Project Name: [4. Bike route planner / tracker]

- “TrackStar”

Group Members: Group #24

- Chase Lampert - Route planner, set up Google places SDK and Google directions API
- Aminata Mamoud - Launch screen, app icon, music feature, helped with any display issues and the original google maps controller set up
- Susan Ngo - Project set up, set up Google Maps, implemented delete feature, implemented saving as .gpx file

Project Location:

- Main: https://github.com/susan-ngo/semesterProjectGroup24
- Secondary: https://gitlab.cs.umd.edu/cmsc436spring2019/cmsc436-sngo/tree/master/semesterProjectGroup24 (will need to update/push)

Description of Project:

User(s) have an option to view a map through Map Kit or Google Maps. Through the Map Kit selection, a user would have the ability to start tracking a trip when turning on the switch. When the switch is turned off, an alert will ask the user to provide a name in order to save the tracked route within core data as well as save as a .gpx file where the user desires from their device, otherwise there is an option to cancel which stops updating the location. If the user is still on the same view and turns the switch back on, the route will continue to track from where it left off if the user previously clicked ‘Cancel’. The user could return to the main navigation page to view saved rides. Clicking on one of the rides will retrieve the saved data from core data and display the saved ride. Swiping left on a ride deletes the ride.

NOTE:

- When git cloning/pulling from repository, may need to remove all images and mp3 files from XCode project and click and drag the physical files back into the project so the file path is correct (it appears that the file path gets messed up when cloning/pulling
- On the “Show Map” view, when turning on the switch, if nothing is being updated even after allowing the app to access one’s location, please try going to the top navigation, click “Debug”, go down and hover over “Location” and try to simulate another ride (such as City Bicycle Ride”) to get the simulator running.

Features of this Project:

1) Uses Google Maps SDK, Google Places API and Google Directions API to plan a bicycle route
2) Able to save a tracked route as .gpx file

3) Alternative (function) view would be using Map Kit to track a route
4) Able to play music while app runs in the background from a list of sample songs listed on the each page (Google maps and mapkit)

Evaluation of Goals Met / Goals Unmet:

Project is functional and uses Google Maps SDK, Google Places API, and Google directions API. Under the Bike Planner tab a user is brought to a new view controller where they should see their current location. This is indicated by both a marker that will stay static to show starting point as well as a blue dot which will move to coincide with the user's location. Below the Google Maps View are 3 buttons; one to choose a destination, and buttons to zoom in and out on your map. Once the set destination button is chosen the google places autocomplete widget pops up. This allows a user to start typing a place and the feature come up with relevant suggestions. Once the user makes a choice the Google Directions API is used to make an HTTP request to get bicycle directions from the current location of the user to the desired destinations. Then GMS polylines was used to paint the directions over the map to clearly display the route on the map.

A user is able to save a tracked route as a .gpx file, to later view and/or delete the route.Rather than saving as a .fit file, our group decided that a .gpx file would be more suitable as it is already built into XCode, as opposed to a .fit file which appears to require the use of FIT SDK. Due to the fact we also needed to add Google Maps SDK, we believed that looking into another new SDK would require looking further into and learning about a lot of new function calls. We felt it would be difficult to balance in the limited amount of time, as we would have to figure out how to implement certain functionalities with each separate SDK. Saving as a .gpx file will allow users to be able to manually upload their tracked route to other apps that support .gpx files (like Strava). Saving as a .gpx file would be more oriented towards users running the app from a desktop computer/laptop as opposed to users running the app on a mobile device.

A user is also able to listen to sample music while within the app without having their biking data lost when exiting the app. This is done using the AVAudioPlayer, which is a part of the AVFoundation Framework. Originally the music feature was to allow users to pick a song from their library within the app, but we ran into many technical and potential cost issues. Since these issues occurred we opted to just allow the user to have sample songs that could be ideal for a typical bike ride in any type of area. The user is able to select three different songs that are instrumentals and can play, pause, adjust volume, and exit the song freely.