

Prototype

Project Gambit (version 0.8.5)

March 28, 2016

App Overview

The current version 0.8.5 handles the biggest challenge in this application, the real time open socket location and heading updates by distance filter to another users (clients). The project is code-named Gambit here on out while in development.



Google APIs used

Many of the current features in the App use Google APIs to get time estimates and address locations. These APIs have limits (described here for [Places](#) & [Distance-Time Matrix](#)).

Real-Time Location updates

A lot of thought has gone into reverse engineering the Uber & Lyft apps in order for project Gambit to be performant, including battery consumption of the driver and client side.

The current product handles the states of both users, the client (the end user making a request), and the driver (who would be fulfilling a request).

Note: Since the version 0.8.5 does not handle Logging out of users, push notifications may be sent to devices that have been logged by a previous user; It is recommended to login to one user per device.

Cloud Server

The current cloud server setup for development would not handle many consecutive open sockets at a time.

Initial Login for Clients (*end users*) and Drivers



Client (*end-user*) Accounts:

- Username: **user**
- Password: **123456**

Drivers Accounts:

- Username: **hortons**
- Password: **123456**

- Username: **elias**
- Password: **123456**

- Username: **richard**
- Password: **123456**

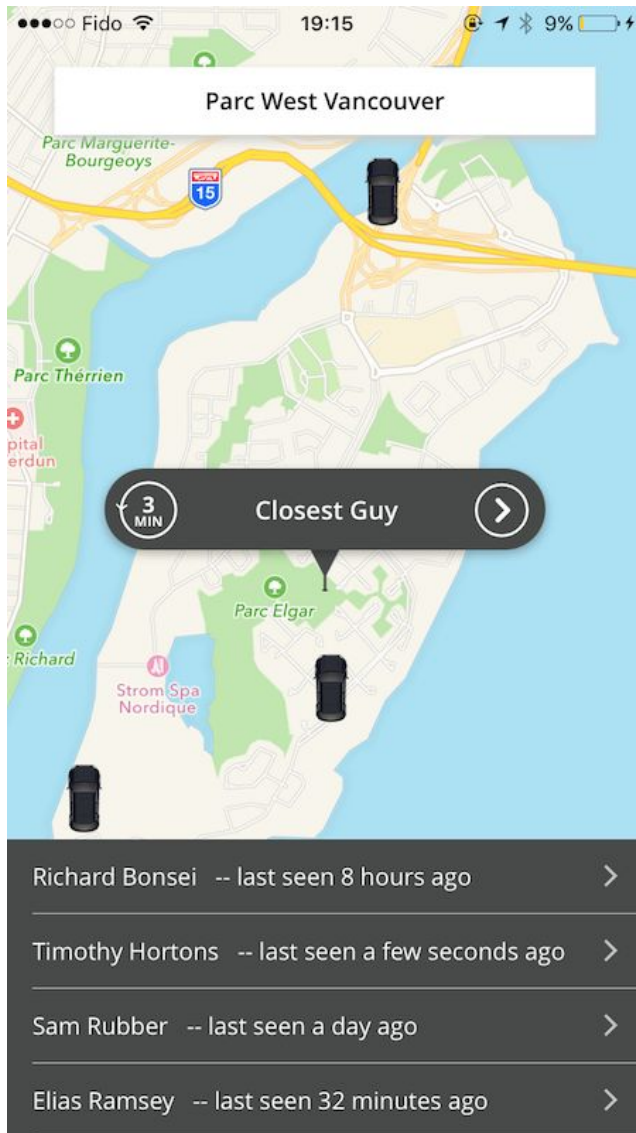
Accounts for both Drivers & Clients currently are only created on request.

When the username and password have been entered, simply tap the **return** button on the bottom keyboard. By default in version 0.8.5, the username is **user** and password is **123456**.

On Login, entering the account usernames will trigger the state of the application for user or

driver functionalities. The subsections Client & Driver below will describe the features of each user.

Client Initial View



By default, the map centers on the user's current location.

The top Text Input updates as user Pan the map to select their pickup location. As the user Pans, the bottom list of users (if any) will go out of the way (effect similar to Uber).

