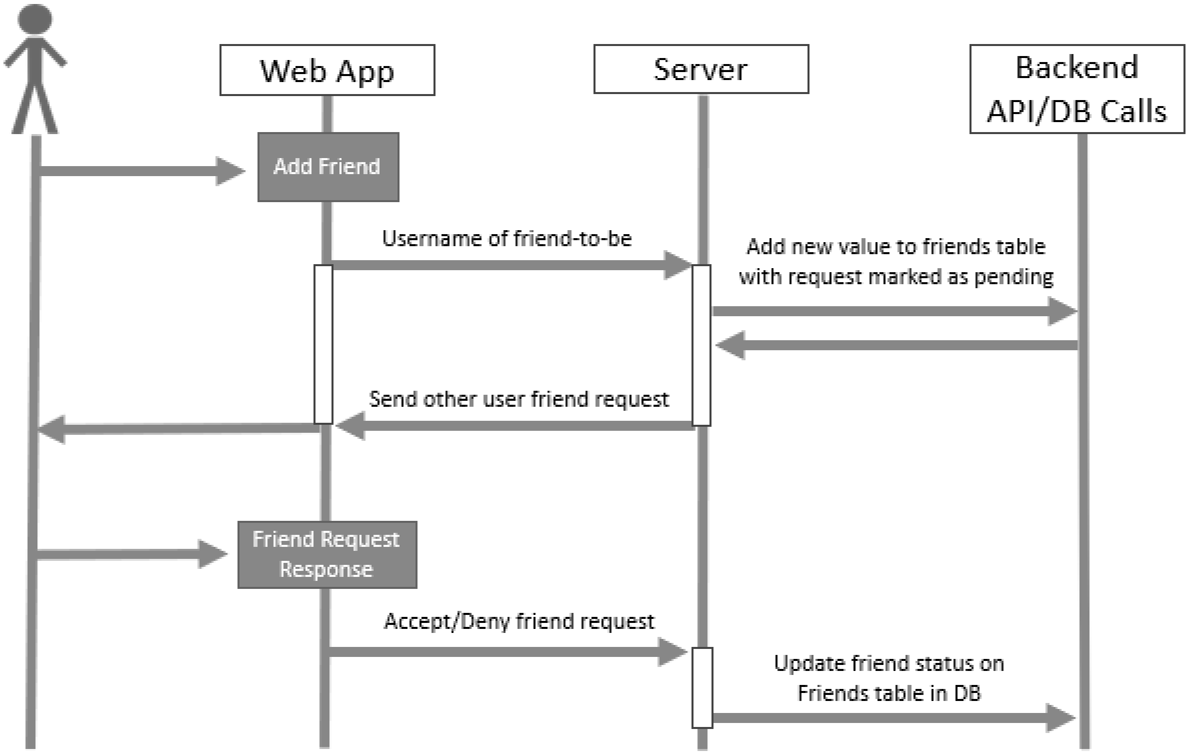
7.1.2 User Login



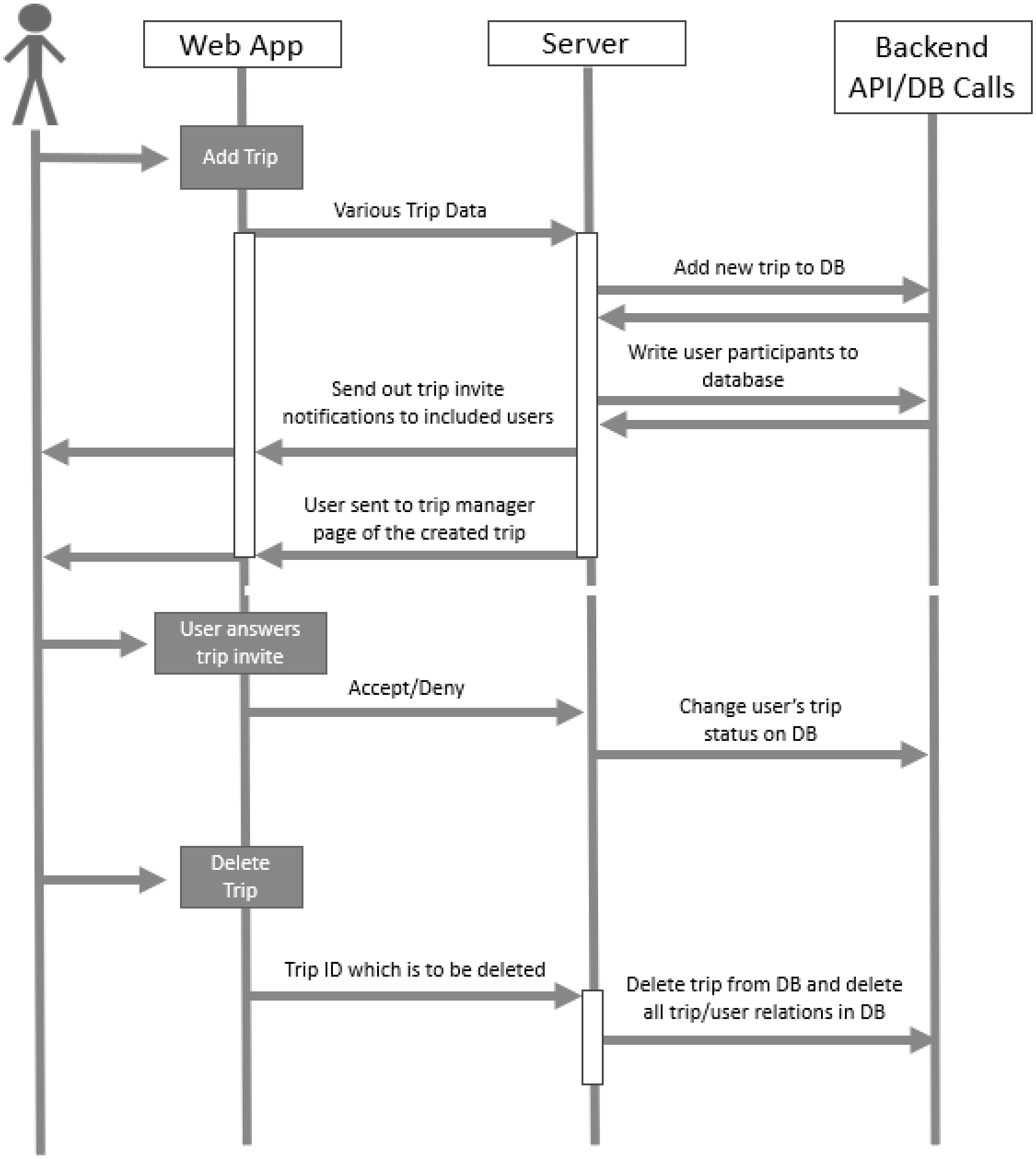
7.1.3 Add/Delete GPS Device


Web App
Server
Backend API/DB Calls
Add Device
3rd Party API key and device data
Validate API keys by calling API
Store device data on DB
Confirm or deny device registration
Delete Device
Device ID which is to be deleted
Delete device from DB
Confirm or deny device deletion

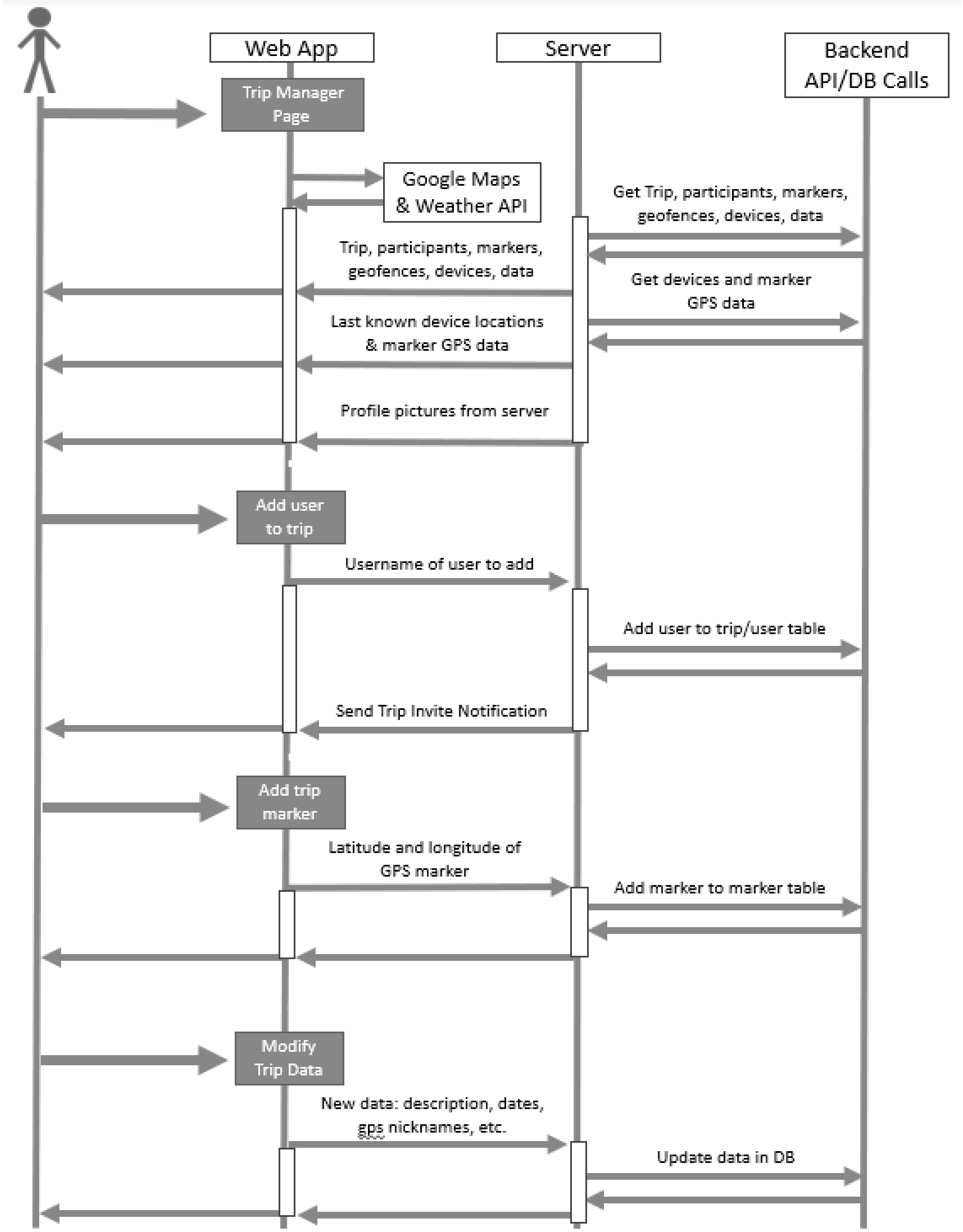
7.1.4 Add Friend


Web App
Server
Backend API/DB Calls
Add Friend
Username of friend-to-be
Add new value to friends table with request marked as pending
Send other user friend request
Friend Request Response
Accept/Deny friend request
Update friend status on Friends table in DB

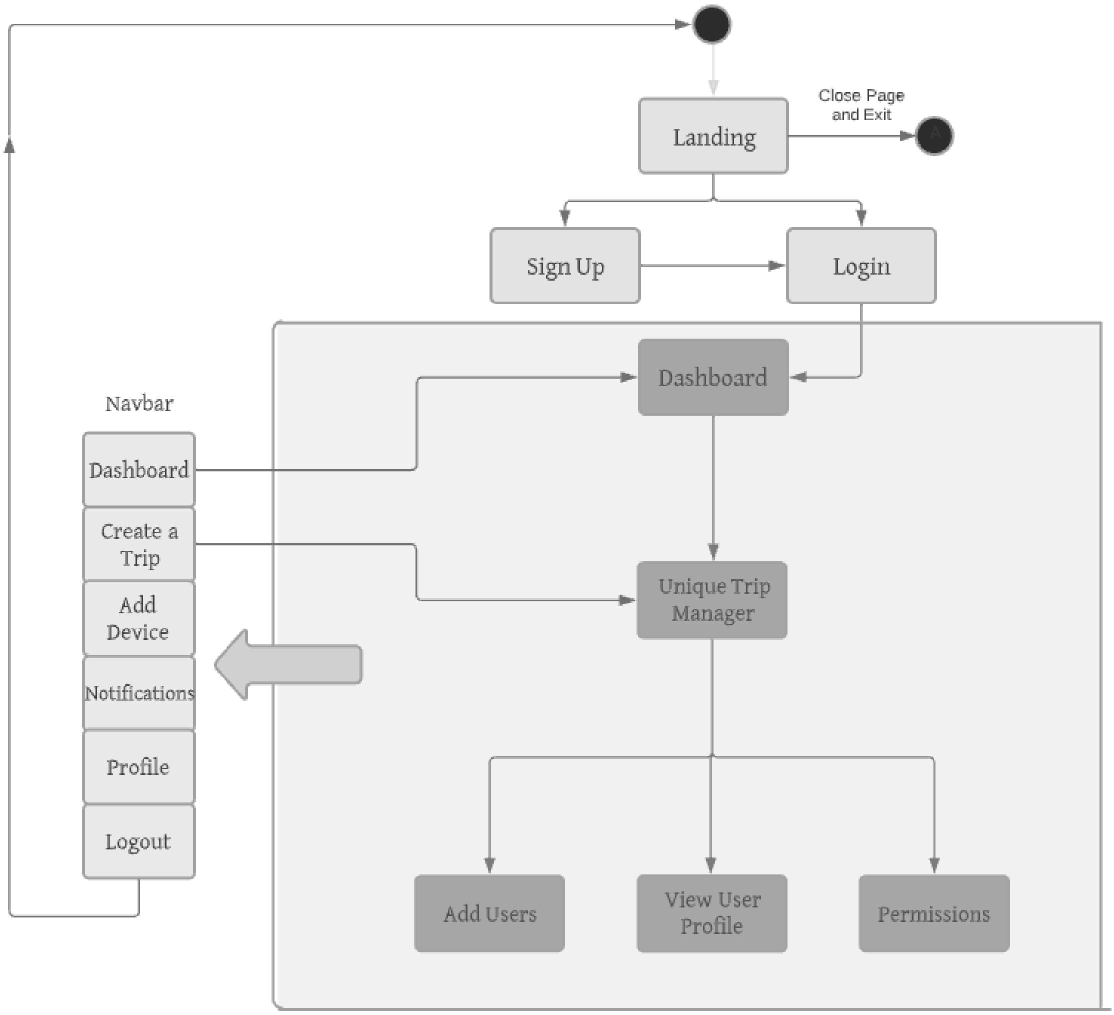
7.1.5 Add/Delete Trip


Web App
Server
Backend API/DB Calls
Add Trip
Various Trip Data
Add new trip to DB
Write user participants to database
Send out trip invite notifications to included users
User sent to trip manager page of the created trip
User answers trip invite
Accept/Deny
Change user's trip status on DB
Delete Trip
Trip ID which is to be deleted
Delete trip from DB and delete all trip/user relations in DB

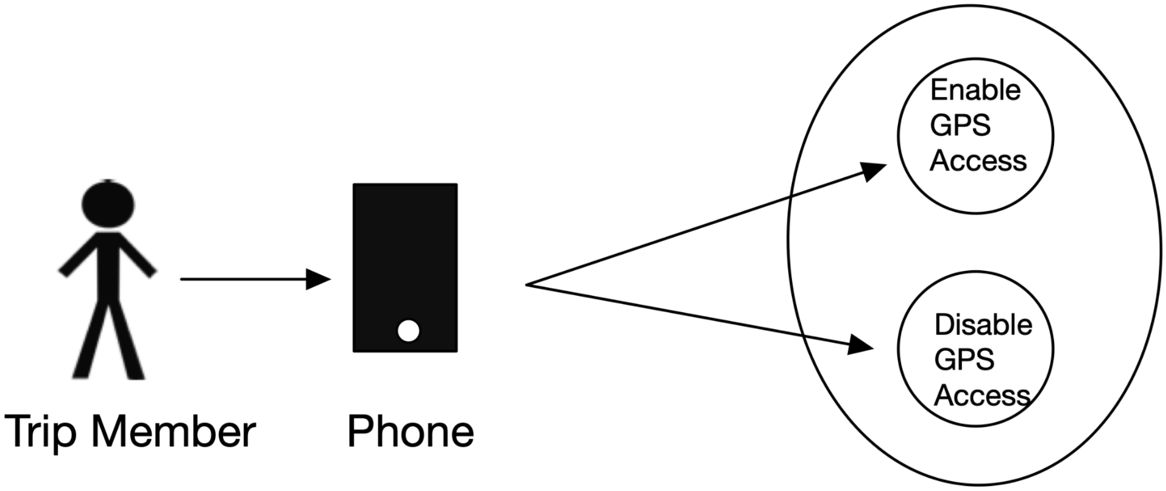
7.1.7 Trip Manager Page



7.2 Page Flow


Close Page and Exit
Landing
Sign Up
Login
Navbar
Dashboard
Create a Trip
Add Device
Notifications
Profile
Logout
Dashboard
Unique Trip Manager
Add Users
View User Profile
Permissions

