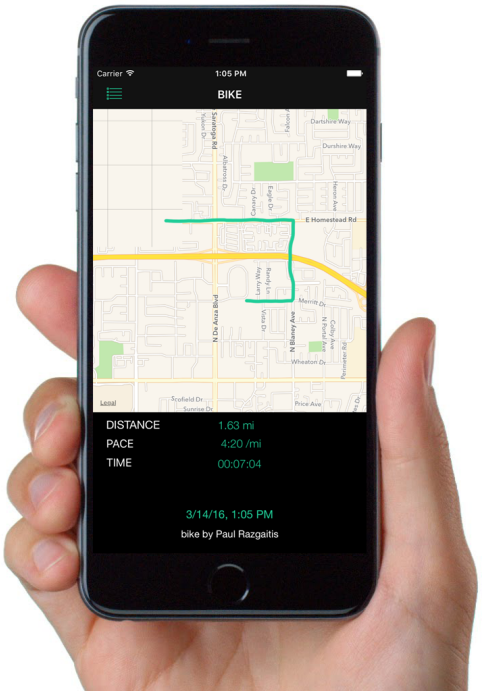
mpcs51030-2016-winter-project-prazgaitis



tri
triathlon training
with friends

Tri

Tri is a workout tracking app that helps triathletes track and log workouts.

Problem

It's hard to train for stuff, especially events like a marathon or triathlon. It's difficult to maintain a training schedule, to stay accountable, and to keep track of progress, let alone the actual training part.

Solution

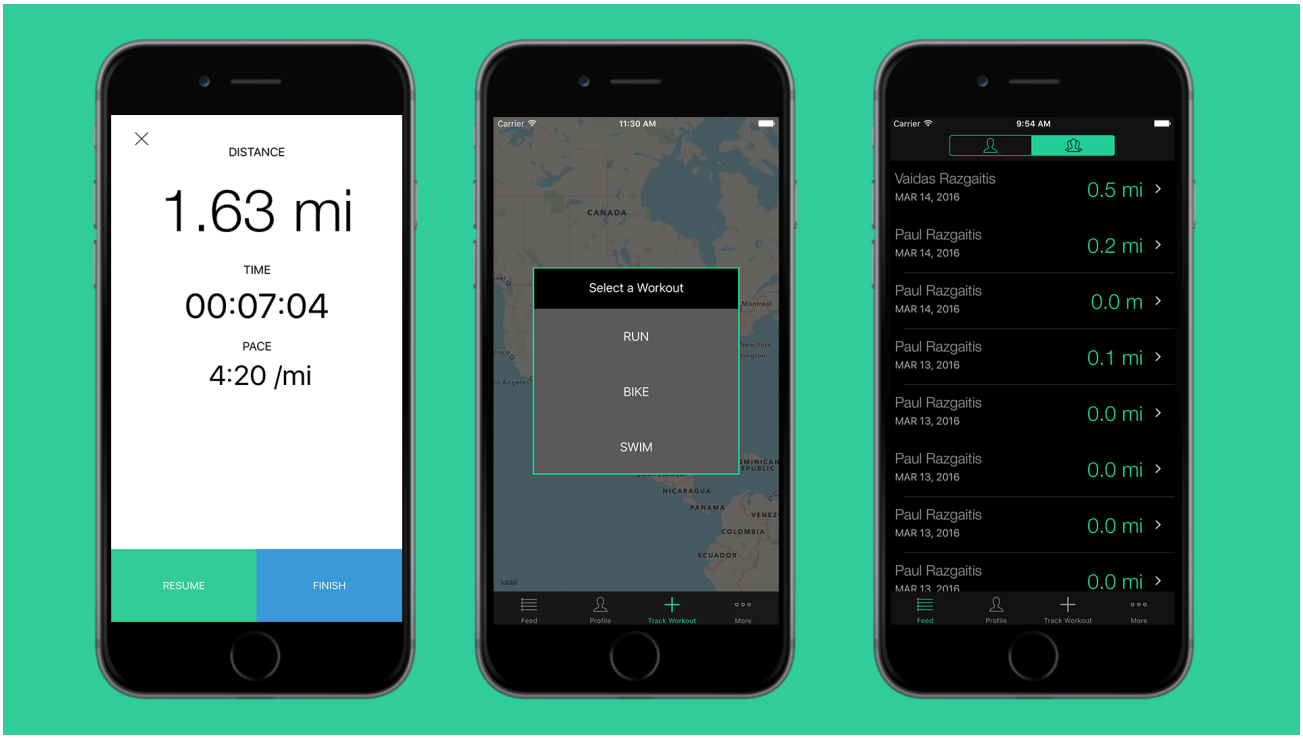
Conventional wisdom states that if you have a workout partner or a group, you are more likely to stick to your goals. Tri helps you stay accountable and accomplish your fitness goals by tracking your workouts.

App Store Landscape

There are several workout tracking apps in the App store. The closest competitor is Strava, a running and cycling GPS tracking app. Others include RunKeeper, Nike+, and MapMyRun.