The address in the top Input updates to the pin's currently selected location. The client may at any time in the initial view, to tap on the top address input bar in order to type in their address, an auto complete address feature will be displayed in a list in order to select the address as the user types. Google Address Autocomplete and Reverse Geocode APIs are used here.

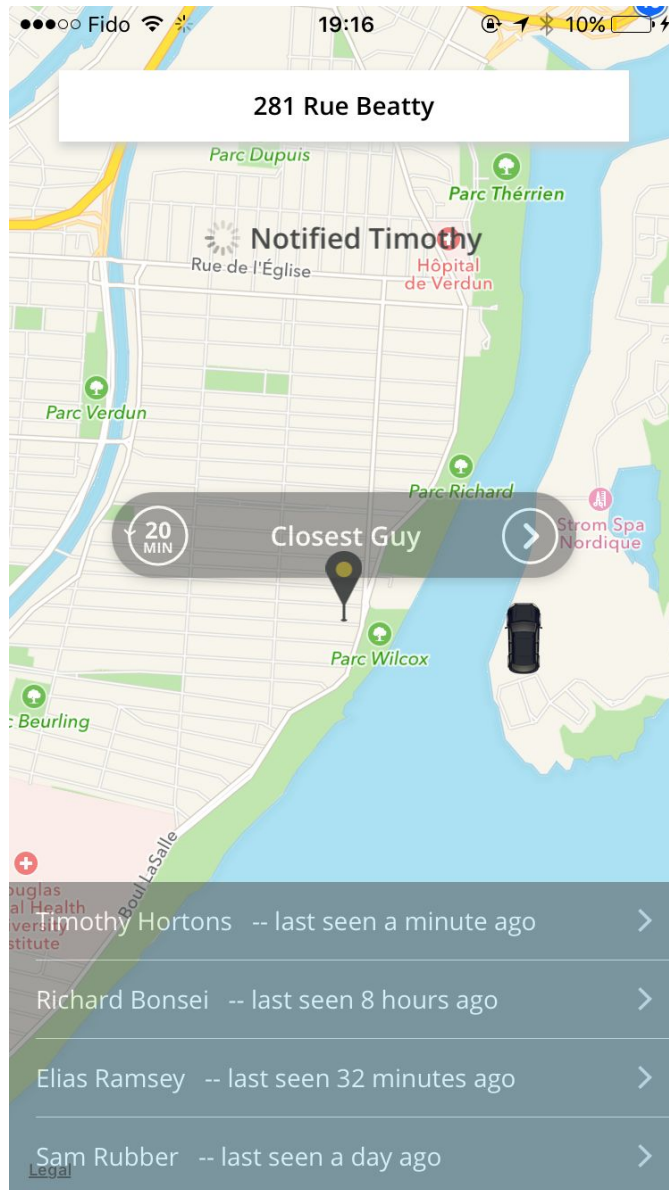
The bottom 4 rows of drivers seen by the current client, are the 4 Closest Drivers **within a radius of 5 Km (3.1 miles)**, and are sorted by closest driver on top. *The last seen time location update from the drivers is currently used to development reasons.*

Tapping on the Driver's row currently doesn't do anything at the moment; Tapping again, will return you to the list of users.

The Map's **Info Bar Button** (which is the bar shown with: "3 MIN Closest Guy >") once tapped on automatically creates a Request to the **closest** (top in the list) driver. The page below describes what happens on the client (end-user side).

We utilize Google's Distance Matrix API to get accurate Estimated Time of Arrival with current Traffic conditions, displayed here as "3 Minutes" in the Info Bar.