7.3 Use Case Diagram

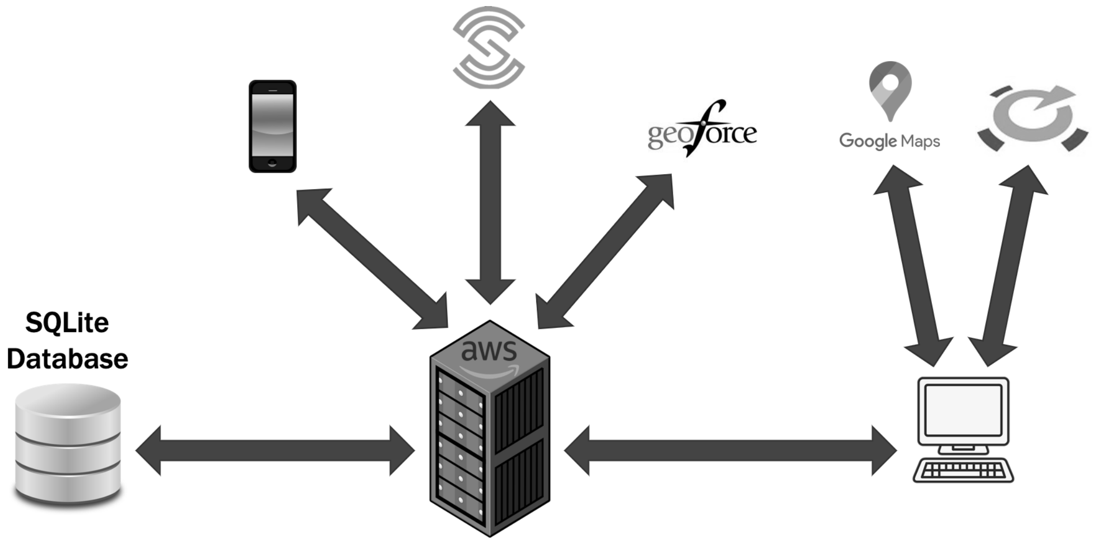


8. Software Architecture

8.1 Architecture Diagram

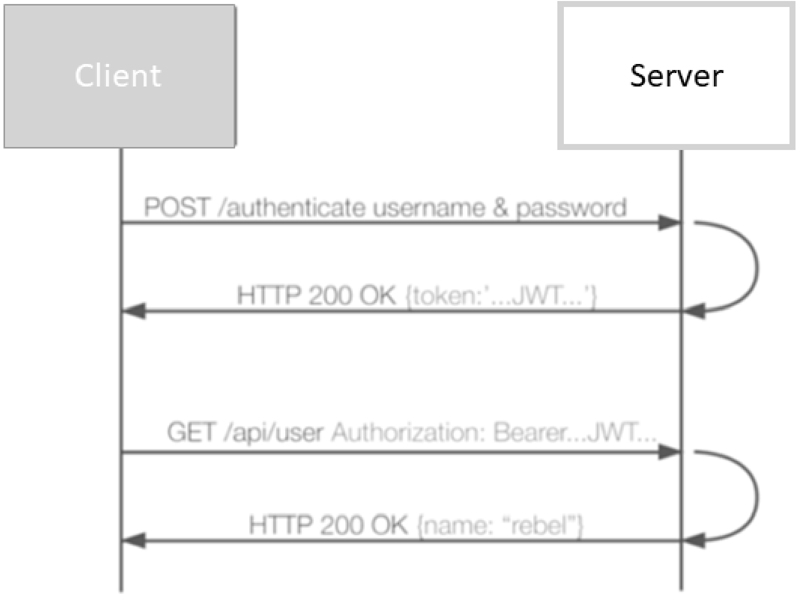
8.1.1 Server/Client Architecture

Below is a diagram which visually describes the overall architecture of the Walks application. The main components include: database, server, client, and 3rd party APIs. These third party APIs will be accessed by credentials provided by the user which are available through their subscription or setup to the GPS device provider. In the diagram below the database is separate from the server, but in actuality the database can be hosted on the same server. The server handles a large portion of the business logic and will also act as a proxy between the client and the 3rd party API calls to process the data securely. The client is responsible for sending, retrieving, and displaying data for the user. The client side will make Google Maps and Visual Crossing Weather API calls to obtain map and weather data with the user.



8.1.2 JWT User Identification

JSON Web Token (JWT) is an open standard that defines a compact and self-contained way for securely transmitting information between parties as a JSON object. This information can be verified and trusted because it is digitally signed. JWT's will be utilized to verify requests from clients are not corrupted and also ensures the user has access to the resources being requested. The JWT is generated upon sign up or login and allows the backend to identify the user making requests.


Client
Server
POST /authenticate username & password
HTTP 200 OK {token:'...JWT...'}
GET /api/user Authorization: Bearer...JWT...
HTTP 200 OK {name: "rebel"}

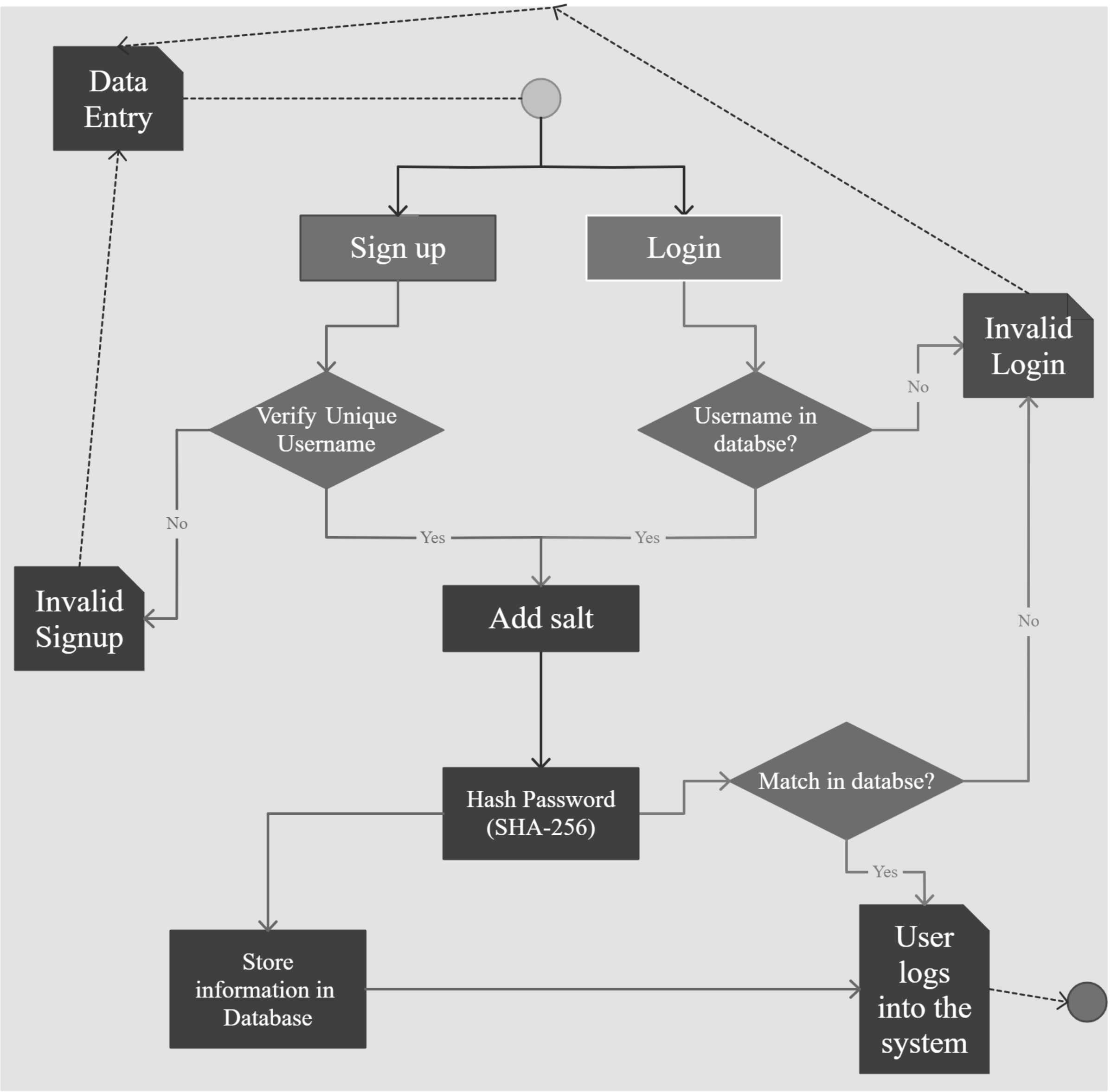
8.1.3 Picture Storage

It is typically best practice to store picture data within a server file system rather than directly on a database. Storing picture data on a database makes reading and writing this kind of data difficult and slow. Thus, to store pictures for the application the picture files will be saved in a location on the server file system. Then, within the database tables, file paths to these pictures will be stored. When a user's profile picture is needed the server will query the database for the file path location of the folder and then the server will utilize the file path to extract the picture and send it to the client.

8.2 Architectural Alternatives

We could separate the database from the web server to make its own server so that it is more protected in the case of a breach on the web server, but for our case we will not be handling the information that would make a separate server necessary. Additionally, combining the two makes for ease of implementation and ease on the budget for this project. Also, the user could be making all the API calls to the likes of Geoforce, but that would be a lot of calls to several targets so it's easier on the user end to simply communicate with our AWS server.

8.3 Activity Diagram Login Model

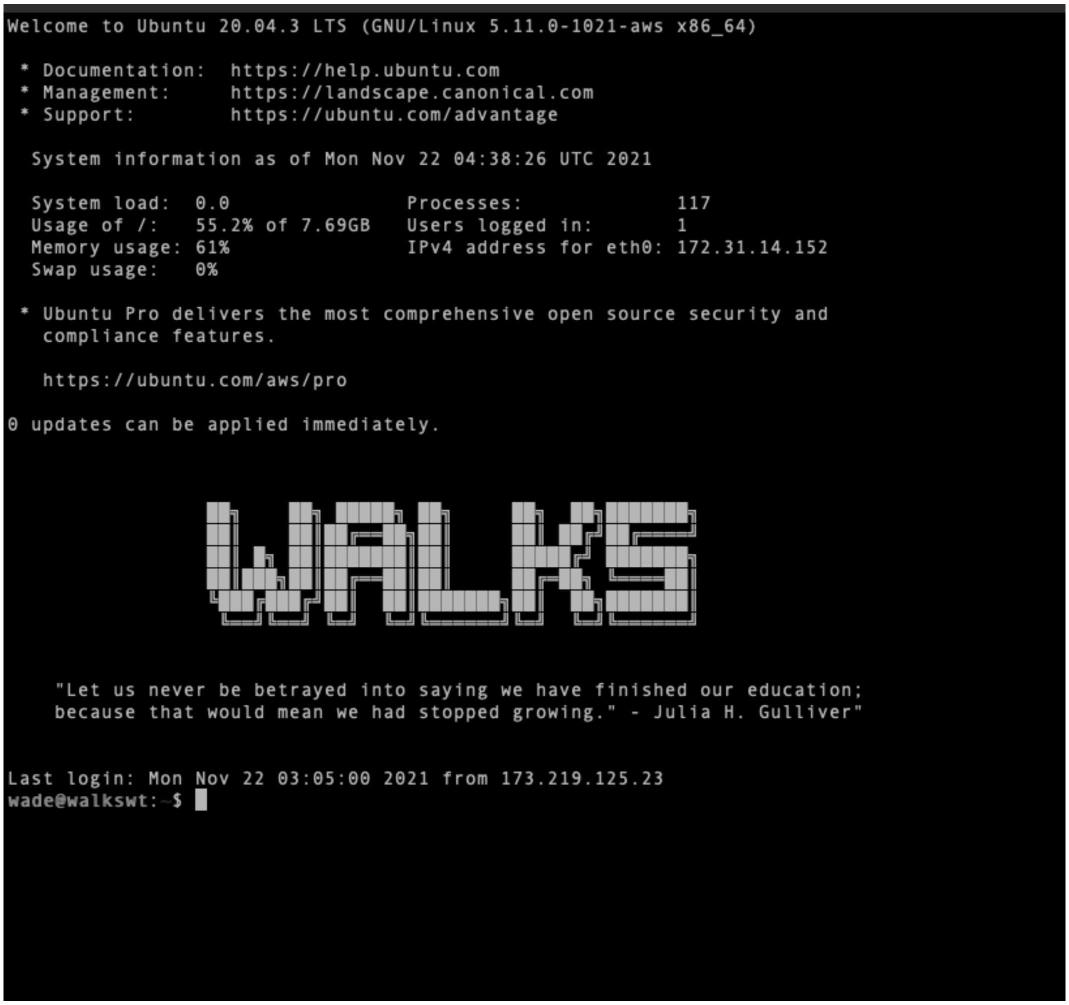


8.4 Design Rationale

In order to prevent the user from ever needing direct access to the database, anytime information needs to be sent/received, or tracking information needs to be gathered, an AWS server will handle the transfer of information. This means the user only needs to communicate with the server,Google Maps and Visual Crossing Weather API's.

8.5 AWS Server Customization

With our AWS Server we are running a Linux x86_64 Server running Ubuntu 20.04 which is the same version of Ubuntu that our Docker containers are running for development. With this server created it is customized completely from the ground up giving us complete access and control over all communication. Below is an image that shows the customized ssh message when accessing the server. It should be noted this is only done by being a system admin within a Ubuntu server. This server should be updated and maintained regularly to prevent any security vulnerabilities.



```
Welcome to Ubuntu 20.04.3 LTS (GNU/Linux 5.11.0-1021-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  System information as of Mon Nov 22 04:38:26 UTC 2021

  System load:  0.0               Processes:             117
  Usage of /:   55.2% of 7.69GB   Users logged in:       1
  Memory usage: 61%               IPv4 address for eth0: 172.31.14.152
  Swap usage:   0%

 * Ubuntu Pro delivers the most comprehensive open source security and
   compliance features.

   https://ubuntu.com/aws/pro

0 updates can be applied immediately.

```

```
    "Let us never be betrayed into saying we have finished our education;
    because that would mean we had stopped growing." - Julia H. Gulliver"


Last login: Mon Nov 22 03:05:00 2021 from 173.219.125.23
wade@walkswt:~$ 
```

9. User Interface Design

For a software development team, the user interface of their application serves as the bridge between the application and users of the application. For the user, the interface they interact with *is* the application. Users are not concerned with the complex interactions between data tables, or servers, they want the application to work easily, and to meet their specific needs. As such, the Walks application seeks to provide our diverse group of users with straightforward and intuitive solutions to each of their varied and distinct needs and desires when they use the Walks application. This rests primarily on two factors: firstly, that the frameworks in place serve the front end consistently, and that the artistic elements of the design remain simple and contribute to the intuitive nature of the application functionality.

The frontend of the website was built using the React.js framework. The application has 10 pages:

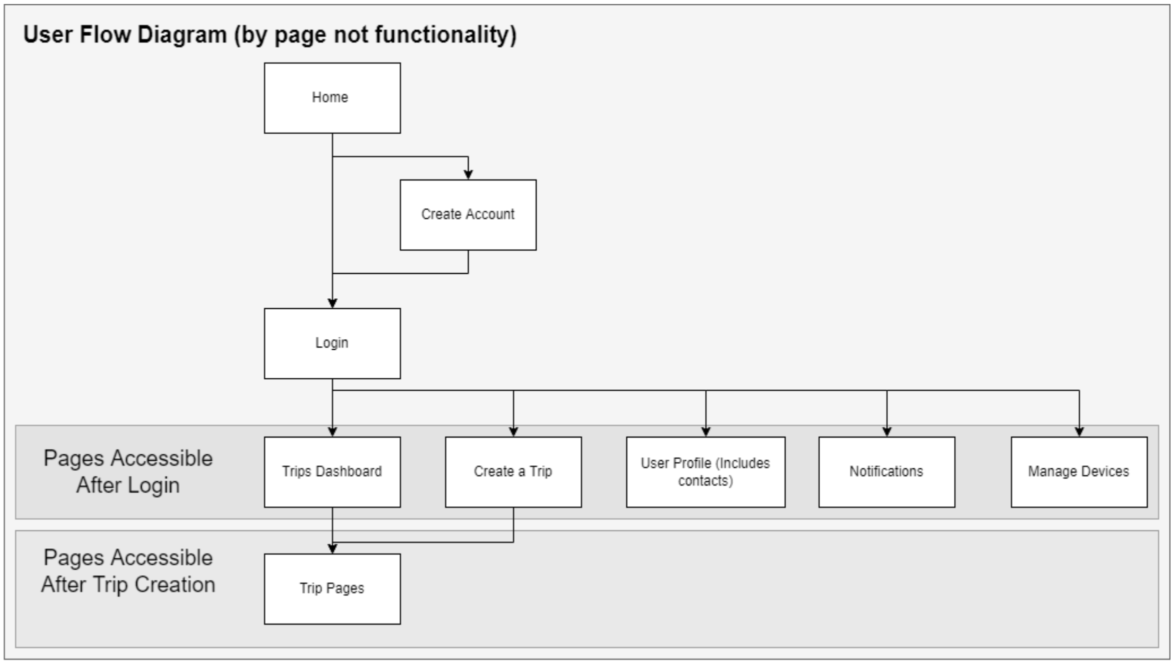
- Landing Page
- Sign Up Page
- Login
- Trip Dashboard
- User Profile
- Individual Trip Manager pages
- Create Trip page
- Add Device
- Public Trips Page
- Notification page

Each page was created as a JavaScript function, instead of using classes. Throughout our website we used some components/templates from material.ui. Material.ui templates were used sparingly and only in less critical webpages. The Login page for example was a materials.ui template. Using this template allowed us to focus more on developing the heart of our application without being caught in the mire of building out each necessary component.

