

Go-Cab

A Cross-Platform Mobile App for Ordering a Taxi

<http://www.go-cab.com>

by Ryan Sundberg
June 18, 2012

Gene Fisher, Advisor

Abstract

Go-Cab is a cross-platform mobile application improving the taxi cab ordering process for both the customer and the dispatcher. Especially in foreign environments, taxi customers suffer the costs of a lack of information. Taxi dispatchers suffer the costs of competition, especially against established names in their area.

Go-Cab addresses problems with ordering a cab from the customer's perspective, and in doing so creates a better experience for both the rider and provider.

1 Introduction

Go-Cab is a cross-platform mobile app for ordering a taxi cab to your current location. It combines the utility of GPS with features of the mobile web to streamline the cab ordering process without the necessity of speaking with a dispatch operator over the phone. Its benefits reach to both the cab customer and company dispatchers. It can make the process more convenient for customers and profitable for companies.

1.1 Problem Description

There are various inconveniences and difficulties for both the customer and the dispatcher when ordering a taxi cab. In environments where reliance on the chance of a drive-by pickup is impractical, frustrations increase. Inconveniences for the customer include:

1. Searching for the phone number of a local cab company
2. Choosing among companies with unknown reputations
3. Choosing among companies with unknown pricing
4. Estimating the cost of a ride
5. Estimating the time of arrival and traffic conditions
6. Determining one's current location or address at an unfamiliar place
7. Having to excuse oneself from conversation in order to call a dispatcher

Difficulties for the cab company include:

1. Establishing a reputation in their area
2. Communication difficulties over the phone
3. Attracting more customers
4. Handling peak volumes of calls and customers

1.2 Solution Overview

Go-Cab addresses the problems of ordering a taxi cab primarily from the customer point of view, and incidentally benefits the cab companies by better serving their customers. As smart phones become increasingly common, the taxi industry will be well served by making use of their potential.

The taxi customer's problems are addressed by Go-Cab as follows:

1. Searching for the phone number of a local cab company
Go-Cab uses GPS to determine the user's current location. The user can also specify a location other than their current position. It then searches a database of cab companies who have signed-on with Go-Cab for those companies which cover the pick-up area. The user is then presented with the list of nearby companies after they submit their pick-up information. In addition, the Go-Cab app shows Google Places API results for local cab companies, so the user has alternative options available. This is useful especially in the case that no companies have signed on with Go-Cab in the user's area.
2. Choosing among companies with unknown reputations
When Go-Cab presents the user with the list of nearby companies, each company has a 5-star rating meter below their name showing the average rating from other go-cab users, and the number of total ratings. This information helps the user make an educated

decision about which company to choose. Written reviews may be added in a future release.

3. Choosing among companies with unknown pricing

When a cab company signs on with Go-Cab, they provide a formula for their driving costs. With this information, Go-Cab can provide customers with an idea of the relative pricing between competing cab companies.

4. Estimating the cost of a ride

If an optional destination address is provided, the list of cab companies also includes a total fare estimate for the distance to be traveled. The fare estimates are created by a formula provided by the cab companies in the form: Estimate: function(Distance)

In a future release, estimated traffic time will be included, so price estimate accuracy can be increased via the formula: Estimate: function(Distance, Time)

5. Estimating the time of arrival and traffic conditions

Traffic estimates are not currently performed by the app, so this is to be addressed in a future release. However Google does have information about current traffic conditions, in addition to time estimates without respect to traffic, so it is possible to add this feature.

6. Determining one's current location or address at an unfamiliar place

At the initial screen to order a cab, the user's pick-up address is pre-populated with their current address according to the available GPS data. They are also shown a map pinpointing their location.

7. Having to excuse oneself from conversation in order to call a dispatcher

Go-Cab allows the user to include a typed message with their order, so they can specify any additional instructions for the dispatcher or driver. They can choose between a text and a call for their preferred contact method, in case the dispatcher needs to re-initiate contact with the user.