Client Request View

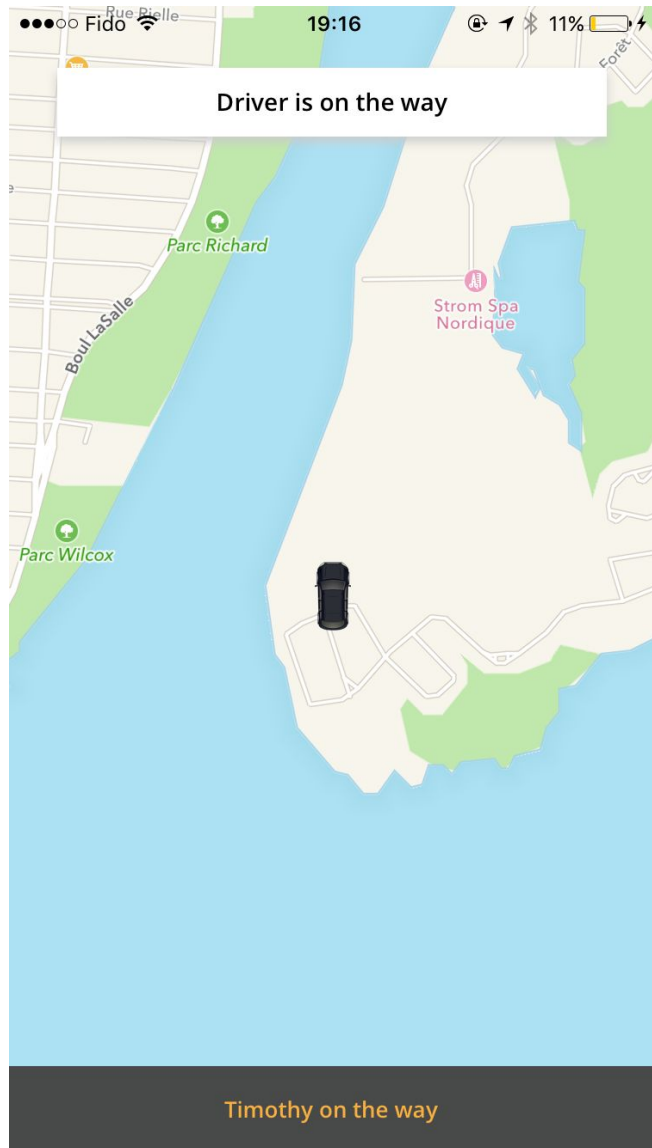


In this time-lapsed state, making **new** requests by this client has been disabled.

If no driver accepts the request within **20 seconds**, the request is cancelled, and the client will have the ability to make new requests after that time lapse.

The client is updated on the request with a loader and brief description as feedback.

Client View of Driver's Acceptance.