One important UI component that was built out is the Navbar (NAVigation BAR). This allows a user to navigate to different pages of the application easily. The Navbar was also built as a javascript function, allowing for easy implementation on each page added to the application, and continuity across the application as a whole.

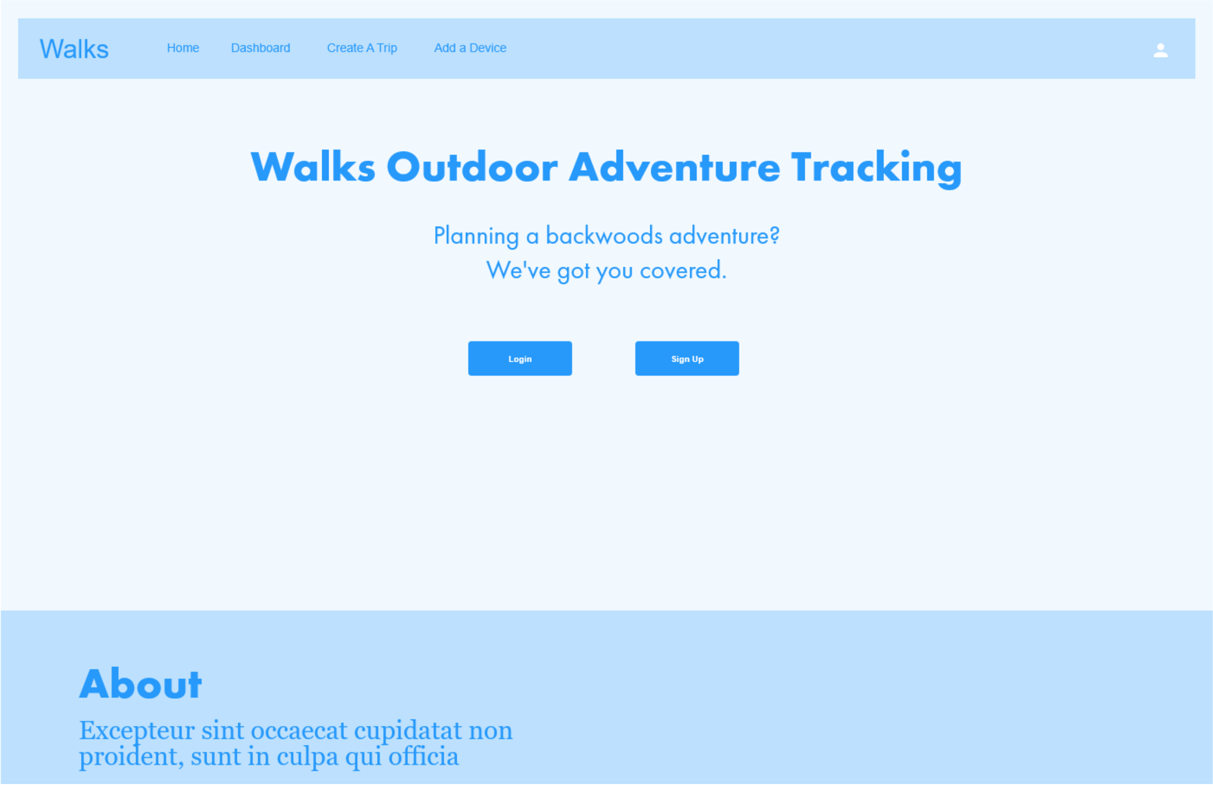
Another feature that has promoted application continuity is the use of corsheaders. Because React.js is not nested inside of the Django directory, it is an outside resource that needs access to django's data. Because the front and back ends are completely separate, there exists a need for a resource to share data across both halves of the application. The Cors (Cross Origin Resource Sharer) header is therefore crucial to ensuring Walks application functionality.

The following User Flow Diagram details the primary flow of pages within the Walks application. The diagram differentiates between pages that can be accessed publicly, and which pages are only accessible after account creation and login. It is important to note that once a user has successfully logged in, any of the "login accessible" pages other than the individual trip pages can be accessed through using the navbar.


User Flow Diagram (by page not functionality)
Home
Create Account
Login
Pages Accessible After Login
Trips Dashboard
Create a Trip
User Profile (Includes contacts)
Notifications
Manage Devices
Pages Accessible After Trip Creation
Trip Pages

9.1 Landing Page for web application

Landing page wireframe:



Walks Outdoor Adventure Tracking

Planning a backwoods adventure?
We've got you covered.

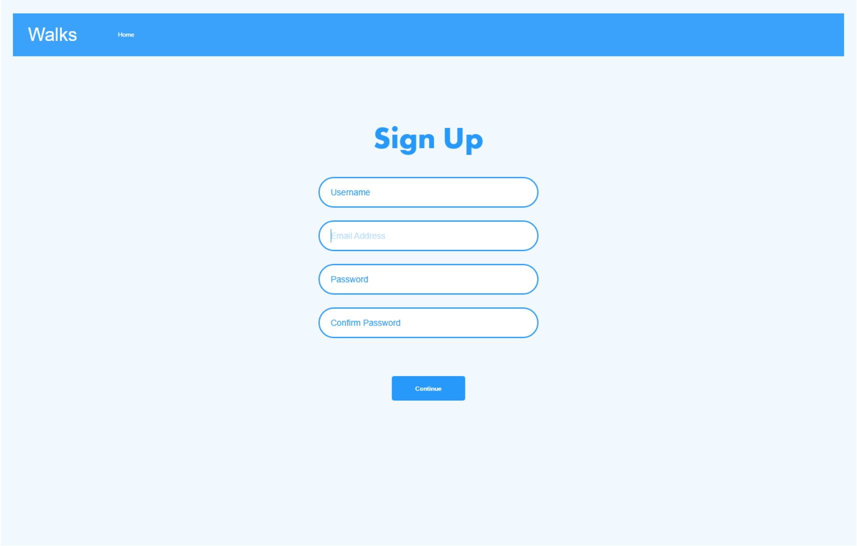
About

Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia

9.2 Sign Up

9.2.1 User account information

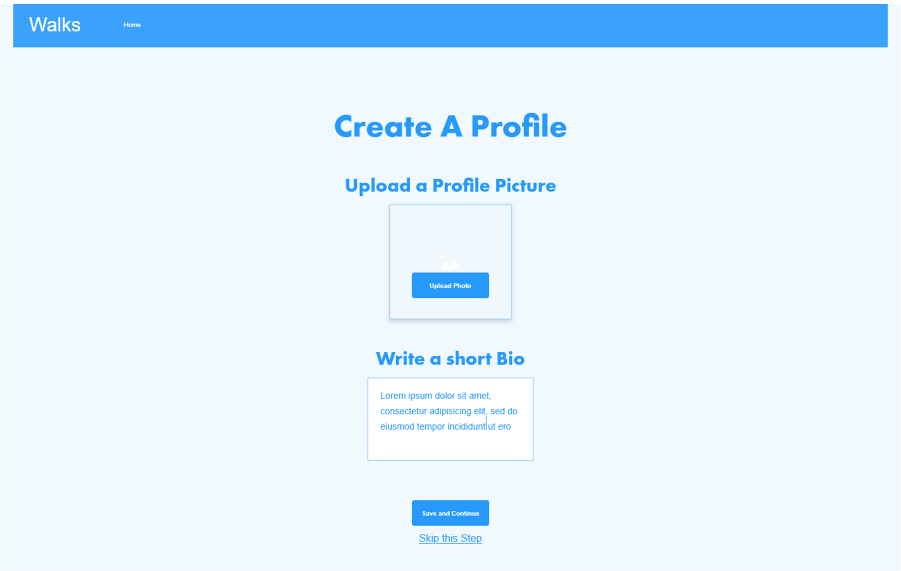
User account creation wireframe:



9.2.2 Profile Bio

Add public profile information (profile picture and short bio) Sharing this information is optional for account creation.

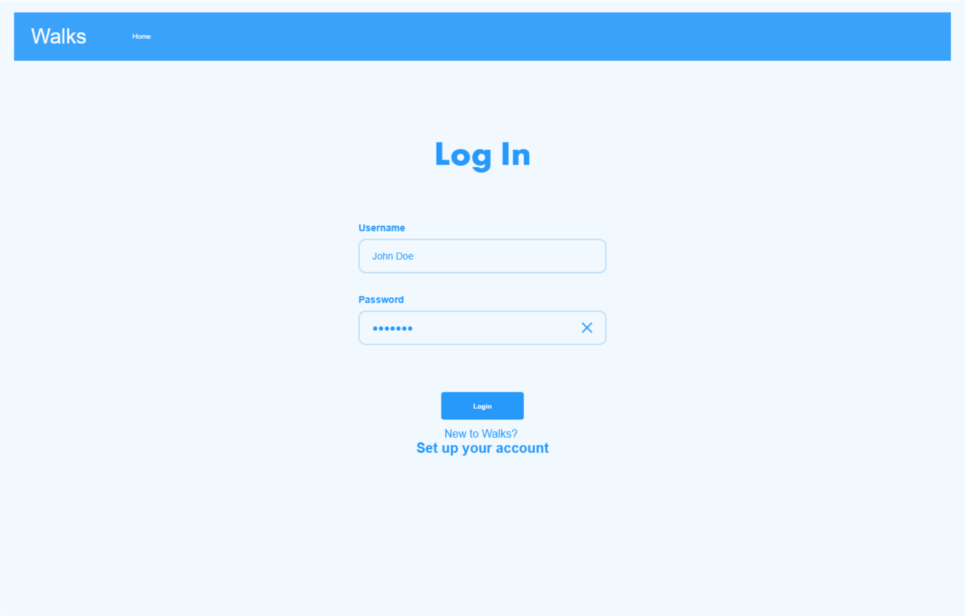
Wireframe for profile creation form:



Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut ero

9.2.3 User Login Page

Wireframe for user login form:



Username

John Doe

Password

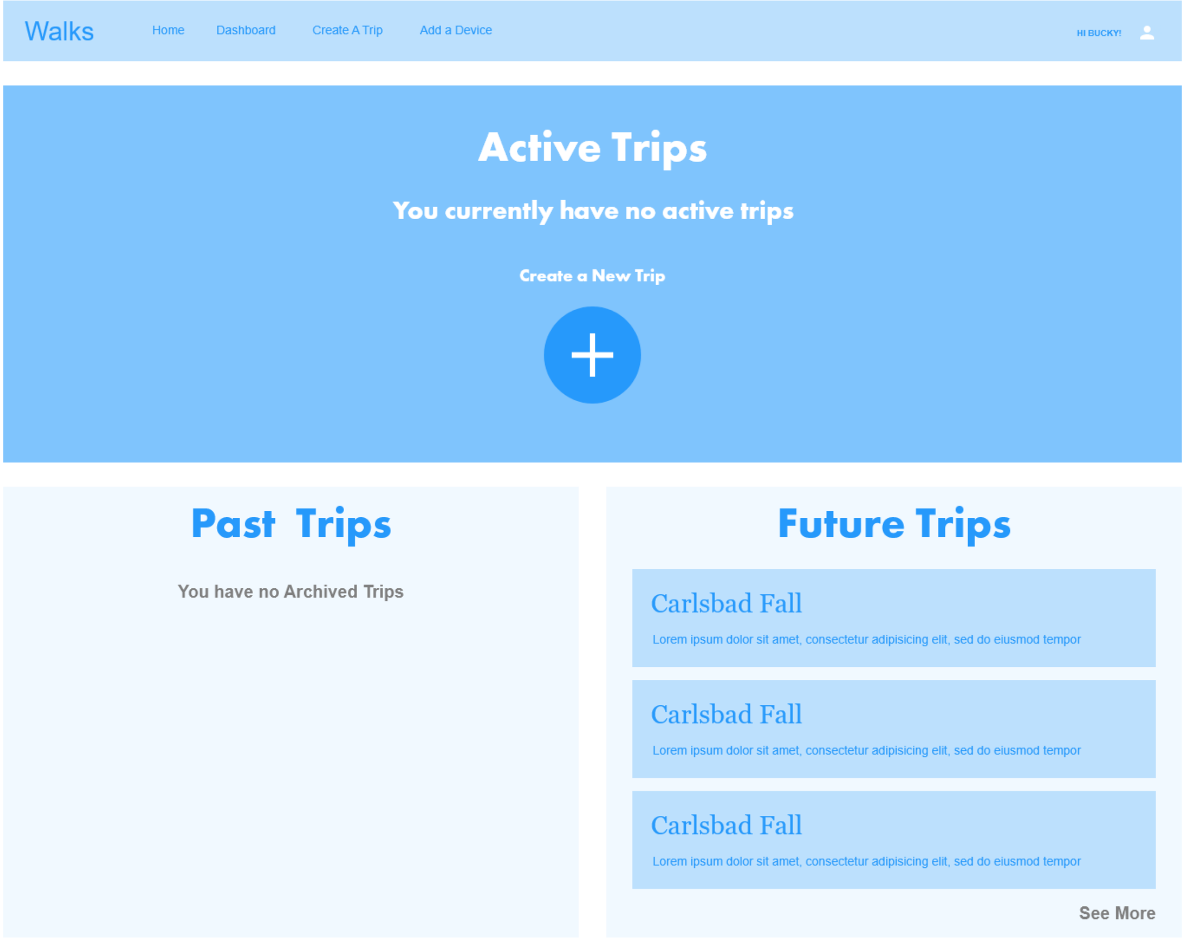
•••••••

9.3 General layout for user dashboard.

9.3.1 E.g. 1

Wireframe for user dashboard (1):

This mock up assumes the user has created or been invited and accepted invitations to trips scheduled for a future date. They have no "active trips", meaning they are not a member or spectator for a trip where the day they are viewing their dashboard is within the date range specified for the trip.



Active Trips

You currently have no active trips

Past Trips

You have no Archived Trips

Future Trips

Carlsbad Fall

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor

Carlsbad Fall

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor

Carlsbad Fall

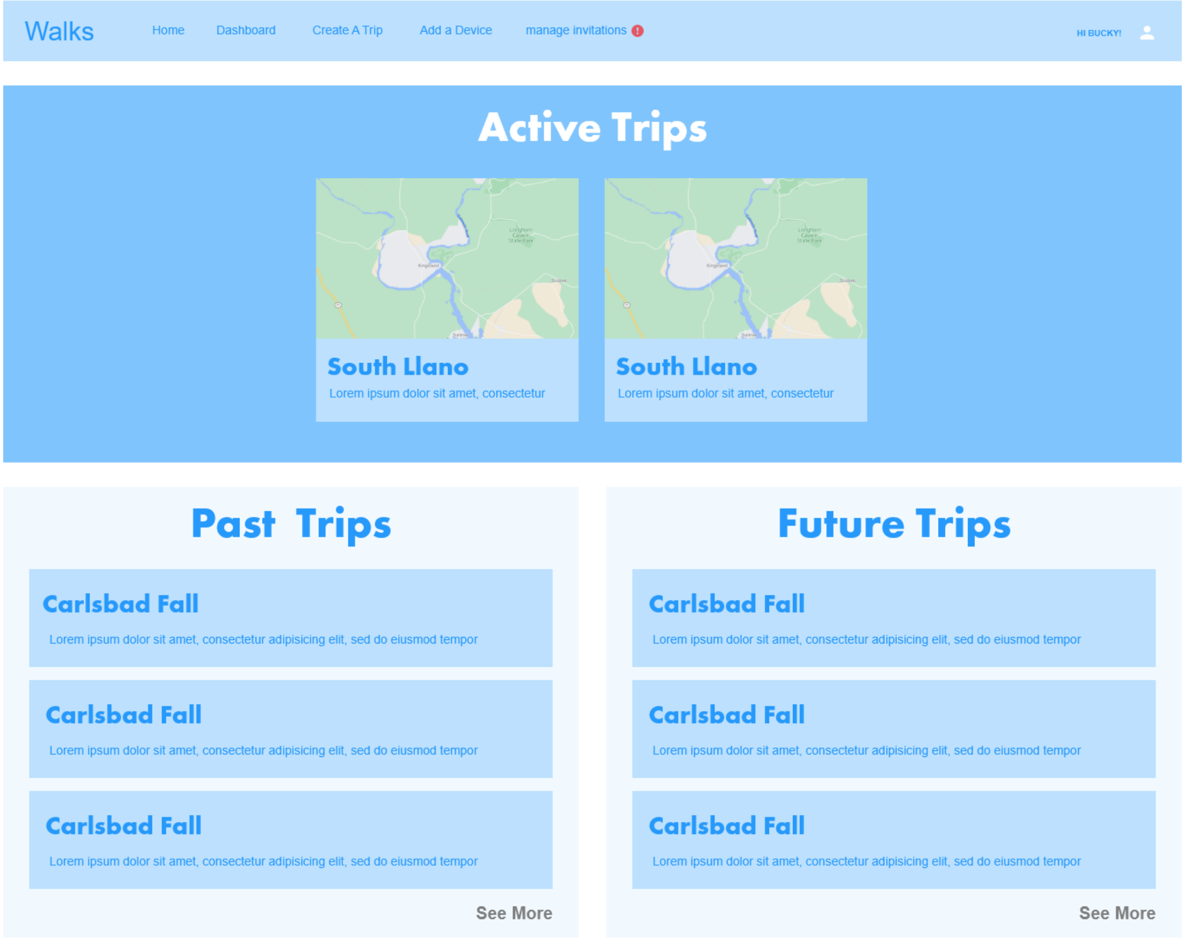
Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor

9.3.2 E.g. 2

Wireframe for user dashboard (2):

This mock up assumes the user has created or been invited to past trips, and future trips. They have also created or invited two trips that are currently active.

