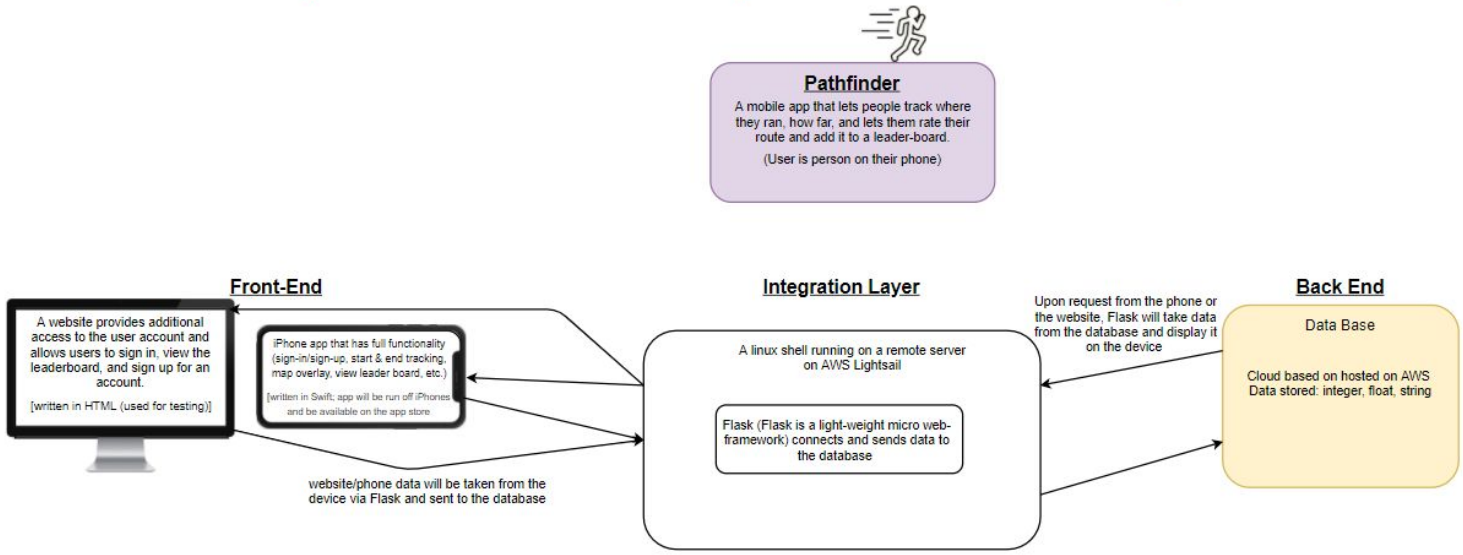
209-7 Agents 007
Emily Parker, Jaryd Meek, Cody Wong, Elena Smith, Manuel Arellano, and Vineet Belur