The taxi provider concerns are addressed by the app as follows:

1. Establishing a reputation in their area

Go-Cab includes a rating system for cab companies. After a user has ordered a ride, they can rate their experience from 1 to 5 stars, depending on the timeliness of their service, their personal experience with the driver, etc. Ratings are displayed in the listing of providers to the user, so new cab companies can quickly establish a good reputation in their area.

2. Communication difficulties over the phone

Go-Cab eliminates the necessity of speaking with the dispatcher over the phone by including all of the necessary information in the order form. The customer indicates if their preferred mode of communication is text or a call in case the dispatcher needs to re-initiate contact with them.

3. Attracting more customers

Along with the rating system, by simply being in the Go-Cab listing a cab company

increases their exposure to the market of Go-Cab users. A cab company can secure all Go-Cab users as customers if their competition has not signed on.

4. Handling peak volumes of calls/customers

Since Go-Cab does not require a phone call with a dispatcher, lost customers due to busy signals and placement on hold are a non-issue for Go-Cab users. Additionally, it frees up phone time for those other customers who do call in.

1.3 Focus and Scope

The focus of the Go-Cab project is to develop a prototype app that can be used to secure contracts with a set of pilot cab companies, or funding or sale to interested investors. Go-Cab is focused on solving problems from the customer side, and leaves dispatchers to keep their current systems and processes for handling and routing cab orders. That is, Go-Cab provides a minimal software interface for dispatchers to handle order requests.

Although the product is not ready for release to market, it establishes a foundation for gathering feedback and evaluations from users.

Known issues and lacking features include:

- Better aesthetics and theming
- Glitches with address auto-completion opening Google Maps
- Glitches with reverse-geocoding (near, but incorrect addresses are initially loaded sometimes)
- Glitches with the map loading – some areas of the map image fail to load
- Lacking traffic detection for time estimates
- Route distances are spherical path distances, not street travel distances
- The app should have shortcuts to recently used locations
- SMS notifications

2 *Application Control Flow*

The following state chart shows the cab ordering process from the user's perspective:

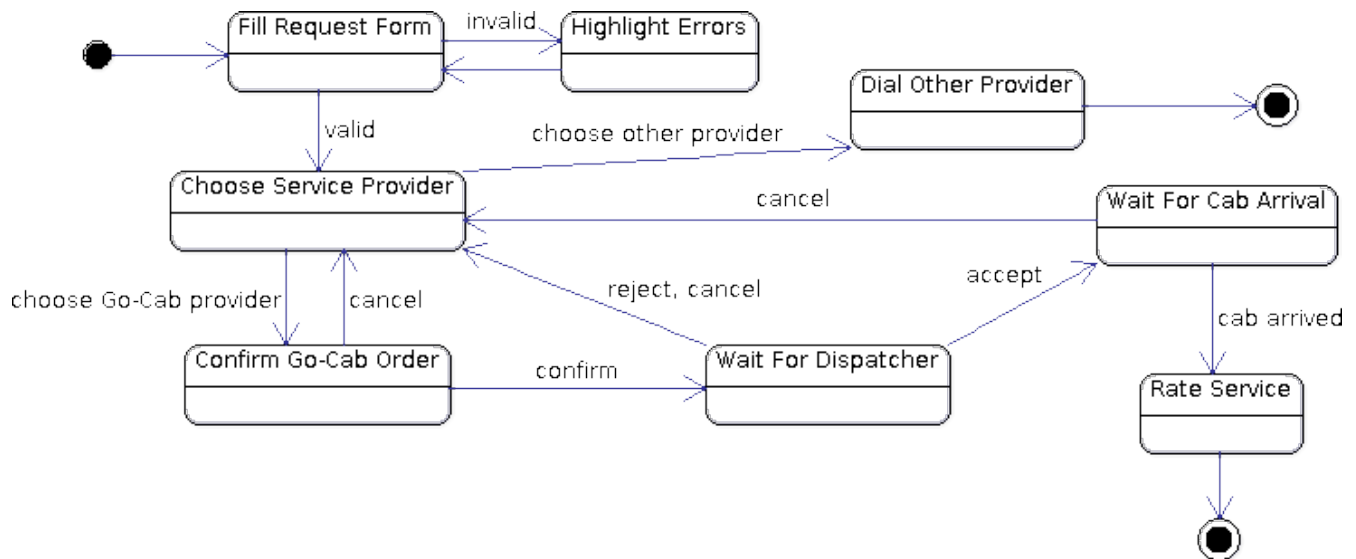


Figure 1: State Chart

3 System Design

Go-Cab is a web application using HTML, CSS, and JavaScript to render the app on the client side. PHP and MySQL are used server-side to process the control flow of cab requests and orders.