This example also showcases a notification alert for incoming trip invitations seen in the toolbar at the top.



Active Trips

South Llano

Lorem ipsum dolor sit amet, consectetur

South Llano

Lorem ipsum dolor sit amet, consectetur

Past Trips

Carlsbad Fall

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor

Carlsbad Fall

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor

Carlsbad Fall

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor

Future Trips

Carlsbad Fall

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor

Carlsbad Fall

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor

Carlsbad Fall

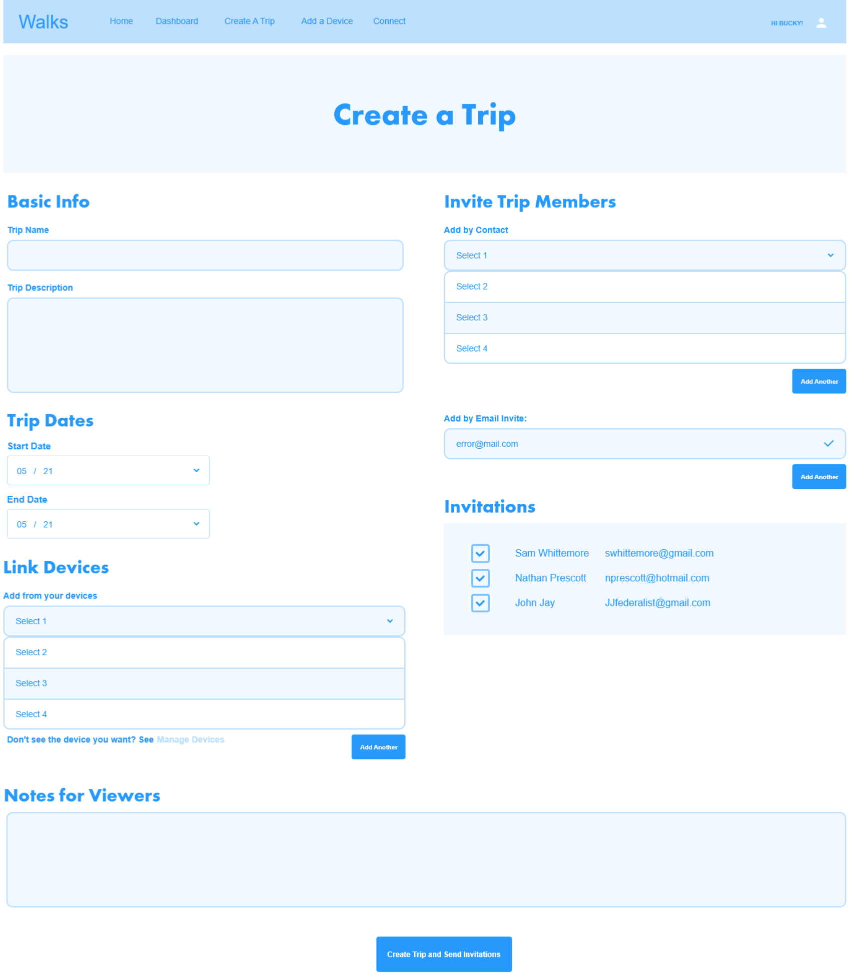
Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor

9.4 Create a trip form

Allows users to:

- Add details such as trip name, trip description and notes to display to trip viewers and participants.
- Link devices the trip creator has associated with their account
- Send trip invitations to others by contact or by email (this will ask the email recipient to register with a Walks account)

Wireframe for create trip form:

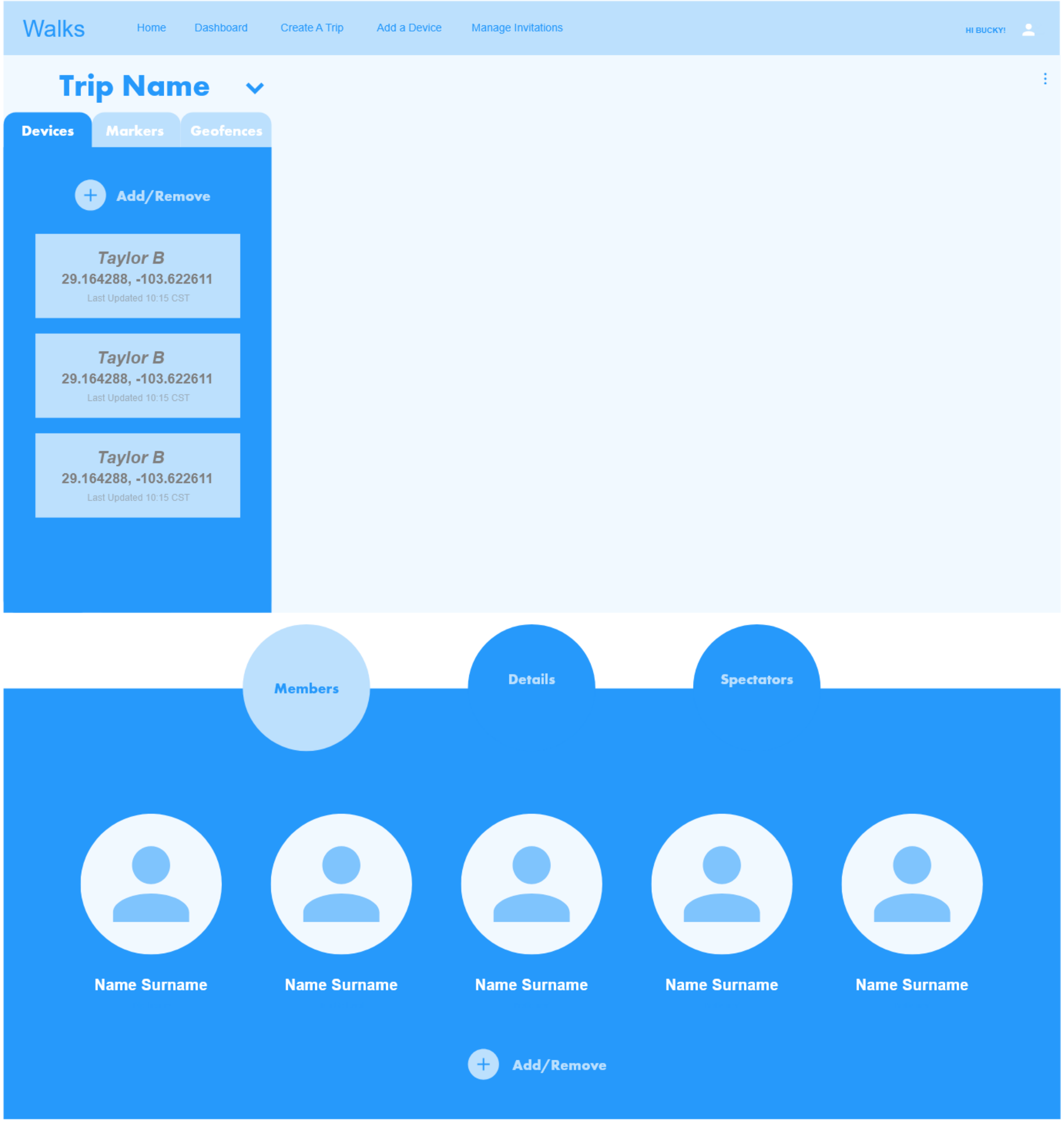


☑	Sam Whittemore	swhittemore@gmail.com
☑	Nathan Prescott	nprescott@hotmail.com
☑	John Jay	JJfederalist@gmail.com

9.5 Individual Trip Page

Displays map data from API, devices linked with trip, trip markers, trip geofences, description, notes from trip managers, members of trip, and trip spectators.

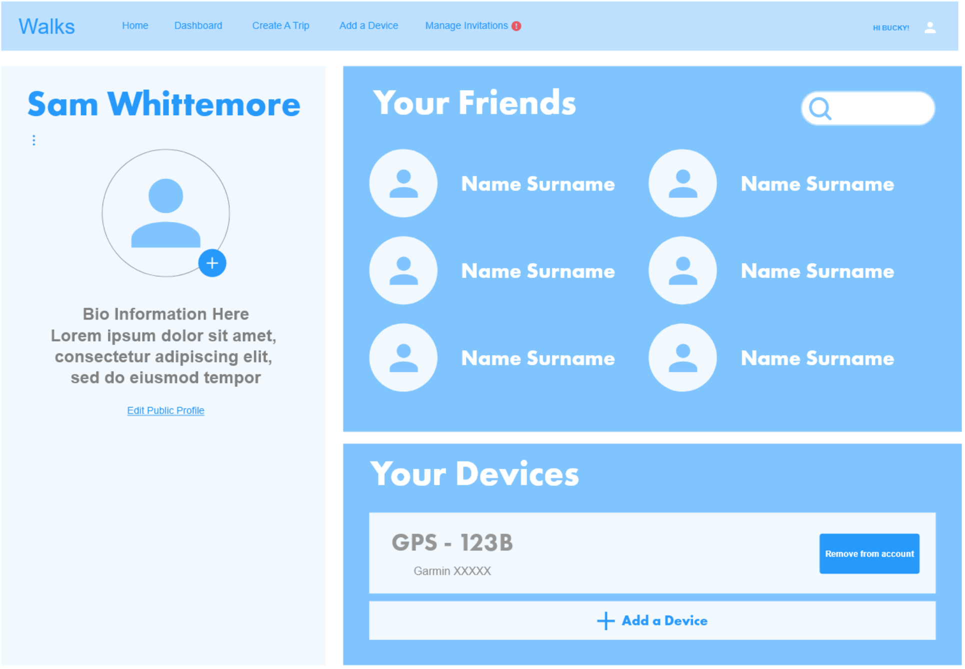
Wireframe for trip page:



9.6 User Profile and Notifications

A user’s view of their own profile page. It allows them to see and edit their profile picture and bio information, see and search for contacts, and to manage and add their own personal GPS devices.

Wireframe for user profile page:

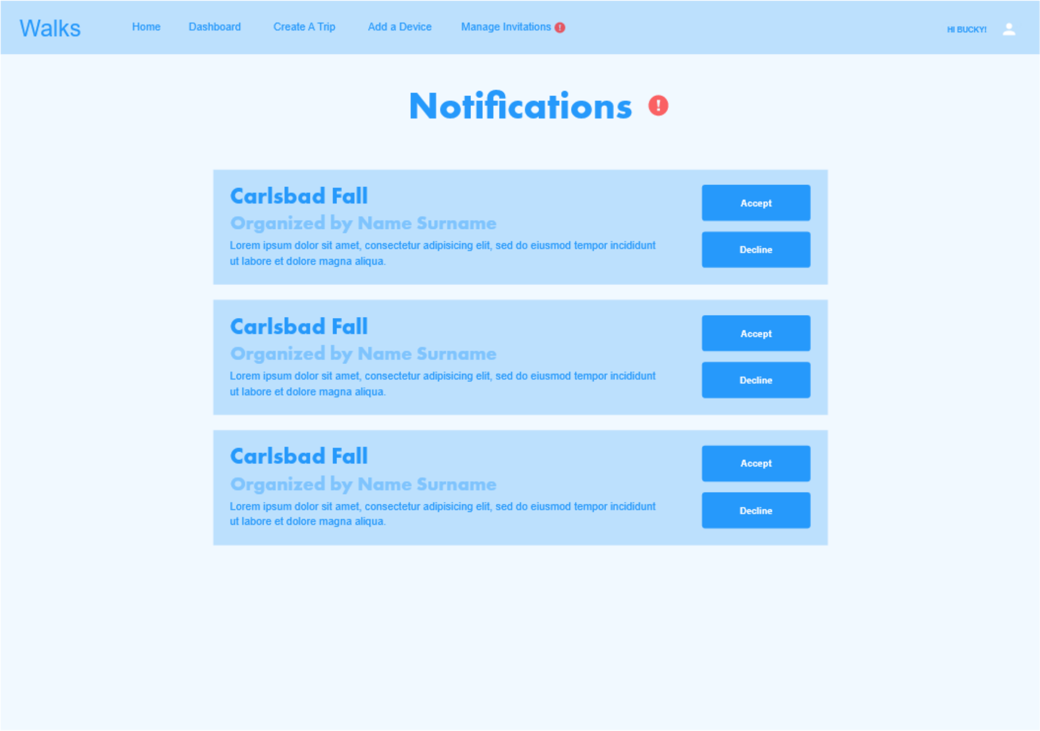


Bio Information Here
Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor

9.7 Notifications Page

A user’s view of their own notification page. This page allows users to manage incoming notifications for friend requests, and rip and spectator invitations.

Wireframe for user notification page:



