Randy Fitzmorris
Senior Design
Wednesday November 11, 2015

4Ride Mobile: Design Document

Project Objectives

The 4Ride Mobile project aims to create a central-dispatch framework that computes optimal vehicle itineraries for 4Ride drivers. This framework will be deployed over a web service such that mobile applications for students and drivers can connect to its data feeds. A student mobile application will be able to view all 4Ride vehicle locations, request a pickup, and cancel a pickup. A driver mobile application will be pushed optimal itineraries of pickup and drop-off locations in the order they should be serviced. The project aims to develop both these applications in iOS for completeness, but leaves the web-based dispatch framework open to any application that follows its protocol.

End Users

User Description

1) GW Student: This user will be interacting with the 4Ride Mobile: Student application, using the application to schedule a 4Ride pickup and destination location.
2) 4Ride Driver: This user will be interacting with the 4Ride Mobile: Driver application, using the application to view the vehicle's updated service itinerary.

Use Cases:

1) GW Student:
    a. Student will need to own an iPhone running at least iOS 8.0 and initially download the application from the Apple App Store.
    b. Student will initially input identifying credentials in the settings view.
    c. When the 4Ride service is required, a student will simply open the installed application for immediate use.
2) Driver:
    a. Driver will need to own an iPhone running at least iOS 8.0 and initially download the application from the Apple App Store.
    b. Driver will initially input identifying credentials in the settings view.
    c. Driver is familiar with referencing routing information on the itinerary view rather than calling a central dispatcher.
    d. When the driver's vehicle is in service, he/she will leave the application open and running to retrieve real-time itinerary information.

Specific Work Flow

1) GW Student:
    a. Student uses app to access 4Ride scheduling service.

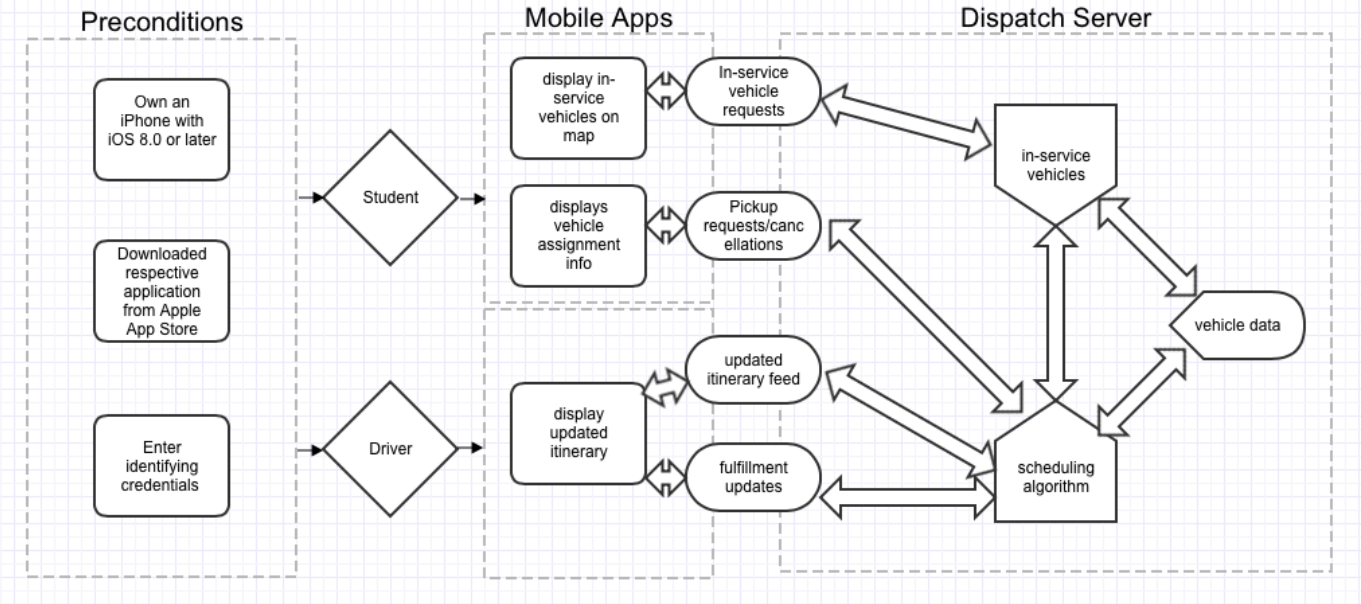
b. The application will query the dispatch server for in-service vehicles and display their current locations on the map in the user's view.
c. Student issues request for pickup including a destination.
d. The application will issue the request to the dispatch server. From there, the server will calculate which in-service vehicle suffers the least time-impact from the request and assigns the student. The vehicle assignment information is then pushed back to the student view.

2) 4Ride Driver
    a. Driver starts the app once his vehicle is in service.
    b. The application issues a signal to the dispatch server that the vehicle is now in service, making the vehicle eligible for future request assignments.
    c. The dispatch server assigns student requests to the driver's vehicle, calculates the optimal itinerary, and pushes the itinerary to the driver view.
    d. The driver services the requests in the order given from the itinerary.
    e. The app will send a fulfillment verification to the dispatch server when the driver arrives, and the location will be dropped from the itinerary. An updated itinerary will be issued to the driver's application.
    f. Further requests and cancellation are automatically pushed and updated on the driver's application.

Functional Components

1) GW Student:
    a. 4Ride Mobile: Student application: This app will display in-service vehicles on a central map and allow a user to request and cancel service requests, manually entering a destination point. When a vehicle is assigned to the student, he or she will be able to view identifying information about the driver and his/her vehicle.
    b. Dispatch Server: This server will allow the student application to make web service calls with requests, student location and destination points, and cancellations. It will also send in-service vehicle data in real time as well as vehicle assignments and corresponding information.
2) 4Ride Driver
    a. 4Ride Mobile: Driver: This app will display an optimal itinerary of pickup and drop-off service assignments. The driver needs only to leave the application running during his or her time in service. Identifying information like distance, address, and student name will be given for each itinerary item.
    b. Dispatch Server: This server, which is the same as for the student application, will issue updated itineraries for the driver's vehicle.

Control Flow Graph



Functional Requirements

1) Collect vehicle location data and display in student app
2) Collect driver credentials data
3) Collect student credentials data
4) Send pickup and drop-off requests/cancellations to server from student app
5) Send fulfillment updates to server from driver app
6) Server should assign pending student requests to vehicles (and remove fulfilled requests) based on optimal allocation, calculate resulting optimal tour route for each vehicle, and push updated assignments to student/drivers

Non-Functional Requirements

1) Both mobile applications will be intuitive
2) Unit tests will exist for all functional components
3) Dispatch server framework should be extendible
    a. Easily accommodate any additional 4Ride service locations in future
    b. Support any mobile development or web-enabled platform\
    c. Additional weighting factors should be easily added to optimal-scheduling algorithm