3.1 Go-Cab Server Class Diagram

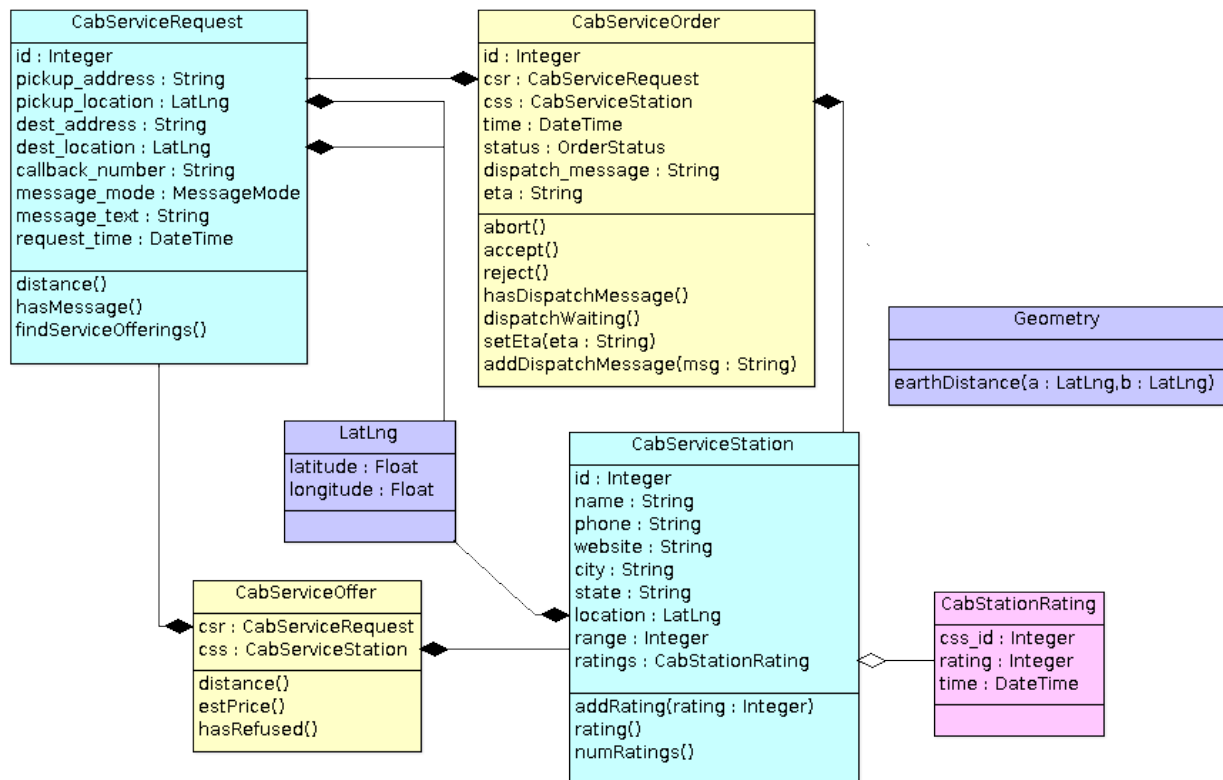


Figure 2: Class Diagram

3.2 Cross-Platform Compatibility

Go-Cab is packaged as a native app in order to provide the quickest access for the user and to leverage features provided by the various mobile operating systems. It runs inside the PhoneGap framework, essentially a chromeless browser, which is now being absorbed by the Apache Software Foundation as “Apache Cordova”¹. PhoneGap/Cordova “is a platform for building native mobile applications using HTML, CSS, and JavaScript.”² PhoneGap enables Go-Cab to be packaged on any mobile OS with zero to minimal code changes. The current prototype version of Go-Cab is compiled for the Android 2.3 API.