Notifications

Carlsbad Fall

Organized by Name Surname

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

Carlsbad Fall

Organized by Name Surname

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

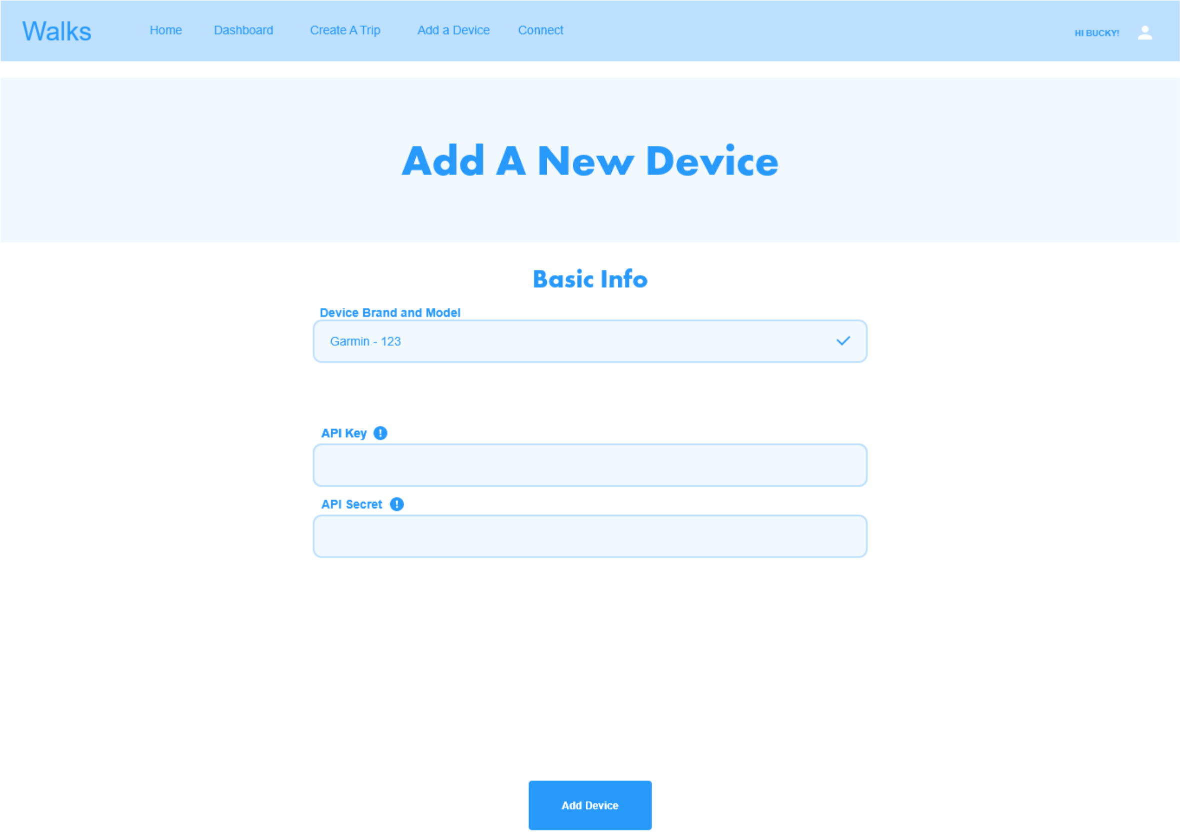
Carlsbad Fall

Organized by Name Surname

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

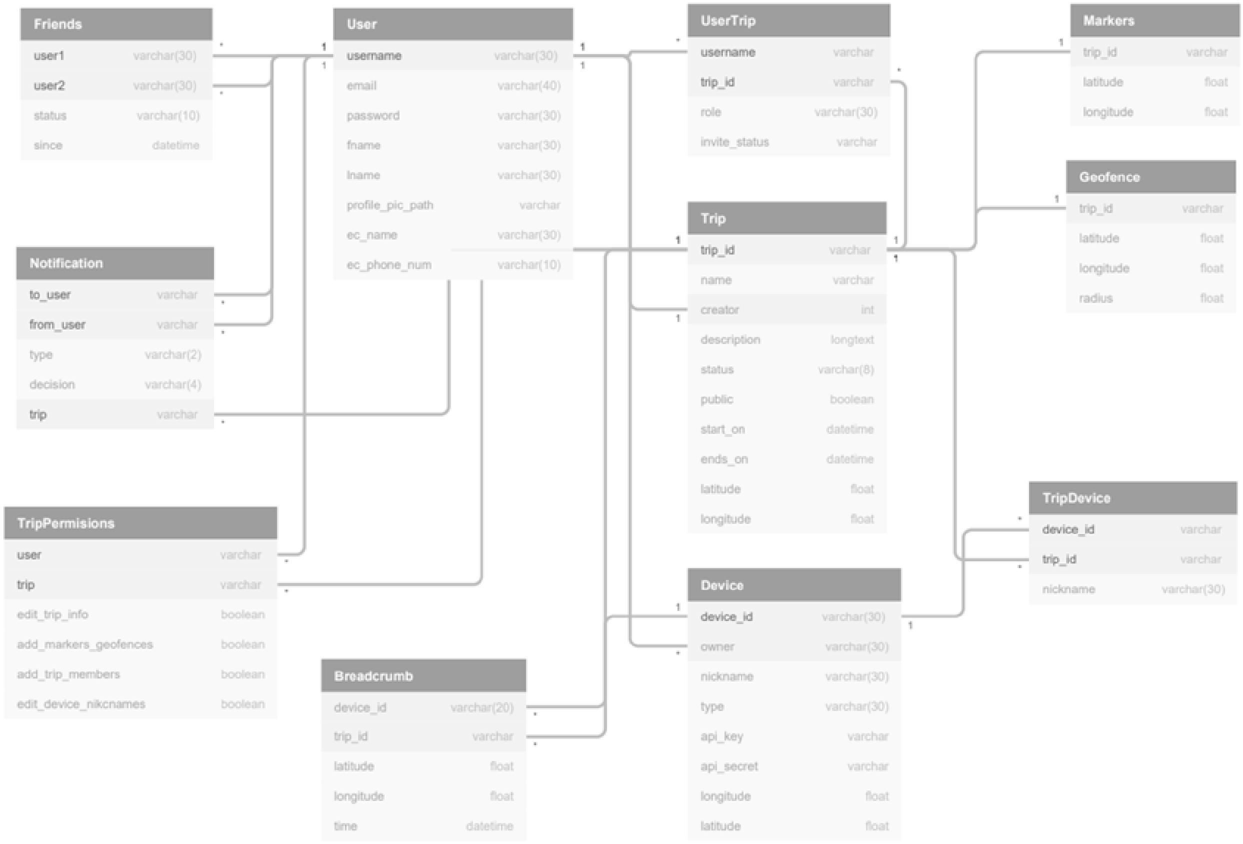
9.8 Add a GPS device form

Example of a form a user could use to add a GPS device to their account. The information fields and directions provided on this page would be specific to the brand and type of device specified in the top dropdown field.



Device Brand and Model: Garmin - 123

10. Data Model



11. Design Constraints

The backend of the web application was developed using python and utilized the web development framework Django. The frontend was developed using JavaScript by deploying the React.js framework. The database portion of the application makes use of SQLite to store all necessary data. Additionally we have multiple third party API's implemented throughout the application provided by Geoforce, Visual Crossing, and Google. The overall functionality of the application is dependent on these various frameworks, services, and API constraints.

12. Mobile Application

12.1 User Functionality

The mobile application is a single-page application with custom built components to retrieve mobile GPS data and send the data to the correct user profile. The primary function of the mobile application is to retrieve mobile GPS data and display the data as a device on a specific user's profile and trips.

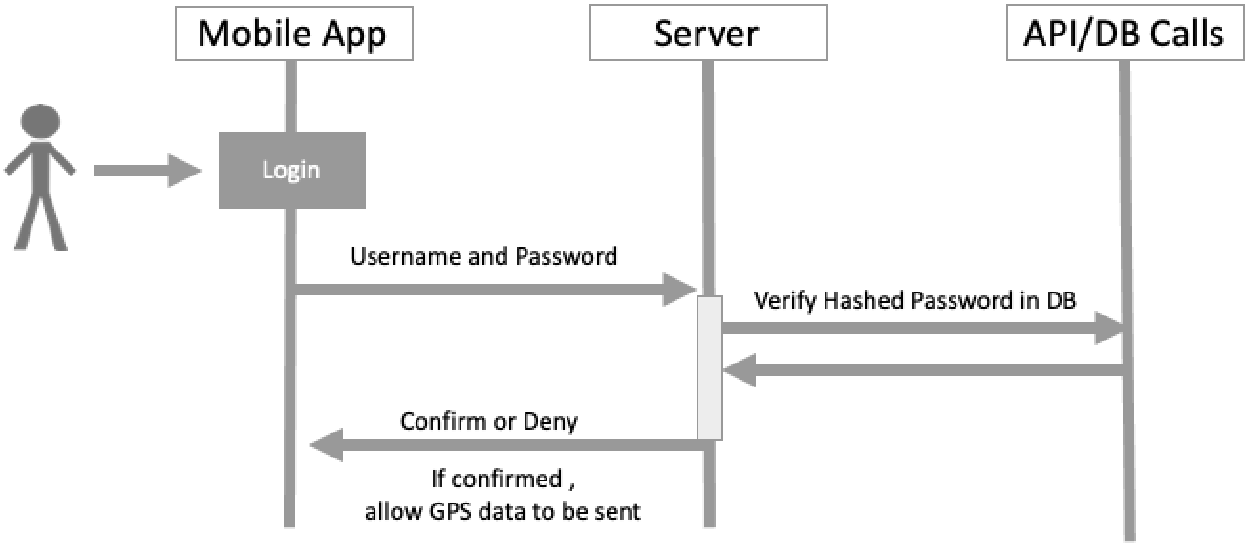
12.2 Development Tools

- **Git -** enables collaboration among team members while keeping development separated using version control. Git will keep track of all additions, updates, and deletions.
- **Xcode 13 -** enables team development features by directly integrating Git and other collaboration features. Xcode also offers helpful functionality that allows iOS apps to request GPS data and control data in a secure way.

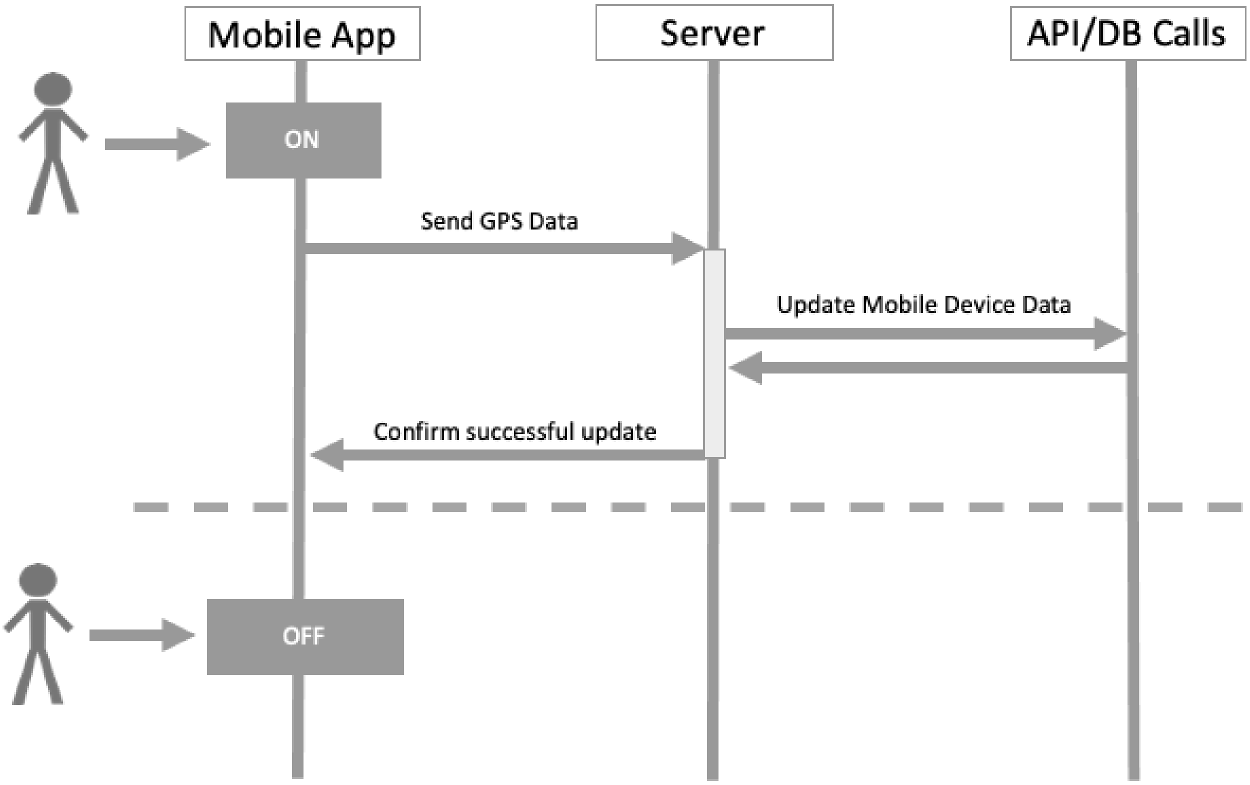
12.3 Mobile App Sequence Diagrams

The following diagrams describe the interactions between the mobile application, Walks server, and Apple Standard location service. The goal of these diagrams are to give visual representation of how data flows for the mobile application. These models are then utilized to implement the various components of the mobile application.

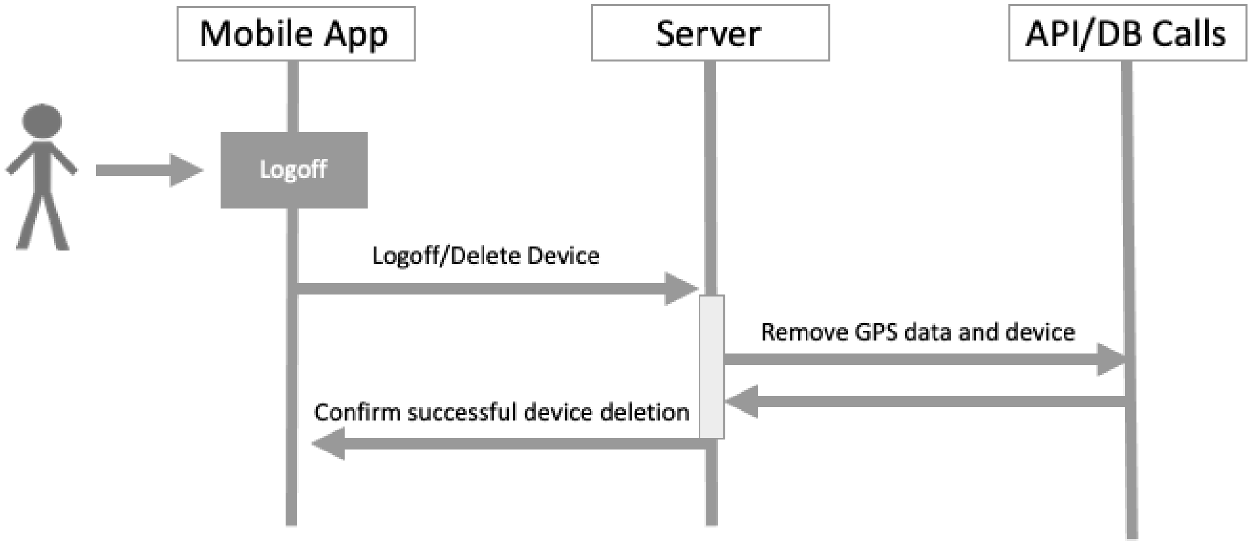
12.3.1 Login Interaction



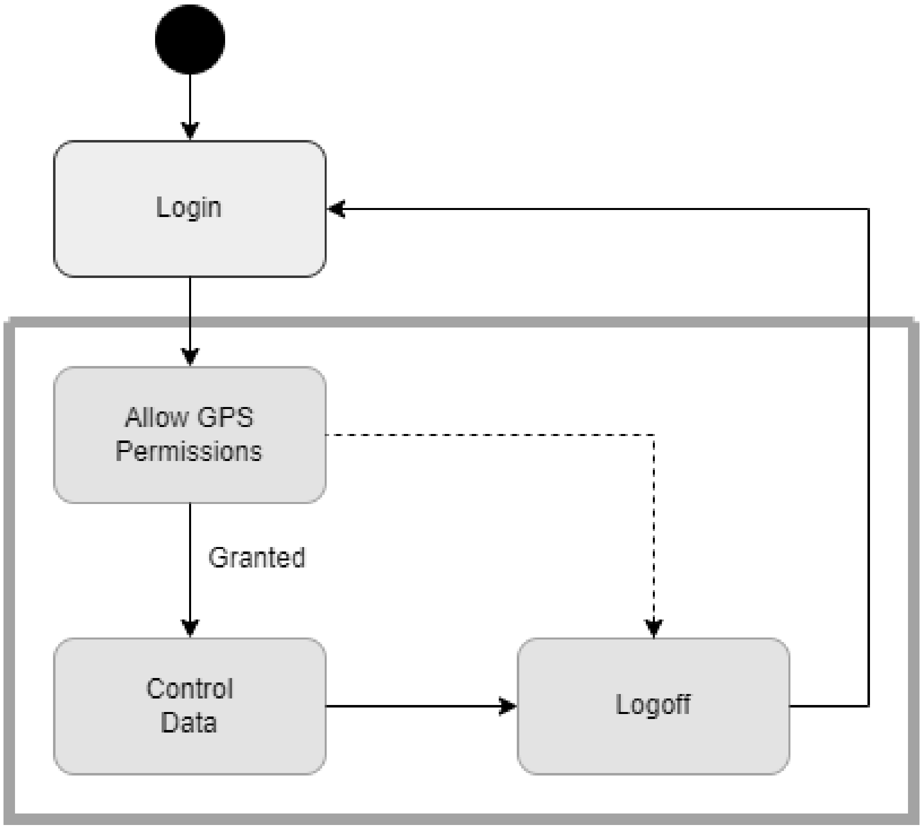
12.3.2 GPS Control Interaction



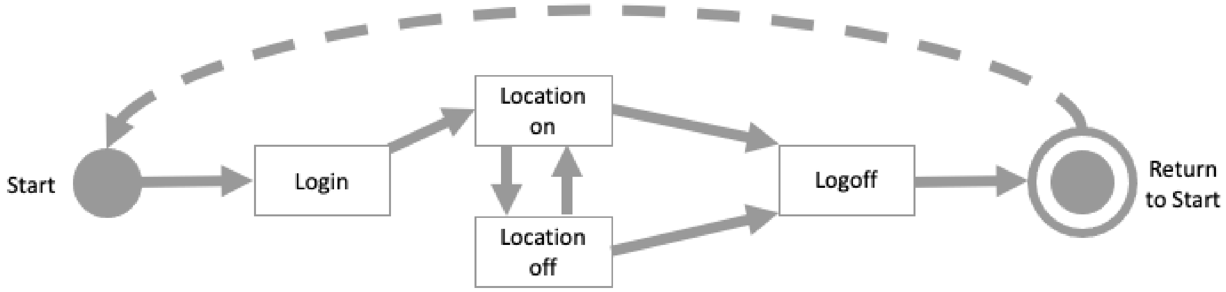
12.3.3 Logout/Remove Device Interaction


Mobile App
Server
API/DB Calls
Logoff
Logoff/Delete Device
Remove GPS data and device
Confirm successful device deletion

12.3.4 Page Flow Diagram



12.3.5 State Diagram



Implementation and Testing

13. Automation of Populating Database with Data

Throughout the implementation and testing of the application it was useful to have some amount of data in the database. Through writing a script to populate the database with data it allowed for the application to be put in a state with known data in the database. Before the script, it was required that the developer manually create users, trips, devices, markers, etc. This was inefficient and when the database needed to be deleted (for migrating purposes), all these entities had to be created again.

14. Comprehensive Testing

14.1 Account Creation Test

14.1.1 Create Account: Verify Account Creation

Test Author: Kerighan Wheeler
Description: Verify that new users with valid information can create a unique account with Walks.

Preconditions: User has internet connection and has successfully loaded the web application on their browser

Step	Test Steps	Test Data	Expected Result	Actual Result	Status	Notes
1	Navigate to the Walks home page		Page and elements fully load	Page and elements fully load	PASS	
2	Click "Sign up button"		Click should navigate user to sign up form page	navigated user to sign up form page	PASS	
3	Fill in valid data (Email address, first name, last name, username, password),	"kellenw556@gmail.com", "Kerighan", "Wheeler", "Keurig23", "t3st*Pswd",	Data fields accept data inputs, submission of valid data goes through to the backend. Page redirects to sign in form.	Data fields accept data inputs, submission of valid data goes through to the backend. Page redirects to sign in form.	PASS	

	click “sign up” button to submit.					
4	Sign in to the new account using test data. (Username, password)	“Kellenw556@gmail.com”, “t3st*Pswd”	Page should redirect to the user's unique dashboard. No trips should populate since the user is new.	Page should redirect to the user's unique dashboard. No trips should populate since the user is new.	PASS	

Postcondition: User has a unique account associated with their email address. The user has the ability to join or create trips, add devices, etc.