Milestone 4

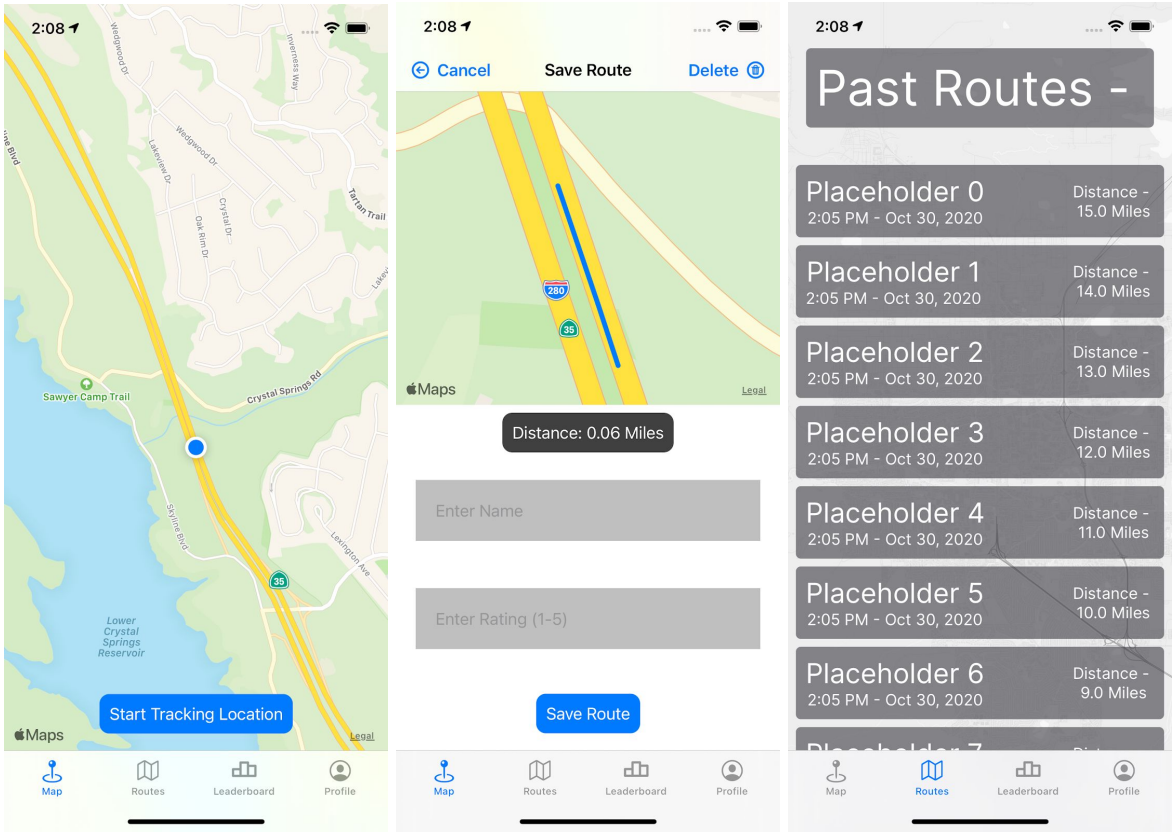
Revised List of Features

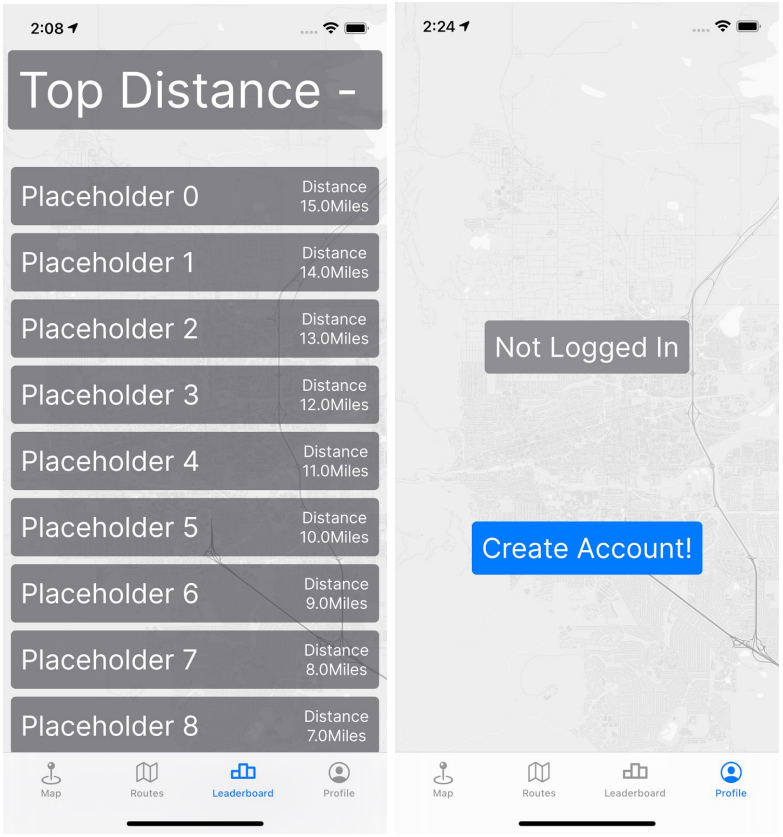
- Sign in
    - Allows ‘sign in’ functionality - which entails creating a personal account w/ user and password. Alongside this, baseline functionality will include resetting your password, resetting your username, etc. Each individual will also be able to store their personal statistics tethered to their account.
    - Salting + hashing passwords implemented for security through swiftui.
    - For the most part already implemented
- Mobile App
    - The ability to access and use the app (with full functionality) via a smartphone
    - Allows you to manage your account, see your route, see your stats, and follow your route with GPS integration.
- Leaderboard
    - Being able to see top distances. Will have local/global results based on location proximity.
- Route rating
    - If you like a certain route you are able to give it a rating out of five, share the route, and have others rate it as well.
- Route saving
    - The ability to save a specific route that has been set by the app to the profile of the user. This allows them to access that specific route again if they so desire.
- Routing NOT LIKELY
    - An intelligent routing algorithm that allows more than just getting from point A to point B, it allows you to just pick a distance you want to go and gives you a path.
    - The last priority to be completed
- Potential Website
    - A possible website to follow the same features of the mobile app, but without routing. For checking accounts and leaderboard. Not definite, but may happen as html pages are being developed to help create functionality of the app.

Architecture Diagram



Front End Design





Web Service Design

Using Flask to create an API to connect to HTML pages and iOS front end through Swift. These are written to connect to a database, which is in the process of being moved from locally to cloud hosted. This API will be used to transfer encrypted login information and route information both to and from the database.

Database Design

Using AWS to host a VM using SQLite + Flask server.

We currently are using two tables in SQLite namely: route & login.

Route consists of four columns: token(integer), user(string(80), email(string(80), and password(string(80))

Login consists of five columns: route_id(integer), distance(float), about(string(500)), rate(integer), useradded(string(80))

We chose to use SQLite over postgres due to the ease of implementation alongside flask, our choice of web service.

Individual Contributions

- Jaryd Meek - Pretty much all iOS app development
- Cody Wong - Created architecture diagram
- Elena Smith - Edited HTML pages; added a page for leaderboard; added a navigation bar for easier access to the other pages.
- Manuel Arellano - Added the connection between the html pages with the nav bar Elena created.
- Emily Parker - Work with flask to connect to iOS app and help research on setting up aws
- Vineet Belur - Set-up aws vm for flask/sqlite3. Some html/python added.

Challenges

- One challenge that we're facing is remembering to update our Jira board. We normally talk over what each of us needs to do during our weekly meetings, so we sometimes forget to update the Jira board because we already have an idea of what we need to do. We will try to remember to keep updating the Jira Board.
- Our biggest challenge is creating a cloud based database. We ended up setting on using lightsail to host a virtual machine to run flask/sqllite3.
- Another challenge is equal workloads per week for each team member since we all have different classes with different amounts of work per week and different midterm schedules. Our solution to that is to let a member of the group have a lighter workload on the project if they have a really busy week, and then have them work normally for the weeks that they're not really busy with other classes.
- We do have a basic version of our project that works, so if all else fails, we can always count on that.