Features

- Track run/bike data with GPS
- Input swim workout details
- view friends' workouts and how you stack up
- View stats and data about your own workouts

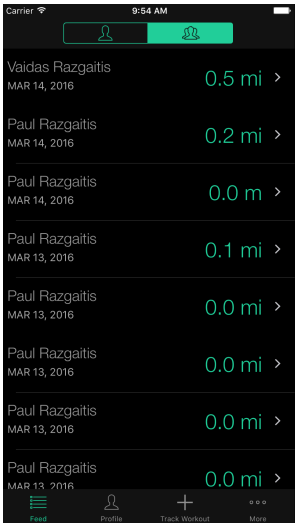


Technical Details

UI / Layout

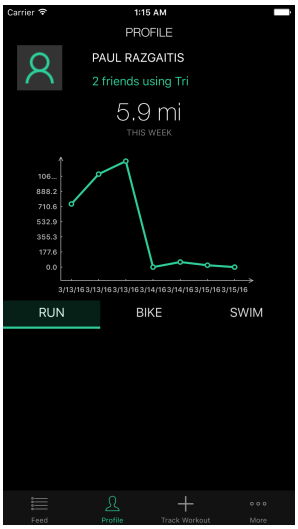
Tri uses a UITabBarController with 4 tabs.

Feed tab



The Feed tab lists all of a User's workouts. The UISegmentedControl toggles whether the feed is displaying a User's own workouts, or all of the workouts logged by the user AND the user's friends. Each cell contains the user's name, the date of the workout, and the distance of the workout. Tapping on the distance label shows the avg per-mile pace for that workout. Tapping on the cell causes the activity detail page to show up, which shows a Map with all of the workouts and other data.

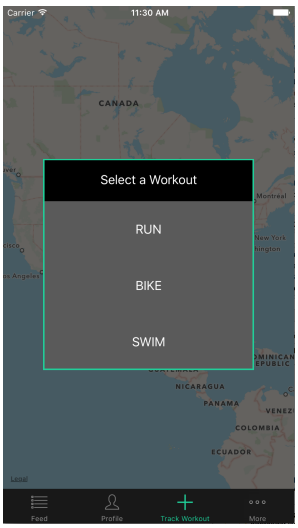
Profile Tab

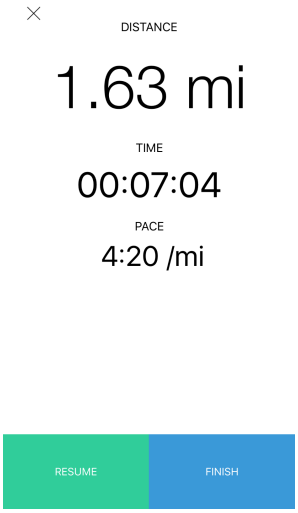


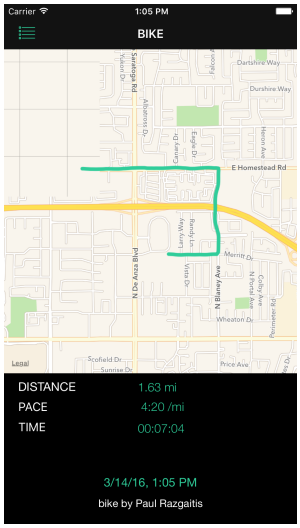
The profile tab shows the current user's workouts for the current week. The three buttons under the graph allow the user to toggle which activities (run/bike/swim) are displayed in the graph.

This view also shows how many friends of the user are also using Tri. Contact data is pulled from Cloudkit. There is no way to discover or add friends (yet), so the user is automatically "friends" with any of their contacts that also have the Tri app.

Track Workout Tab







This is where users go to start tracking a new workout. After selected either Run or Bike, the user is redirected to a the center view (above), which tracks the current workout with GPS. If a swim was selected, the user is shown a form to input distance, date, and duration.

Settings

Not much here yet, but a key feature is the ability to go to settings and easily give Tri access to location. If the user doesn't grant location access in the beginning, at least it's easy to turn it back on.

Libraries/Frameworks

PNChartSwift

The only third party framework was PNChartSwift, a library for drawing awesome graphs. You can see these graphs on the Profile tab.

CoreLocation

Tri uses CoreLocation to get the user's position with GPS. Location is retrieved with CLLocationManager every 1 second and added to an array of CLLocation objects. This array is saved to the Cloudkit database in an Activity object, which also contains duration (seconds), distance (meters), activityType ("run", "bike", or "swim"), the creator's name, and creator's unique ID.

MapKit

Maps are handled by Apple's own MapKit. Tri uses Mapkit to draw an MKPolyLine on top of a map to represent all GPS locations from one a run/bike session.

Cloudkit

Tri uses Cloudkit to get a user's info (first name, last name, id) from Apple, and then searches for friends who also use the app. There are two main Cloudkit queries (CKQuery) that retrieve user Activity data. The first one returns all activities where the creatorRecordUserId (the ID of the user who created the record) matches the current loggen in users id. Simply put, it returns all of the User's own activities.

The other returns all activities by timestamp. This will later be amended to return only Activity records that have a user ID that is also present in the current user's array friends' id's.