14.1.2 Create Account: Empty Data Fields

Test Author: Kerighan Wheeler
Description: Verify that new users with valid information can create a unique account with Walks

Preconditions: User has internet connection and has successfully loaded the web application on their browser

Step	Test Steps	Test Data	Expected Result	Actual Result	Status	Notes
1	Navigate to the Walks home page		Page and elements fully load	Page and elements fully load	PASS	
2	Click “Sign up button”		Click should navigate user to sign up form page	Navigated user to sign up form page	PASS	
3	Fill in valid data for each field but one. Alternate which field is left blank. (Email address, first name, last name,	“kellenw556@gmail.com”, “Kerighan”, “Wheeler”, “Keurig23”, “t3st*Pswd” {Alternating each field as blank}	Empty Data field prevents user from submitting the form and creating their account	Empty Data field prevents user from submitting the form and creating their account	PASS	

	username, password), click “sign up” button to submit.					
1	Navigate to the Walks home page		Page and elements fully load	Page and elements fully load	PASS	

Postcondition: User cannot create an account with any field left blank on the registration form. They cannot access any other pages of the application other than the home screen and sign in pages.

14.1.3 Create Account: Invalid Data Fields

Test Author: Kerighan Wheeler
Description: Verify that new users with valid information can create a unique account with Walks

Preconditions: User has internet connection and has successfully loaded the web application on their browser

Step	Test Steps	Test Data	Expected Result	Actual Result	Status	Notes
Step 1	**Test Steps** Navigate to the Walks home page	**Test Data**	**Expected Result** Page and elements fully load	**Actual Result** Page and elements fully load	**Status** PASS	**Notes**
2	Click “Sign up button”		Click should navigate user to sign up form page	Click should navigate user to sign up form page	PASS	
3	Fill in INVALID data for each field (Email address, first name, last name,	“123”, “123”, “123”, “123”, “123”, “123” {Alternating each field as blank}	email address without “...@emailprovider” shouldn’t allow submission of form. password less than 8 characters should	email address without “...@emailprovider” shouldn’t allow submission of form. password less than 8 characters should not	PASS	

	username, password), click “sign up” button to submit.		not allow submission of form non-uniuqe username will not allow user to submit form.	allow submission of form non-uniuqe username will not allow user to submit form.		

Postcondition: User cannot create an account with an invalid email address or a password less than 8 characters in length. Every field on the registration form should be filled out for the user to submit the form and create their account.

14.1.4 Create Account: Non-Unique Data Fields

Test Author: Kerighan Wheeler
Description: Verify that new users with valid information can create a unique account with Walks

Preconditions: User has internet connection and has successfully loaded the web application on their browser

Step	Test Steps	Test Data	Expected Result	Actual Result	Status	Notes
1	Navigate to the Walks home page		Page and elements fully load	Page and elements fully load	PASS	
2	Click “Sign up button”		Click should navigate user to sign up form page	Navigated user to sign up form page	PASS	
3	Fill in INVALID, NON-UNIQUE data for each field (Email address, first name, last name, username, password),	“kellenw556@gmail.com”, “Kerighan”, “Wheeler”, “Keurig23”, “t3st*Pswd”, “Kenny Wheeler”, “972-691-1614”	Non unique email addresses or usernames should not allow the user to submit the registration form and create an account. Passwords, names, do not need to be unique.	Non unique email addresses or usernames should not allow the user to submit the registration form and create an account. Passwords, names do not need to be unique.	PASS	

	click “sign up” button to submit.					

Postcondition: User cannot create an account with a non-unique email address or password. User with invalid registration fields cannot access other pages of the application other than the home and sign in pages.

14.2 Adding Device

14.2.1 Add Geoforce Device

Test Author: Lance Fletcher
Description: Verifying that the add device page accepts and processes valid inputs and rejects invalid inputs

Preconditions: User has successfully signed in

Step	Test Steps	Test Data	Expected Result	Actual Result	Status	Notes
1	Navigate to Add Device Page					
2	Click “Add Device” Button		Page should prompt use that they must fill in all of the form fields	Page prompted user to fill in required fields	PASS	
3	Provide Device Name	Name: “Test Device”	No errors thrown	No errors thrown	PASS	
4	Select Brand	Brand: “Geoforce”	No errors thrown	No errors thrown	PASS	
5	Input Serial Number	Input Valid serial num	No errors thrown	No errors thrown	PASS	
6	Input Invalid API Key	API Key: “1234”	No errors thrown	No errors thrown	PASS	
7	Input Invalid API Secret	API Secret: “Wrong”	No errors thrown	No errors thrown	PASS	

8	Click “Add Device” Button		Page should prompt the user that they key and secret provided are not legitimate	Prompt displayed	PASS	
9	Change API Key	Input Valid API Key	No errors thrown	No errors thrown	PASS	
10	Change API Secret	Input Valid API Secret	No errors thrown	No errors thrown	PASS	
11	Click “Add Device” Button		“Device Successfully Added” Popup appears then page sends you to trip dashboard and popup should say	Popup appears then redirected to dashboard page	PASS	
12	Navigate to profile page to verify device has been added to account		Device should be present under “Devices” section on profile page	New device listed under “Devices”	PASS	

Postcondition: User now has a Geoforce device added to their account.

14.2.2 Add SPOT Device

Test Author: Lance Fletcher
Description: Verifying that the add device page accepts and processes valid inputs and rejects invalid inputs

Preconditions: User has successfully signed in

Step	Test Steps	Test Data	Expected Result	Actual Result	Status	Notes
1	Navigate to Add Device Page					
2	Click “Add		Page should prompt	Page prompted user	PASS	

	Device” Button		use that they must fill in all of the form fields	to fill in required fields		
3	Provide Device Name	Name: “Test Device”	No errors thrown	No errors thrown	PASS	
4	Select Brand	Brand: “SPOT”	No errors thrown	No errors thrown	PASS	
5	Input Serial Number	Serial: “987654321”	No errors thrown	No errors thrown	PASS	
6	Input Invalid API Key	API Key: “1234”	No errors thrown	No errors thrown	PASS	
7	Input Invalid API Secret	API Secret: “Wrong”	No errors thrown	No errors thrown	PASS	
8	Click “Add Device” Button		Page should prompt the user that SPOT Device is coming soon and not supported.	Prompt displayed	PASS	

Postcondition: User did not add spot device.

14.2.3 Add Garmin Device

Test Author: Lance Fletcher
Description: Verifying that the add device page accepts and processes valid inputs and rejects invalid inputs

Preconditions: User has successfully signed in

Step	Test Steps	Test Data	Expected Result	Actual Result	Status	Notes
1	Navigate to Add Device Page				PASS	
2	Click “Add Device” Button		Page should prompt use that they must fill in all of the form fields	Page prompted user to fill in required fields	PASS	

3	Provide Device Name	Name: “Test Device”	No errors thrown	No errors thrown	PASS	
4	Select Brand	Brand: “Garmin”	No errors thrown	No errors thrown	PASS	
5	Input Serial Number	Input Valid serial num	No errors thrown	No errors thrown	PASS	
6	Input Invalid API Key	API Key: “1234”	No errors thrown	No errors thrown	PASS	
7	Input Invalid API Secret	API Secret: “Wrong”	No errors thrown	No errors thrown	PASS	
8	Click “Add Device” Button		Page should prompt the user that SPOT Device is coming soon and not supported.	Prompt displayed	PASS	

Postcondition: User did not add a Garmin device to their account.

14.3 Add Device Popup

14.3.1 Verify Add Device Popup

Test Author: Spencer Parton
Description: Verifying that the add device popup adds or removes devices from the trip

Preconditions: User has successfully signed in and is on the trip manager page. It is also assumed the user has previously registered one or more devices to their account.

Step	Test Steps	Test Data	Expected Result	Actual Result	Status	Notes
1	Open Popup: Click Add Device Button	Personal Devices: GTO, Backpack Device	Popup will be opened with a checkbox to the left of the devices name.	Popup opened with a checkbox to the left of the devices name.	PASS	Will load all of the users personal devices.
2	Close Popup: Click Out of Popup	Personal Devices: GTO, Backpack Device. GTO checkbox is True	Nothing Happens. Popup stays open	Nothing happened. Popup stayed opened	PASS	We only want the popup to go away when a user clicks on either *Cancel* or

						Submit
3	Close Popup: Click Cancel	Personal Devices: GTO, Backpack Device. GTO checkbox is True. Backpack checkbox is True however it is not “on the trip”	After clicking cancel, the page is reloaded and add device is clicked again. GTO will show as True and Backpack still False	After clicking cancel, the page is reloaded and add device is clicked again. GTO shows as True and Backpack is still False	PASS	Current state should remain after clicking cancel and the data should not be sent to backend.
4	Close Popup: Click Submit	Personal Devices: GTO, Backpack Device. GTO checkbox is True. Backpack checkbox is True	Personal Devices: GTO, Backpack Device. GTO checkbox is True. Backpack checkbox is True	Personal Devices: GTO, Backpack Device. GTO checkbox is True. Backpack checkbox is True	PASS	Device is now assigned to trip.

Postcondition: User now has a Geoforce device added to their account.

14.4 Browser Load

14.4.1 Browser Load: Google Chrome

Test Author: Kerighan Wheeler
Description: Verify Walks web application pages load on multiple browsers

Preconditions: User has internet connection and Google Chrome, Safari, or Firefox browsers installed on their machine

Step	Test Steps	Test Data	Expected Result	Actual Result	Status	Notes
1	Navigate to the Walks home page on Google Chrome		Page and elements fully load	Page and elements fully load		
2	Navigate to sign up page		Page and elements fully load	Page and elements fully load	PASS	

3	Navigate to log in page (log in with valid credentials)	“lance@test.com”, “12345678”	Page and elements fully load	Page and elements fully load	PASS	
4	Navigate to trip dashboard		Page and elements fully load	Page and elements fully load	PASS	
5	Navigate to create a trip page		Page and elements fully load	Page and elements fully load	PASS	
6	Navigate to trip manager page for “Joshua Tree”, by clicking trip icon on trip dashboard		Page and elements fully load	Page and elements fully load	PASS	
7	Click every tab and dropdown on trip manager page to ensure rendering		Elements fully load and are visible	Page and elements fully load	PASS	
8	Navigate to Profile Page via nav bar		Page and elements fully load	Page and elements fully loaded after refresh	PASS	
9	Navigate to a friend’s profile page via friends list		Page and elements fully load	Page and elements fully load	PASS	
10	Navigate to Add Device page via nav bar		Page and elements fully load	Page and elements fully load	PASS	
11	Navigate to How To page via add device page		Page and elements fully load	Page and elements fully load	PASS	

12	Log out		Page and elements fully load	Page and elements fully load	PASS	

Postcondition: User can access every page of the application using the Google Chrome Browser

14.4.2 Browser Load: Safari on Mac

Test Author: Kerighan Wheeler
Description: Verify Walks web application pages load on multiple browsers

Preconditions: User has internet connection and Google Chrome, Safari, or Firefox browsers installed on their machine

Step	Test Steps	Test Data	Expected Result	Actual Result	Status	Notes
1	Navigate to the Walks home page on the Safari Browser		Page and elements fully load	Page and elements fully load	PASS	
2	Navigate to sign up page		Page and elements fully load	Page and elements fully load	PASS	
3	Navigate to log in page (log in with valid credentials)	"lance@test.com", "12345678"	Page and elements fully load	Page and elements fully load	PASS	
4	Navigate to trip dashboard		Page and elements fully load	Page and elements fully load	PASS	
5	Navigate to create a trip page		Page and elements fully load	Page and elements fully load	PASS	
6	Navigate to trip manager page for "Joshua Tree", by clicking trip		Page and elements fully load	Page and elements fully load	PASS	

	icon on trip dashboard					
7	Click every tab and dropdown on trip manager page to ensure rendering		Elements fully load and are visible	Page and elements fully load	PASS	
8	Navigate to Profile Page via nav bar		Page and elements fully load	Page and elements fully load	PASS	
9	Navigate to a friend’s profile page via friends list		Page and elements fully load	Page and elements fully load	PASS	
10	Navigate to Add Device page via nav bar		Page and elements fully load	Page and elements fully load	PASS	
11	Navigate to How To page via add device page		Page and elements fully load	Page and elements fully load	PASS	
12	Log out		Page and elements fully load	Page and elements fully load	PASS	

Postcondition: User can access every page of the application using the Safari Browser

14.4.3 Browser Load: Mozilla Firefox

Test Author: Kerighan Wheeler
Description: Verify Walks web application pages load on multiple browsers

Preconditions: User has internet connection and Google Chrome, Safari, or Firefox browsers installed on their machine

Step	Test Steps	Test Data	Expected Result	Actual Result	Status	Notes
1	Navigate to the					

	Walks home page on Mozilla Firefox		Page and elements fully load	Page and elements fully load	PASS	
2	Navigate to sign up page		Page and elements fully load	Page and elements fully load	PASS	
3	Navigate to log in page (log in with valid credentials)	“lance@test.com”, “12345678”	Page and elements fully load	Page and elements fully load	PASS	
4	Navigate to trip dashboard		Page and elements fully load	Page and elements fully load	PASS	
5	Navigate to create a trip page		Page and elements fully load	Page and elements fully load	PASS	
6	Navigate to trip manager page for “Joshua Tree”, by clicking trip icon on trip dashboard		Page and elements fully load	Page and elements fully load	PASS	
7	Click every tab and dropdown on trip manager page to ensure rendering		Elements fully load and are visible	Page and elements fully load	PASS	
8	Navigate to Profile Page via nav bar		Page and elements fully load	Page and elements fully load	PASS	
9	Navigate to a friend’s profile page via friends list		Page and elements fully load	Page and elements fully load	PASS	
10	Navigate to		Page and elements	Page and elements	PASS	

	Add Device page via nav bar		fully load	fully load		
11	Navigate to How To page via add device page		Page and elements fully load	Page and elements fully load	PASS	
12	Log out		Page and elements fully load	Page and elements fully load	PASS	

Postcondition: User can access every page of the application using the Mozilla Firefox Browser

14.5 Set Permissions Popup

14.5.1 Verify setting of trip members permissions

Test Author: Spencer Parton
Description: This test ensures that the permissions for the trip members that were added by means of create trip dropdown box are able to be adjusted and saved before submission of trip data.

Preconditions: User has entered all valid entries for create trip fields and has clicked submit button.

Step	Test Steps	Test Data	Expected Result	Actual Result	Status	Notes
1	Did Popup Open: No members selected	Empty List of Members	Popup does not open. Trip data is immediately submitted.	Popup does not open. Trip data is immediately submitted.	PASS	Since no members were added to trip permissions do no need ot be set and data is promptly submitted. Popup is not needed.

2	Did Popup Open: One or more trip members selected	Lance Fletcher, Kerighan Wheeler	Popup Opens with each name and 3 radio button options beneath each name. 3 radio button options are: Participant, Manager and Spectator. Radio button should show as selected already for Participant	Popup Opens with each name and 3 radio button options beneath each name. 3 radio button options are: Participant, Manager and Spectator. Radio button should show as selected already for Participant	PASS	Popup should show up if user selects one or more members in create trip page.
3	Changing Permissions: Set to Manager	Lance Fletcher, Kerighan Wheeler	Radio button selection will move to Manager value. Upon submission role will be set to Manager instead of Participant.	Radio button selection will move to Manager value. Upon submission role will be set to Manager instead of Participant.	PASS	Changing the radio button selection will alter a members role (permissions) for the trip.
4	Changing Permissions: Set to Manager	Lance Fletcher, Kerighan Wheeler	Radio button selection will move to Spectator value. Upon submission role will be set to Spectator instead of Participant.	Radio button selection will move to Spectator value. Upon submission role will be set to Spectator instead of Participant.	PASS	Changing the radio button selection will alter a members role (permissions) for the trip.
5	Clicking off Popup window	N/A	If user clicks outside of popup nothing should happen.	Nothing happens when click out outside the dialog box.	PASS	We only want for the submission to happen when user clicks on *Done* button
6	Submit Data/Permissions: Click on Done	N/A	If user clicks on *Done* permissions will be save, the data will be sent to backend, the popup window will be closed and the user will be redirected to the *Trip Dashboard* Page.	When user clicks on *Done* permissions are saved, the data is sent to backend, the popup window is closed and the user is redirected to the *Trip Dashboard* Page.	PASS	Once the permissions have been set by the user and they click done all trip data should be posted to backend and the user is now able

						to see the trip that they just created in the *Trip Dashboard* page.

Postcondition: User has successfully created a trip that shows up in the dashboard.

14.6 Create Trip

14.6.1 Verify successful trip creation

Test Author: Spencer Parton
Description: Test the create trip page

Preconditions: User has logged in successfully.

