

CPSC 490 Project Proposal

Christopher Chute

September 22, 2016

Ride Hailing for Yale's Special Services Van

Problem

Yale's handicap shuttle service, also known as the Special Services Van, is largely obsolete. If you want to schedule a ride, you call the Special Services Van Dispatch at 203-432-2788, and schedule a ride with a dispatcher. The process takes an average of about three minutes, during which the dispatcher negotiates a pick-up time. There are often conflicts, especially in the morning, and the dispatcher must list the options of times to be picked up. Moreover, drivers often go to the wrong spot (they are only given an address), and there is no functionality to take your course schedule into account. The time wasted dealing with this system is especially precious, since handicapped students are already more time-constrained in general.

Solution

My goal is to build a modern system to dispatch handicapped rides at Yale. The deliverables will be an iOS application and (secondarily and with limited functionality) a web interface for requesting shuttle rides. The app must address the issues laid out above to be effective, and I try work to make sure it actually gets adopted by Yale Special Services.

Specification of the App

1. It must verify via NetID that a person requesting a ride is a Yale student, and is on the whitelist for requesting the Special Services Van.
2. It needs to provide fine-grained GPS-based routing to a person's location, so that the van drivers can be updated more precisely about where the requester is located (e.g., which side of the street, and where along the street).
3. It needs to allow recurring scheduling of a ride. In particular, the app should be able to read in a course schedule, ask how early a student wants to arrive and how long afterwards they would like to be picked up. It should then schedule rides every week at those times, and not on days when classes do not meet.

4. It should take advantage of push notifications to tell a requester when their ride is coming.
5. It must provide a simple interface for requesting one-off rides, and it should show which time slots are available to be requested.
6. It must be easy for the drivers to use, and should route them directly to students. It should not schedule rides that cannot be completed with time to travel between them, and at the very least it must notify a requester when there will be a delay to pick them up.

Itemized Plan

1. *Preliminary research phase.* First I will gather information on precisely defining the problem and possible solutions.
 - (i) Gather statistics on Yale students and what they would use to request rides. (Done).
 - (ii) Gather information on what students would like in a ride-requesting interface. (Done).
 - (iii) Start learning the tools needed to build this app. (80% done).
 - (iv) Get information from the Special Services department on ride usage (*e.g.*, number of registered riders, number of calls per day, current system). This will be used for evaluating the system later.
2. *Design planning.* In this stage I'll lay out the design specification for the app.
 - (i) The data model and integration with a central server or iCloud will come first. Plan how to model course schedules, rides, riders, drivers all in one. How to keep this information synced and reliable.
 - (ii) The user interface design will come second. This includes the storyboard layout and the controls that will be used at each step in scheduling a ride. Another UI is needed for drivers.
 - (iii) The superficial design and branding will come last. Items include color scheme, headers and footers, affiliation with other Yale apps.
3. *Implementation.* I will need to start implementing the app by the start of October. I plan to have the following checkpoints:

- (i) *Skeleton Storyboard*. Storyboard in place with all segues, so UI design is ready to house the logic. Both drivers and riders.
 - (ii) *Local Scheduling*. Data model in place with rides able to be scheduled and stored persistently, but only locally. Testing will be done here to ensure the data model works.
 - (iii) *Request Generator-Only*. All functionality in place only for riders, so it will just generate requests that can be sent to the current dispatching system.
 - (iv) *UI Polish and Testing*. At this point I will try to get the UI in a polished state, and I will put it on multiple devices for people to test.
 - (v) *Networking*. While testing is underway, I will work on getting the service networked. This includes getting the driver-side app to read the models and get GPS coordinates. It will store the schedules in iCloud or some other centralized data store service.
4. *Rolling Out*. At this point I will hopefully have gotten the Special Services department on board, and I'll work with them on trying to get the app rolled out. There will most likely be two stages:
- (i) *Experimental*. The usual dispatch system will remain in use, and the dispatcher will also schedule rides on the app. Riders will have both options, and their requests in the app will be duplicated in the other system. Drivers will have the option of using either system, and bugs can be reported.
 - (ii) *Adoption*. If all goes well, the project app will be the default for getting a Special Services ride, and riders/drivers will default to this option.

Deliverables

1. Design Documents:

- (a) Collected statistics
- (b) Storyboard and UI
- (c) Specification of data model
- (d) Tutorial or user's manual

2. Finished Products:

- (a) iOS application written to the specs above.