4 GPS Use

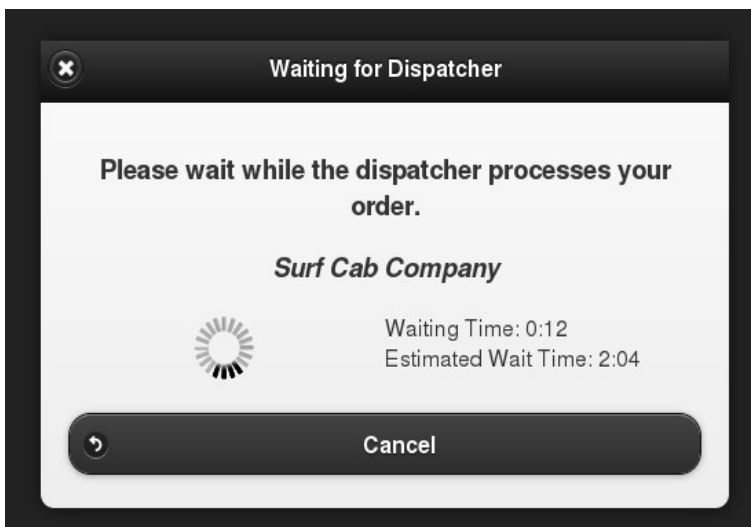
With the PhoneGap framework, we are able to install Go-Cab with GPS permission granted by the operating system, guaranteeing the availability of GPS location. PhoneGap implements the W3C Geolocation API Specification³ which is a JavaScript API to obtain the device's location.

Go-Cab uses the Google Maps API V3 in concert with the W3C API to perform a function called “reverse geocoding.” Reverse geocoding returns a street address from a geolocation (a latitude and longitude pair). Thus Go-Cab is able to prepare the user's current approximate street address as their pick-up address automatically when starting the cab request form.

GPS is also involved in the auto-completion of street addresses. The Google Maps API includes functionality to implement an auto-completing search box for street addresses. These auto-completed results are biased towards local results, so a search for “1 Main street” will auto-complete to the closest Main street, instead of one across the country. When an address is selected, Go-Cab uses the Google Maps API to pinpoint the location on a street map.

5 Real-Time Synchronization with AJAX

Once the user has confirmed their order with a cab company, they enter into a state where they are waiting for an acknowledgment of their order from the cab dispatcher.



The response is time sensitive, since the user is more likely to grow impatient and cancel their request with each additional second of delay.

One solution is to poll the server every n milliseconds, checking to see if the status of the order has changed from a wait state to an acceptance or refusal. This solution requires opening many new connections to the server during the waiting time, which places a heavy burden on the server and also on the mobile device's network use.

Another solution is called AJAX long polling, which simulates a push from the

Figure 3: *Waiting for Dispatcher* screenshot
server as opposed to polling from the server. AJAX long polling starts with a single HTTP request from

the client, and then the server loops, or hangs, until the status of the order changes. This reduces the load to one connection which is kept open until there is some significant data to return, instead of repeatedly polling the server to see if there is a change.

With AJAX long polling, the Go-Cab user is able to recognize an order's acceptance or refusal within one second of the dispatcher's input.

6 Related Work

Mobile ride ordering is an emerging market, and there are two established competitors in the market so far:

6.1 Uber

<http://www.uber.com/>

Uber is a black-car driver service. They have a mobile app for iPhone and Android which you can order a pick-up to your current location with, and it requires paying with your credit card on file. Uber contracts for their own drivers, and will pick customers up within a longer range than a taxi cab would. Black car services like Uber are not regulated in the same way as public taxi companies are, so they have more discretion with their business practices. Uber's market demographic overlaps with the taxi market, so they would be in competition with Go-Cab.