Step	Test Steps	Test Data	Expected Result	Actual Result	Status	Notes
1	Provide Valid Trip Name: Empy	Empty Value	Form Error on Submission. “*Please Enter Trip Name*”	Form Error on Submission. “*Please Enter Trip Name*”	PASS	Trip name Cant be Null
2	Provide Valid Trip Name: Long Name	*jfdls;ajfljewoajfeiwjajfioea;jfldjlajfkljelaw;jjflkejalkfdsja;fjdajclkjafjeoijajfjeal;jflejaojfjewajflkejljfioeojjf;eljaiiofeajfeaj;lfjieojfje;jiejaoijfelkwjflkejiofj;ewajiofeajiojfe;wajfioewjaifje;jlkkjojfejaoijfioea*	Submission Success	Submission Success	PASS	There are formatting errors with long trip names. Not ideal but does not break website.
3	Provide Valid Trip Name: Good	Testing Name	Submission Success	Submission success provided successful entry of rest of data	PASS	Trip name can be anything else but null

4	Provide Valid Trip Description: Empty	Empty Value	Form Error on Submission. “*Please Enter Trip Description*”	Form Error on Submission. “*Please Enter Trip Description*”	PASS	Trip description cant be null
5	Provide Valid Trip Description: Good	This is a test	Submission Success	Submission Success	PASS	As long as trip name is not null value is ok.
6	Provide Valid Start Date: Null	Empty Value	Form Error on Submission. “*Please Enter Start Date*”	Form Error on Submission. “*Please Enter Start Date*”	PASS	Start Date cannot be null
7	Provide Valid Start Date: Good	04/04/2022	Submission Success	Submission Success	PASS	N/A
8	Provide Valid End Date: Null	Empty Value	Form Error on Submission. “*Please Enter Start Date*”	Form Error on Submission. “*Please Enter Start Date*”	PASS	End date cannot be null
9	Provide Valid End Date: Good	04/08/2022	Submission Success	Submission Success	PASS	N/A
10	Provide Valid Date Range: End Before it Begins	Start: 04/08/2022 End: 04/04/2022	Form Error on Submission. “*Start Date Cannot be After End Date*”	Form Error on Submission. “*Start Date Cannot be After End Date*”	PASS	Start date cannot come after end date
11	Provide Valid Latitude: Empty	Empty Value	Form Error on Submission. “*Please Enter Latitude*”	Form Error on Submission. “*Please Enter Latitude*”	PASS	Latitude cannot be null
12	Provide Valid Latitude: Type String	“Thirty”	Form Error on Submission. “*Please Enter Latitude*”	Form Error on Submission. “*Please Enter Latitude*”	PASS	Latitude cannot be a string, must be a number
12	Provide Valid Latitude: Out of Range	-300	Form Error on Submission. “*Enter Value Between -90 and 90*”	Form Error on Submission. “*Enter Value Between -90 and 90*”	PASS	Valid latitude range is only between -90 and 90
13	Provide Valid Longitude: Empty	Empty Value	Form Error on Submission. “*Please Enter Longitude*”	Form Error on Submission. “*Please Enter Longitude*”	PASS	Longitude cannot be null

	Provide Valid Longitude: Type String	“Thirty”	Form Error on Submission. “*Please Enter Longitude*”	Form Error on Submission. “*Please Enter Longitude*”	PASS	Longitude cannot be a string, must be a number.
14	Provide Valid Longitude: Out of Range	-300	Form Error on Submission. “*Enter Value Between -180 and 180*”	Form Error on Submission. “*Enter Value Between -180 and 180*”	PASS	Valid longitude range is only between -180 and 180
15	Select Available Friends: Drop Down	*Click in field*	List of friends shows up in drop down menu	List of friends shows up in drop down menu	PASS	As long as there is no slow network/problems dropdown box will populate list of users friends.
16	Select Available Friends: Loading	*Click in field: network issues (shut down back end)*	While friends data loads a loading wheel is displayed as right adornment	While friends data loads a loading wheel is displayed as right adornment	PASS	Loading wheel shows that data is being loaded.
17	Select Available Devices: Drop Down	*Click in field: no network issues*	List of users devices shows up in drop down menu	List of users devices shows up in drop down menu	PASS	As long as there is no slow network/problems dropdown box will populate a list of users' devices.
18	Select Available Devices: Drop Down	*Click in field: network issues*	While devices data loads a loading wheel is displayed as right adornment	While devices data loads a loading wheel is displayed as right adornment	PASS	Loading wheel shows that data is being loaded.
19	Public Checkbox	*Select Public Checkbox.* Previous form boxes will be appropriate.	After submission, trip will be categorized as public trip.	After submission, trip will be categorized as public trip.	PASS	Trip showed up in public trip page.

Postcondition: If user added one or more members to the trip then a permissions popup will be displayed. Otherwise data is sent to backend.

14.7 Edit Profile

14.7.1 Verify edit trip popup

Test Author: Spencer Parton
Description: test functionality of edit trip popup

Preconditions: User has valid account set up and is logged in. User has navigated to profile page. It is also assumed that the user registered with profile data that was not null.

Step	Test Steps	Test Data	Expected Result	Actual Result	Status	Notes
1	Click Edit Profile	*Click Button*	Edit Profile popup opens.	Edit Profile popup opens.	PASS	Opens the popup
2	Change First Name	Initial: Lance After: Spencer	First name in text box is Spencer	First name in text box is Spencer	PASS	Testing onChange method for state value
3	Change Last Name	Initial: Fletcher After: Parton	Last name in text box is Spencer	Last name in text box is Spencer	PASS	Testing onChange method for state value
4	Change Bio	Initial: Whether its fishing, camping, hiking, or kayaking, I love them all! After: This is my bio now	Bio is updated to: This is my bio now.	Bio is updated to: This is my bio now.	PASS	Testing onChange method for state value
5	First Name Validation: Null	Empty Value	Form Validation Error: "*First Name Must Have a Value*"	Form Validation Error: "*First Name Must Have a Value*"	PASS	Testing form validation
6	Last Name Validation: Null	Empty Value	Form Validation Error: "*Last Name Must Have a Value*"	Form Validation Error: "*Last Name Must Have a Value*"	PASS	Testing form validation

7	Bio Validation: Null	Empty Value	Form Validation Error: “*Last Name Must Have a Value*”	Form Validation Error: “*Last Name Must Have a Value*”	PASS	Testing form validation
8	Validation: Spamming Submit	Induce a error is any of the 3 fields	After multiple clicks on submit the popup stays open and data is not submitted	Popup stays on and no data is submitted	PASS	Testing soundness of logic for validation
9	Cancel Button	click Cancel button	popup will be closed and profile data will not be updated.	popup will be closed and profile data will not be updated.	PASS	Testing cancel button
10	Submit Button	click Submit Button	Popup will be closed and profile data will be updated to new data	Popup will be closed and profile data will be updated to new data	PASS	Profile data is updated at the local (state) leve as well as on database

Postcondition: User is validated with the database and successfully logs into the account. The account session details are logged in the database

14.8 Logout

14.8.1 Ensure logout revokes access to website

Test Author: Spencer Parton
Description: Test Logout functionality

Preconditions: User has logged in with valid user credentials.

Step	**Test Steps**	**Test Data**	**Expected Result**	**Actual Result**	**Status**	**Notes**
1	Logout	*click logout*	sends user to landing page	sends user to landing page	PASS	
2	Manual Access: Input URL to secured data	URL - https://walkswt.com/tripmanager/2	Access to webpage is denied to user.	Access to webpage is denied to user. Blank white page shown	PASS	Users token should be blacklisted. A white page is not ideal but the security is there.

3	Manual Access: New tab logout	Webpage 1: https://walkswt.com/add-device/ Webpage 2: https://walkswt.com/add-device-walk-through	User is prevented from further access to website.	Access is denied to refresh page or to a new web page. White web page is displayed.	PASS	From UX standpoint white webpage is not ideal but security is maintained.

Postcondition: User is no longer allowed access to secured pages of the website.

14.9 Mobile Device

14.9.1 Verify add Device page

Test Author: Wade Carter
Description: Verifying that the Mobile Application acts properly and allows movement only when valid inputs are given.

Preconditions: User has successfully downloaded and installed the Walks Mobile Application on iOS.

Step	Test Steps	Test Data	Expected Result	Actual Result	Status	Notes
1	Type Valid Username	‘wade@test.com’			PASS	
2	Type Valid Password	‘12345678’			PASS	
3	Click “Login” Button		API response 200 with Access Token and Refresh Token	API response 200 with Access Token and Refresh Token	PASS	
or						
1	Type any Username	‘wade@test.com’			PASS	
2	Leave Password Blank	‘ ’			PASS	

3	Click “Login” Button		Error Message Appears and Missing Value Box Highlights Red	Error Message Appears and Missing Value Box Highlights Red	PASS	
or						
1	Type Invalid Username	‘fail@test.com’			PASS	
2	Type Invalid Password	‘badPassword’			PASS	
3	Click “Login” Button		Error Message Appears and Username/Password Boxes Highlight Red	Error Message Appears and Username/Password Boxes Highlight Red	PASS	Server agrees

Postcondition: Application has received login.

14.9.2 Add Mobile Device (Background Process)

Test Author: Wade Carter
Description: Verifying that the Mobile Application acts properly and allows movement only when valid inputs are given.

Preconditions: User has successfully received a valid Login Access Token and Refresh Token.

Step	Test Steps	Test Data	Expected Result	Actual Result	Status	Notes
1	Send Create Device API	“Unique UUID”	Response Code 200 and DeviceId returned then stored Show Location View	Response Code 200 and DeviceId returned then stored Show Location View	PASS	
or						
1	Send Create Device API	“Non-Unique UUID”	Response Code 400 Error Message in login view “Network/Server	Response Code 400 Error Message in login view “Network/Server	PASS	Testable if application is removed and different user

			Error”	Error”		attempts login

Preconditions: User now has been placed in the location view to control data.

14.9.3 Send Location Data

Test Author: Wade Carter
Description: Verifying that the Mobile Application acts properly and allows movement only when valid inputs are given.

Preconditions: User has successfully received a valid Login Access Token and Refresh Token from Login and has received a valid device ID from mobile device creation. User has also allowed access for Location Data to be retrieved.

Step	Test Steps	Test Data	Expected Result	Actual Result	Status	Notes
1	Toggle “Control Data” OFF		Location Data stops being retrieved and sent	Location Data stopped being retrieved and sent	PASS	Server agrees
or						
1	Toggle “Control Data” ON		Location Data gets retrieved and sent	Location Data gets retrieved and sent	PASS	Server agrees

Preconditions: User has complete control over their data.

14.9.4 Logout/Remove Device

Test Author: Wade Carter
Description: Verifying that the Mobile Application acts properly and allows movement only when valid inputs are given.

Preconditions: User has successfully received a valid Login Access Token and Refresh Token from Login and has received a valid device ID from mobile device creation.

Step	Test Steps	Test Data	Expected Result	Actual Result	Status	Notes

1	Click “Logout” Button					
2.2.1	Send Logout API Request		Response Code 200 [Continue to 2.2.2]	Response Code 200	PASS	
2.2.2	Remove saved data		Return to Login View	Return to Login View	PASS	
or						
1	Click “Wade’s Mobile” from Profile Page on Web App		Rename Device Popup Appears	Rename Device Popup Appears	PASS	Pre-condition user is logged into same user in Walks Web App
2	Click “DELETE DEVICE” button on Web App		Device is removed from account.	Device is removed from account.	PASS	Backend agrees
3	Click “Logout” Button on mobile application				PASS	
3.2.1	Send Logout API Request		Response Code 500 [Continue to 3.2.2]	Response Code 500	PASS	Server Agrees
3.2.2	Remove saved data		Saved data removed [Continue to 3.2.3]	Return to Login View	PASS	Storage Agrees
3.2.3	Display Alert Popup “Could not connect to server!”		Return to Login View	Popup appears and user returned to login view	PASS	

Preconditions: User has logged out and all data is deleted.

14.10 Page Responsiveness

14.10.1 Page Responsiveness: >=1100px in width

Test Author: Kerighan Wheeler
Description: Verify Walks web application pages are responsive to more than one window size width

Preconditions: User has internet connection and supported browser

Step	Test Steps	Test Data	Expected Result	Actual Result	Status	Notes
1	Navigate to the Walks home page on Google Chrome		Page and elements fully load and are visible and accessible	Page and elements fully load and are visible and accessible	PASS	
2	Navigate to sign up page		Page and elements fully load and are visible and accessible	Page and elements fully load and are visible and accessible	PASS	
3	Navigate to log in page (log in with valid credentials)	"lance@test.com", "12345678"	Page and elements fully load and are visible and accessible	Page and elements fully load and are visible and accessible	PASS	
4	Navigate to trip dashboard		Page and elements fully load and are visible and accessible	Page and elements fully load and are visible and accessible	PASS	
5	Navigate to create a trip page		Page and elements fully load and are visible and accessible	Page and elements fully load and are visible and accessible	PASS	
6	Navigate to trip manager page for "Joshua Tree", by clicking trip icon on trip dashboard		Page and elements fully load and are visible and accessible	Page and elements fully load and are visible and accessible	PASS	

7	Click every tab and dropdown on trip manager page to ensure rendering		Page and elements fully load and are visible and accessible	Page and elements fully load and are visible and accessible	PASS	
8	Navigate to Profile Page via nav bar		Page and elements fully load and are visible and accessible	Page and elements fully load and are visible and accessible	PASS	
9	Navigate to a friend's profile page via friends list		Page and elements fully load and are visible and accessible	Page and elements fully load and are visible and accessible	PASS	
10	Navigate to Add Device page via nav bar		Page and elements fully load and are visible and accessible	Page and elements fully load and are visible and accessible	PASS	
11	Navigate to How To page via add device page		Page and elements fully load and are visible and accessible	Page and elements fully load and are visible and accessible	PASS	
12	Log out		Page and elements fully load and are visible and accessible	Page and elements fully load and are visible and accessible	PASS	

Postcondition: User can access every page of the application using a browser window size greater than 1100px in width.

14.10.2 Page Responsiveness: <1100px in width

Test Author: Kerighan Wheeler
Description: Verify Walks web application pages are responsive to more than one window size width

Preconditions: User has internet connection and supported browser

Step	Test Steps	Test Data	Expected Result	Actual Result	Status	Notes
1	Navigate to the Walks home page on the firefox Browser		Page and elements fully load, are, visible, but are not fully accessible or fully responsive to extremely small pixel sizes	Page and elements fully load, are, visible, but are not be fully accessible or fully responsive to extremely small pixel sizes	PASS	
2	Navigate to sign up page		Page and elements fully load, are, visible, but are not fully accessible or fully responsive to extremely small pixel sizes	Page and elements fully load, are, visible, but are not be fully accessible or fully responsive to extremely small pixel sizes	PASS	
3	Navigate to log in page (log in with valid credentials)	"lance@test.com", "12345678"	Page and elements fully load, are, visible, but are not fully accessible or fully responsive to extremely small pixel sizes	Page and elements fully load, are, visible, but are not be fully accessible or fully responsive to extremely small pixel sizes	PASS	
4	Navigate to trip dashboard		Page and elements fully load, are, visible, but are not fully accessible or fully responsive to extremely small pixel sizes	Page and elements fully load, are, visible, but are not be fully accessible or fully responsive to extremely small pixel sizes	PASS	
5	Navigate to create a trip page		Page and elements fully load, are, visible, but are not fully accessible or fully responsive to extremely small pixel sizes	Page and elements fully load, are, visible, but are not be fully accessible or fully responsive to extremely small pixel sizes	PASS	
6	Navigate to trip manager page for		Page loads error message instructing user to use a larger	Page loads error message instructing user to use a larger	PASS	

	“Joshua Tree”, by clicking trip icon on trip dashboard		browser window to view trip manage page content.	browser window to view trip manage page content.		
7	Click every tab and dropdown on trip manager page to ensure rendering		Page and elements fully load, are, visible, but are not fully accessible or fully responsive to extremely small pixel sizes	Page and elements fully load, are, visible, but are not be fully accessible or fully responsive to extremely small pixel sizes	PASS	
8	Navigate to Profile Page via nav bar		Page and elements fully load, are, visible, but are not fully accessible or fully responsive to extremely small pixel sizes	Page and elements fully load, are, visible, but are not be fully accessible or fully responsive to extremely small pixel sizes	PASS	
9	Navigate to a friend’s profile page via friends list		Page and elements fully load, are, visible, but are not fully accessible or fully responsive to extremely small pixel sizes	Page and elements fully load, are, visible, but are not be fully accessible or fully responsive to extremely small pixel sizes	PASS	
10	Navigate to Add Device page via nav bar		Page and elements fully load, are, visible, but are not fully accessible or fully responsive to extremely small pixel sizes	Page and elements fully load, are, visible, but are not be fully accessible or fully responsive to extremely small pixel sizes	PASS	
11	Navigate to How To page via add device page		Page and elements fully load, are, visible, but are not fully accessible or fully responsive to extremely small pixel sizes	Page and elements fully load, are, visible, but are not be fully accessible or fully responsive to extremely small pixel sizes	PASS	
12	Log out		Page and elements fully load, are,	Page and elements fully load, are,	PASS	

			visible, but are not fully accessible or fully responsive to extremely small pixel sizes	visible, but are not be fully accessible or fully responsive to extremely small pixel sizes		

Postcondition: User can access every page of the application using a browser window size less than 1100px in width.

14.11 Login

14.11.1 Verify Login Functionality

Test Author: Austin Turner
Description: Testing to see if login accepts no data entered

Preconditions: User has created an account

Step	Test Steps	Test Data	Expected Result	Actual Result	Status	Notes
1	Navigate to login page	Click on login button	User lands on the login page	User lands on the login page		
2	Provide username	No data entered	n/a	n/a	PASS	
3	Provide password	No data entered	n/a	n/a	PASS	
4	Login	Click login button	User is not logged in	User is not logged in	PASS	

Postcondition: User is not logged in and must try again

14.12 Page Protection

14.12.1 Verify Page Protection

Test Author: Austin Turner
Description: Ensure a logged in user can't navigate to other random user’s trips via a url

Preconditions: User is logged in

Step	Test Steps	Test Data	Expected Result	Actual Result	Status	Notes
1	Find a trip number and enter it into the url	https://walkswt.com/tripmanager/8	User is denied access and sent back to trip dashbaord	User is sent to page but no API calls are made, meaning no information is shared.	PASS	

Postcondition: User is sent to a page with nothing on it.

