Emily Parker, Elena Smith, Cody Wong, Manuel Arellano, Vineet Belur, Jaryd Meek
Project Milestone 5

Test Plans

Note: For unit tests, our main implementation in flask has been coded in Python3, therefore we will likely use the given frameworks for querying/asserting/validating.

Feature: Login
User enters details and clicks login

A user must provide the following information when creating an account:
- Username
- Email
- Password

If any of the fields are not filled in then there will be a specific error.

Checks to make sure username hasn't been used, checks to make sure email is unique and a valid email address, and that the password and confirm password fields match.

User enters invalid information that does not match the given criteria, the login page displays an error message.

Currently the html form checks for this, our swift code will soon mirror this format.

Feature: Saves
Distance when tracking and show starts and stops (the line basically)
User clicks the save button (only on the app, but we are testing inserting on the demo as well) given it's all numbers that you don't plug in yourself; it should already be in the correct values (Id and Distance). Every user gets a limited amount of saves, so as long as you haven't maxed out it should be fine.

Testing nothing gets overwritten

Verification for saving into the database should not be necessary as users will already be verified once they have signed in.

Per Swift:
Acceptance Criteria:
-username

-distance
-time
-rating

Feature: Leaderboards

The leaderboard is a feature (on the app/website) that shows the top 15 longest routes that have been run by users. It should be accessible on the app and on the website (if we get that far) and will display the username of the account and the total distance that was run. For the app, this also operates under the assumption that the user has already logged in successfully. On the website, it should show the top 15 without a login.

User clicks the leaderboard button on the app (or on the leaderboard tab/button on the navbar of the website) and the app should load the leaderboard page with the top 15 longest routes shown, along with their distance.

The data that will be passed will be a distance calculated from the GPS coordinates collected while tracking. It will take the username and distance from the database.

User completes a certain distance that is lower than all the previous values, the app should not display their contribution.

Acceptance Criteria:
- Top 15 distances
- Usernames associated with them
- Ratings associated with them

Individual Contributions
- Emily Parker - Worked on additions to py flask integration for new features
- Cody Wong - Worked on test cases/descriptions/etc.
- Elena Smith - Updated navigation bar so that it actually works
- Manuel Arellano - Work on descriptions and test cases.
- Vineet Belur - Worked on test cases/acceptance criteria/etc.
- Jaryd Meek - Worked on mobile app including accessing api.

CSCI-3308-FA20-209-7
Classic software project

007-209-7
Board

- Backlog
- Active sprints
- Reports
- Issues
- Components
- Code
- Releases
- Project pages
- Add item
- Project settings

Projects / CSCI-3308-FA20-209-7 / 007-209-7

All sprints

Complete sprint

CW ES EP JM MM VV

Only My Issues · Recently Updated · All sprints ⌄

TO DO	IN PROGRESS	DONE
User's name highlighted — Data Storage & Leaderboard — CSCIF-29	Saving Routes — Route Management — CSCIF-25 CW	User Storage — Registration-Page — CSCIF-32 EP
User Choosing to use Leaderboard — Data Storage & Leaderboard — CSCIF-19	View Leaderboard — Data Storage & Leaderboard — CSCIF-24	Upload Routes to Leaderboard — Data Storage & Leaderboard — CSCIF-20 (Story)
User not choosing to use leaderboard — Data Storage & Leaderboard — CSCIF-22	See Top Scores — Data Storage & Leaderboard — CSCIF-21	Save data from app to device and push to database — CSCIF-36
Publish Mobile Apps — Frontend (Mobile Apps) — CSCIF-27 JM	Seeing what the "top" routes are based off of ratings and compared to others — Data Storage & Leaderboard — CSCIF-28 CW	Research API — Full Stack — CSCIF-23 ES
Using saved routes deleting saved routes	Authentication	Basic Website Design — Frontend (Mobile Apps)

CSCI-3308-FA20-209-7
Classic software project

007-209-7
Board

- Backlog
- Active sprints
- Reports
- Issues
- Components
- Code
- Releases
- Project pages
- Add item
- Project settings

Projects / CSCI-3308-FA20-209-7 / 007-209-7

All sprints

Complete sprint

CW ES EP JM MM VV

Only My Issues · Recently Updated · All sprints ⌄

TO DO	IN PROGRESS	DONE
... Frontend (Mobile Apps) — CSCIF-27 JM	compared to others — Data Storage & Leaderboard — CSCIF-28 CW	... — CSCIF-23 ES
Using saved routes deleting saved routes — Route Management — CSCIF-30	Authentication — Full Stack — CSCIF-16 EP	Basic Website Design — Frontend (Mobile Apps) — CSCIF-33 ES
Basic functionality. — Routing Algorithm — CSCIF-6 VV		Basic iOS Design — Frontend (Mobile Apps) — CSCIF-31 JM
As a user, I want to be able to save a route so that if I like it, I can use it again. — Routing Algorithm — CSCIF-8		Log in page — Registration-Page — CSCIF-17 ES
The user will be able to access the app on their phone so that they can easily access and carry around a device that		Connect the data base with the front end (linking) — Route Management — CSCIF-34 MM