6.2 Cabulous

<http://www.cabulous.com>

Cabulous is a taxi-ordering app very similar to Go-Cab. Cabulous is a more comprehensive solution, requiring the dispatchers to use their dispatching software and drivers to carry mobile devices. Cabulous also enables credit card payments via their mobile app. Since Cabulous drivers are required to have a mobile device, the app can provide automatic estimated arrival times and real time information on the location of taxi cabs. Go-Cab differs from Cabulous in that it does not require such significant changes to a cab company's business practices, thus reducing costs, and it allows users to order cabs from a larger range of companies by integrating the Google Places API. Cabulous is the most significant known potential competitor for Go-Cab.

GoCab.com

<http://www.gocab.com>

GoCab.com is a domain name showing intent to develop a mobile iPhone app to hail a taxi, but there have not been any apparent developments nor is there a development schedule. The name is not trademarked. The Go-Cab application described by this paper uses the domain name go-cab.com to differentiate it.

7 Conclusions and Future Work

Go-Cab is ready to be demonstrated to user focus groups for usability analysis and testing. It should

simultaneously be presented to various cab companies (starting in smaller cities) to determine their interest. Since Go-Cab is still a useful application even when there are no contracted cab companies in an area (because of the Google Places search results), it can be marketed even when there are no cab companies contracting with Go-Cab in an area. After input from user focus groups and interested cab companies, Go-Cab should be piloted in cooperation with those companies at a discounted cost. With some targeted local marketing, Go-Cab can be established in the marketplace as a viable service and a cheaper alternative to Cabulous for cab companies.

8 Interesting Code Sections

8.1 Distance between points on a sphere

Go-Cab uses this formula, published on MathForum.org⁴, to calculate trip distances. It calculates the distance between points on a sphere for angles measured in degrees:

$$\delta = R * \arccos(\sin(\phi_1) * \sin(\phi_2) + \cos(\theta_1 - \theta_2) * \cos(\phi_1) * \cos(\phi_2)) * \pi / 180$$

The following PHP function implements the calculation for angles in radians:

```
function earthDistance(LatLng $a, LatLng $b) {
    $c = (sin($a->rLat) * sin($b->rLat))
        + (cos($a->rLng - $b->rLng) * cos($a->rLat) * cos($b->rLat));
    return RADIUS_EARTH * acos($c);
}
```

8.2 Adjacent points within a certain distance

Finding the set of cab service stations near the pick-up address reduces to the problem of determining the set of points S within a certain distance to another point p .

$S = \{ s \mid \text{earthDistance}(s, p) \leq r(s) \}$, where s is a cab station and $r(s)$ is the service radius of s

With a small set of stations, we can simply compute the distance between the pick-up address and all known stations to determine S . This is the current implementation used by Go-Cab:

"On input $\langle A, p \rangle$ where A is the set of all station points and p is a pick-up point:

1. Compute S , where
 $S = \{ s \mid s \in A \text{ and } \text{earthDistance}(s, p) \leq r(s) \}$

If the set of stations gets sufficiently large, this function may become computationally expensive and we can devise a more efficient algorithm to determine S . We use an Azimuthal Equidistant map projection, maintaining true distance scale from the center of the map⁵:

"On input $\langle A, p \rangle$ where A is the set of all station points and p is a pick-up point:

1. Precompute p , the distance from the origin, for all points $s \in A$ projected onto an Azimuthal Equidistant map
2. Project point p onto the Azimuthal Equidistant map
3. Compute S , where
 $S = \{ s \mid s \in A \text{ and } (|p(s) - p(p)| \leq r(s)) \cap (\text{earthDistance}(s, p) \leq r(s)) \}$

Works Cited

- 1 <http://phonegap.com/>
- 2 <http://incubator.apache.org/cordova/>
- 3 <http://dev.w3.org/geo/api/spec-source.html>
- 4 <http://mathforum.org/library/drmath/view/54680.html>
- 5 <http://books.google.com/books?id=nPdOAAAAMAAJ&ots=f3tsjnUnlX&dq=azimuthal%20equidistant%20map%20projection&lr&pg=PA193#v=onepage&q=azimuthal%20equidistant%20map%20projection&f=false>