14.13 User Testing Model

Because Walks prioritizes the needs and desires of our users in our design and implementation, it follows that the Walks testing process has an emphasis on user testing. Each user approaches our application with a complex variety of goals, assumptions, and experience. Our tests should reveal the areas of our design that do not best meet our users needs and preferences. In addition to our functional testing (much of which is accomplished by using pre-populated data), we underwent some simple live user testing which is outlined below in the following process.

Our test user was prompted with the following information after being directed to walkswt.com:

- “Walks is a web application that allows users to plan and organize trips where they can upload different types of GPS devices. In this test case, you are using Walks because you want to plan a hiking trip with your friend across the state. Show me how you would use the Walks app to do that”.

After being prompted, and with the permission of the user, we recorded their response to the prompt as they worked through the application to accomplish this task. Their response to the prompt revealed some bugs that were not noted in previous testing, as well as the areas in which certain features of the application are less intuitive or frustrating. Specific conclusions from this test are outlined in the chart below. This feedback will be valuable for future iterations of development for the Walks Application.

Observed Behavior	Proposed (NON ESSENTIAL) Fix
Scrolling up and down, rapid clicks through different pages searching for notifications.	Socketed application (more real-time notifications), or adding notification animations, or putting notification link on trip dashboard.

Repeated aimless clicks on map after clicking minus button under geofence tab on Trip Manager page	Include instructions for minus button on Geofence tab or include container for geofences (and potentially markers) to be removed from the side panel rather than directly on the map
Repeated scrolling behaviors on dashboard after account creation	Potentially direct users to different page after account creation to showcase how to information for the application, or include filler information in the case that a user does not have any past, active, or future trips
User had several failed login attempts because of auto populating username rather than email address for login validation	Potentially remove the username field for sign up, or allow users to enter either their username or their email for login validation

15. Development Issues

15.1 Known Bugs

15.1.1 Style Inconsistencies

A persistent issue we have found with our web application is a failure for the customized, Walks themed styles to load initially. An example of this problem manifesting is when a user signs up for our application the buttons on the next web page will be blue rather than green. This issue is related to our use of Material-UI components. Only the styling for a Material-UI component is affected. We believe that this issue is caused by using old, depreciated import statements for the Material-UI components alongside new Material-UI imports.

15.1.2 Google API Loading Error

If a user logs in and then clicks into a trip that they have signed up for in a previous session and then, before the google map loads on that page they click to a different section of our website google maps will throw an error. This will not shut down the website since the server suppresses all development errors but it is still problematic.

15.1.3 JWT Refresh Token

There have been some cases where a refresh token fails to generate and then the user will not be able to interact with our website. At the present time, it seems we have fixed the refresh token error.

15.1.4 Failure to Route to Personal Profile

If a user goes to another user's profile and then clicks the profile button in the NavBar our website will not redirect the user to their own profile, which is presumably what the user would be wanting to do. The profile button within the NavBar is encapsulated by a React.js <Link> component. These allow a user to smoothly navigate to another page within the application rather than using an <a> html element. The URL for the profile page is "https://walkswt.com/profile/1/", where 1 is the user ID. Since the page navigation is handled by a react.js component called router, which is set to send people this the profile page when /profile/ is the specific route, it does not feel the need to send the user a new page. Thus, the useEffect (which handles the API GET requests) on the webpage never executes which causes the user's profile data to be loaded.

15.1.5 Device Not Deleted Upon Removal from Trip

With the help of Dr. Siddiqui, we identified that if a user has a device registered on a trip and then the user removes themselves from the trip the device will remain on the trip.

15.2 Known Vulnerabilities

15.2.1 Sign-up Validation

We currently do not have validation in either the frontend or the backend for any of the input fields on our sign-up page. This could induce an error if a user puts in garbage input. This could be addressed by validating that input is formatted correctly on the frontend and that the information is real on the backend.

15.2.2 Heavy Server Traffic

After stress testing our website we found that under heavy loads the server will begin blocking requests until the traffic dies down.

15.2.3 Unvalidated API Endpoints

We currently have some API endpoints that are not validating that the user is allowed to make the request. This could allow for a user to handcraft an http request to certain API endpoints and access sensitive data that they should not otherwise have access to.

15.3 System Dependencies

15.3.1 Functional Dependencies

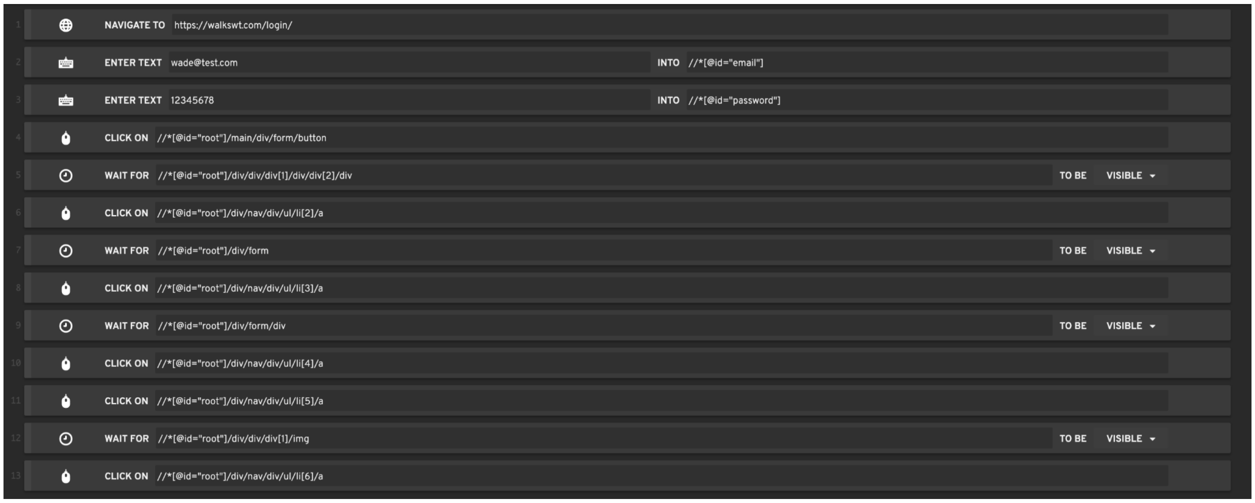
As development is continuing and versions of dependencies change, keeping track of the dependencies will be important as this will reduce errors between development and deployment of the application. When working in the Docker containers, it will contain the same versions of all dependencies that the AWS E2 instance is running. Below is a list of all major Dependencies that are required for applications to execute in Development and Deployment environments.

- Ubuntu Server System Dependencies List:
  - nodejs
  - npm
  - python3-pip
- Python (Pip) Dependencies List:
  - PyJWT==2.3.0
  - Django==3.2.9
  - django-cors-headers==3.10.0
  - django-crispy-forms==1.13.0
  - djangorestframework==3.12.4
  - djangorestframework-simplejwt==5.0.0
  - simplejson==3.17.5

16. Stress Testing of Deployed Server

With our latest version of Walks deployed on an AWS E2 instance we should note that our server is limited in its capacities. To test these capacities we utilized a free online service, loadster, that tests a custom script to simulate a large number of users simultaneously on a browser. In result we tested our servers full capabilities on CPU, RAM, and networking outputs.

16.1 Script Utilized



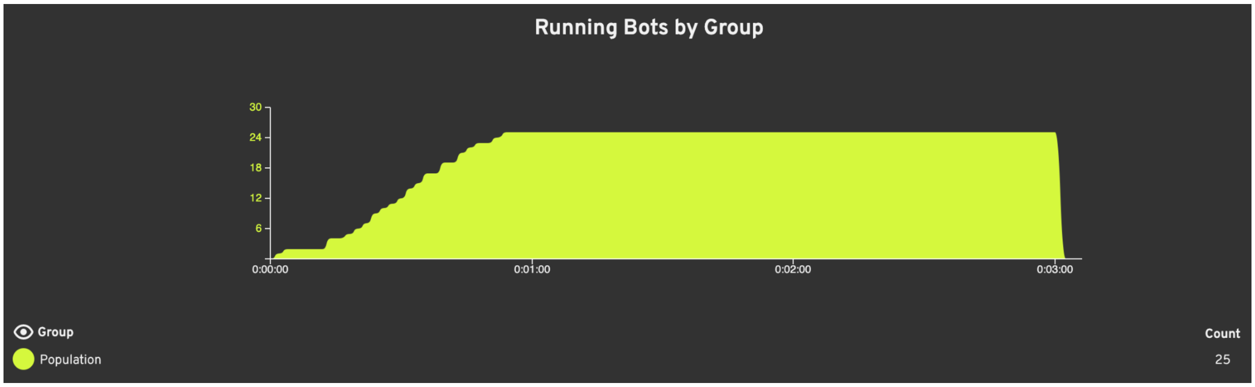
```
NAVIGATE TO  https://walkswt.com/login/
ENTER TEXT  wade@test.com                                  INTO  //*[@id="email"]
ENTER TEXT  12345678                                       INTO  //*[@id="password"]
CLICK ON  //*[@id="root"]/main/div/form/button
WAIT FOR  //*[@id="root"]/div/div/div[1]/div/div[2]/div                 TO BE  VISIBLE
CLICK ON  //*[@id="root"]/div/nav/div/ul/li[2]/a
WAIT FOR  //*[@id="root"]/div/form                                      TO BE  VISIBLE
CLICK ON  //*[@id="root"]/div/nav/div/ul/li[3]/a
WAIT FOR  //*[@id="root"]/div/form/div                                  TO BE  VISIBLE
CLICK ON  //*[@id="root"]/div/nav/div/ul/li[4]/a
CLICK ON  //*[@id="root"]/div/nav/div/ul/li[5]/a
WAIT FOR  //*[@id="root"]/div/div/div[1]/img                            TO BE  VISIBLE
CLICK ON  //*[@id="root"]/div/nav/div/ul/li[6]/a
```

The script that was created to navigate to the login page then enter the main application. Once the bot has entered the application it navigates through each item in the navbar ending with logout. This script does not only bring up static pages but submits API requests that are attached to the buttons resulting in a decent load for one bot in a short amount of time.

16.2 Results

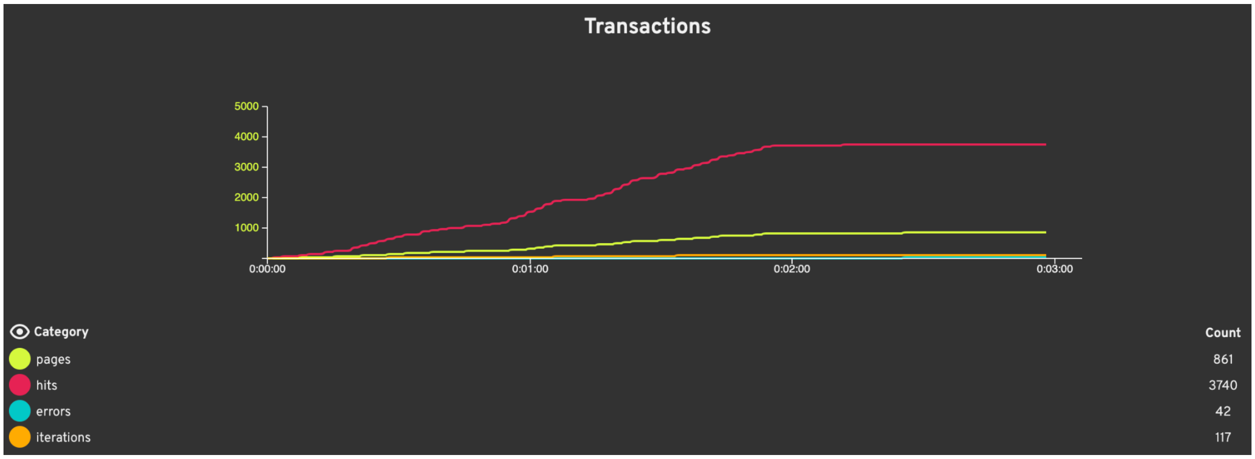
As a result from the testing we can see that the server was able to handle a load of 25 users sending large amounts of requests for a minute before experiencing bottle necking from timeout errors being thrown on the login page. From this test alone it should be noted that the login page does take up a more significant load on our server and could be optimized in the future. Overall the results are impressive for a small AWS EC2 instance running all our required processes.

16.2.1 Created Bots



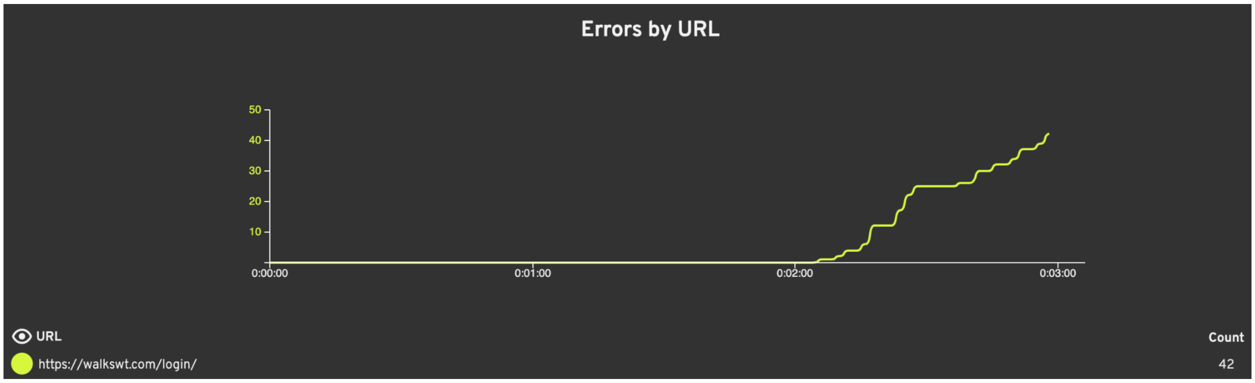
This image represents the number of bots to begin testing over a time of 3 minutes.

16.2.2 Transactions



This image shows the number of transactions by category over the time of 3 minutes.

16.2.3 Errors



This image shows after two minutes that the login page started to experience a timeout error resulting in transactions to begin to plato and come to a halt.

16.3 Future Expansion

If our user base was to ever grow to a larger audience a new server should be utilized with more CPU power and RAM. Another capacity enhancement would be to use a separate server for the database which would reduce the load of the server and use it as a middle man as designed. With the scope of our project we are not anticipating a large user base that would exceed our current servers specs.

17. Backlog

The following is a backlog of the entire project. All items that remain for the overall project are specified as "in progress".

Task #	Priority Level	Task Objectives	Assignee	Status
Task 1	High	Integrate proper GET APIs into Trip Manager to get necessary trip data	Lance	Complete
Task 2	Medium	Implement user profile page to display profile information	Kerighan, Lance	Complete
Task 3	High	Add notifications page to allow accepting of friend requests and trip invitations	Kerighan, Lance	Complete
Task 4	Low	Clean up the current file system. Clean up global.css file	Wade	Complete
Task 5	High	Implement friends functionality of application	Lance	Complete
Task 6	Low	Integrate Weather API into Trip Manager Page	Kerighan, Lance	Complete
Task 7	Medium	Implement ability for user to be able to place trip markers on the map	Lance	Complete
Task 8	High	Secure pages by restricting users from accessing unauthorized data	Austin	Complete
Task 9	High	Improve functionality of Create Trip Page	Spencer	Complete
Task 10	High	Add trip permissions functionality	Lance, Spencer	Complete
Task 11	High	Deploy web application with backend structure	Wade	Complete
Task 12	High	Implement ability to add personal devices to trip	Spencer, Lance	Complete
Task 13	Low	Integrate SPOT Device into application	Lance	In Progress
Task 14	Low	Implement ability for user to upload profile picture	Austin	In Progress

Task 15	High	Implement ability to alter trip, user, device data through PUT requests	Lance, Spencer	Complete
Task 16	Medium	Implement public trips functionality	Kerighan, Lance	Complete
Task 17	Low	Implement feature for map to be centered on desired map component	Lance	Complete
Task 18	Medium	Implement ability to edit profile information	Spencer	Complete
Task 19	High	Implement search ability to application to find other users	Lance	Complete
Task 20	High	Implement Geofences functionality	Lance	Complete
Task 21	High	Validating User Inputs and prompting users with input errors	Austin, Kerighan, Lance, Spencer, Wade	Complete
Task 22	Medium	Mobile - Build loginView and locationView UI experiences	Wade	Complete
Task 23	Medium	Mobile - Build web services and location services classes	Wade	Complete
Task 24	High	Mobile - Implement refresh token, logout, and app restart functionality	Wade	Complete

18. References

Faulhaber, M., Ruedl, G., Schneider, F., Walter, D., Sterr, R., Schobersberger, W., Schwendinger, F., & Pocecco, E. (2020). Characteristics of Victims of Fall-Related Accidents during Mountain Hiking. *International journal of environmental research and public health*, *17*(3), 1115. https://doi.org/10.3390/ijerph17031115

"IEEE Guide for Software Requirements Specifications," in IEEE Std 830-1984 , vol., no., pp.1-26, 10 Feb. 1984, doi: 10.1109/IEEESTD.1984.119205.