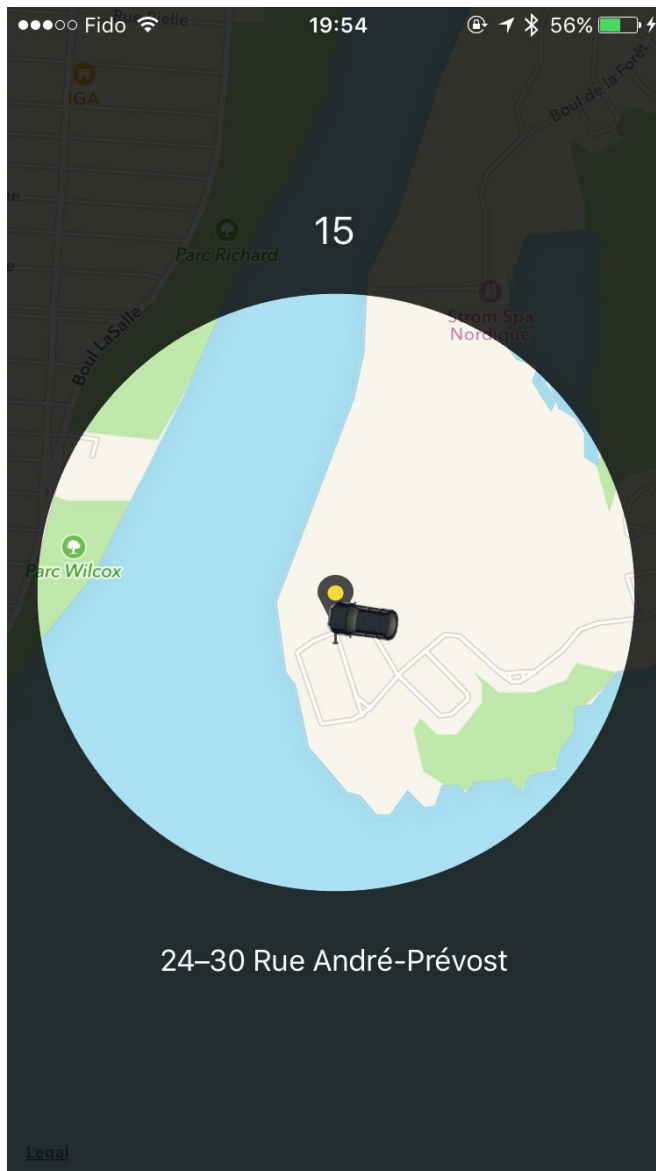


Once the Driver accepts the Request, the open socket is triggered with a 'beep' sound, and the client views the location of the driver in real time, in addition to the heading (orientation), as the driver starts driving. The location updates every 15 to 20 meters (and not based on time for very good reasons). If the driver stops at a red light for example, the GPS and the open socket is not being consumed, optimizing performance of data transfers and more importantly, on the battery consumption.

This view shows in real time the location of the Driver.

The bottom & top bar describes the state of the client's view as the driver is on the way.

Driver's View of a Request - 15 second countdown.



devices.

Once a Driver has logged in with the credentials of a driver, they have to go online in order to start accepting requests from clients.

The Driver's view of the App prevents the App and the device from sleeping and from automatically shutting down.

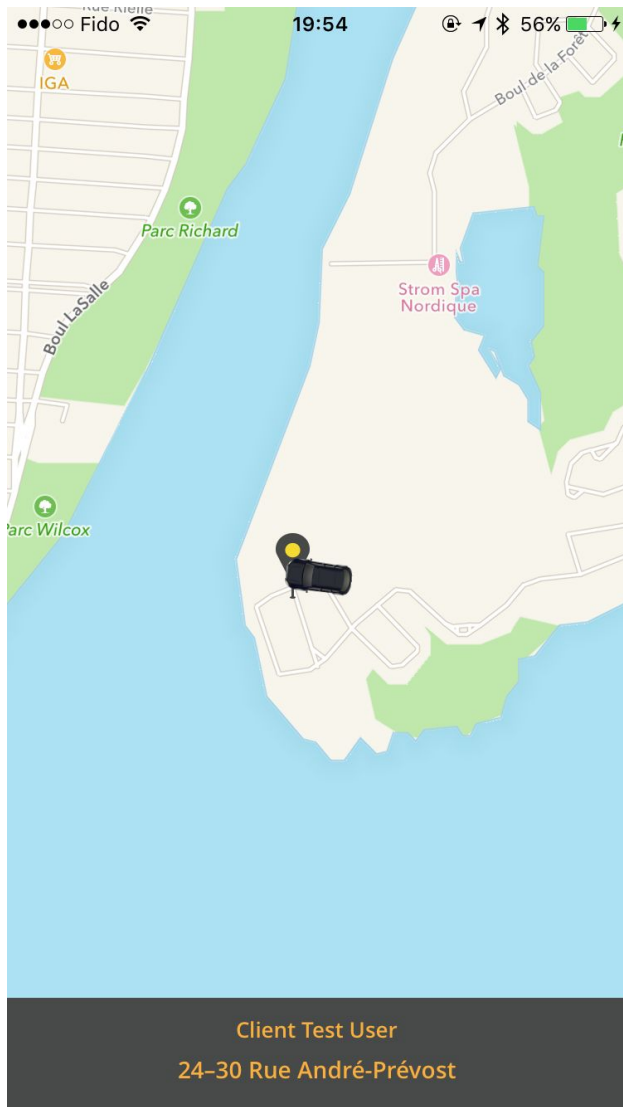
The snapshot on the left is the view the Driver receives when a Request is made. It indicates the location of where the client pickup address is and a pin marker on the map, with a 15 second countdown in order for the Driver to accept.

The driver accepts the request, simply by tapping anywhere on the screen.

NOTE: Currently in version 0.8.5, the driver may receive multiple requests. However, no implementation has been done when a driver receives another request while 'On Duty'.

This view of the request seen by the driver, is optimized for iPhone 6 and iPhone 6S

Driver's View when On Duty (accepted a request).



When the Driver accepts the request by tapping the view above, the location of the driver is updated as they drive to the client in real time with a banner at the bottom with the name of the User (client) with the address to go to.

In addition, a marker on the map is kept to show the pick up location.

When the driver is 250 meters away from the client, an alert is shown to the driver that they are close to arriving the pickup destination.