Christopher Guzowski 5/24/16

**UBER**

**Company Description**

Uber; the most recent, trending and escalating cab service is a quick and reliable transportation system. Their growth spread rapidly as they began in 2009 and currently today serve over 70 countries and more than 400 cities worldwide. If we were able to build this new advanced system of operations, the new information system would gradually; but definitely, make way to each location benefitting not only the people we serve, but also our company. As we generate 20% of each transaction an Uber driver makes as a return for being registered in our system and employed under our name. For software transactions, firstly we obviously operate using GPS software to lace drivers to nearby requesting clients, after the client chooses a driver through an app or the web we signal the driver to the passengers location. After arriving at the desired destination all information such as time, place and distance traveled is stored as a record and if the client does not offer feedback, then the cycle finishes. One of Ubers motto is “Tap the app, get a ride” and the most significant difference to most competitors is that the car is usually black and luxuriously comfortable on the inside. They are also reliable “Ready anywhere, anytime” for instance in early morning commute or after late night drinks. A flexible image they represent is “low-cost to luxury” where they stated “You can always request everyday cars at everyday prices. But sometimes you need a bit more space. Or you want to go big on style. With Uber, the choice is yours.” Lastly, what makes this company great is they say the driver is a person just like you, going your way (they are free (no boss no schedule) and earn a portion of the money you give them) where they say “What makes the Uber experience truly great are the people behind the wheel. They are mothers and fathers. Students and teachers. Veterans. Neighbors. Friends. Our partners drive their own cars—on their own schedule—in cities big and small. Which is why more than one million people worldwide have signed up to drive.” The company itself estimates to be worth $62.5 billion by the end of 2015. You can imagine the size and the rate this company is escalating at which answers; yes, they need a new info system as they are currently working on a project that takes multiple customers to the same location in just one vehicle (upmost efficiency).

**Detail Description**

How big: Multinational

Yearly Revenue: Keeps 20% of transactions, 50,000,000$

Locations: 70 countries 400 cities

What do they sell: Transportation Services

How they make money: Fulfilling the request of a customer in need of a vehicle.

How many people use uber: over 50 million -/+ (8 mil – 7/9/14)

How many people drive for uber: 300,000 -/+ (160k – 1/22/15)

Total number of uber trips taken: 1 billion rides (12/30/15)

Average # of new drivers for uber monthly: 50,000 drivers (12/17/14)

**Transactions**

Per hour: 4,761 (114,285 / 24 hours in a day)

Per day: 114,285 (800k / 7 days a week)

Per week: 1.1 million but 800k are completed (generating $20 million)

Per month: 3.2M (800kx4 weeks)

Per year:38.4M (3.2Mx12 months)

**Transactions (New york only)**

Per hour: 500

Per day: 12000

Per week: 84000

Per month: 336000

Per year:38.4M (3.2Mx12 months)

**Data Requirements**

What is essential for a huge data storage is for it to handle a huge load of that flow, very large amounts of data and keep scaling to keep up with the company’s growth. Also for our system, it is of concern that it provides good Input/output operations per second, which is required to deliver the data to different analytics tools.

Hyper scale computing: A distributed computing environment where the volume of data and the demand for specific types of workload can sky rocket exponentially and still be accommodated quickly. This configuration is known to preserve the largest web-based operations which would satisfy our needs. The engine that drives our analytics team would be comprised of Hadoop, NoSQL or Cassandra. In addition adding PCIe flash storage into the server to result in cutting the storage latency to as low as possible.

**WANS**: Multiple cities containing WANS, preferably use 10 Gigabit Ethernet LAN PHY as it provides 10G bit/s, more than enough to accompany even thousands of clients simultaneously.

**Throughput:** Only a clients request and transaction will be packed and sent to through the server. One request consists of DriverID (5-6 INT’s (4 bytes ea=8bits ea) = 40-48 bits, UserID (also 40-48 bits), location (5-20 characters = 50-160 bits), date and time(10 numbers = about 80 bits) in total adding up to approximately 340 bits for the request throughput of one client. The transaction consists of the same information as the request (340bits) in addition to transactionID and payment amount, which would go up to 550-600 bits of data. Since this operation would only launch in New York, estimating about 500 requests in an hour, the maximum number of requests simultaneously would be (500 requests/ 60 minutes = 8.3 a minute) so ideally and preferably having 10 servers would eliminate potential conflict.

**Sign-ups:** Uber sees about 79,000 new sign-ups per week, which projects to 316,000 sign-ups per month and 3.8 million per year.

**Purpose:** Keep record as proof/to gather raw data/create collective data.

**Clients:** People in need of Car transportation

**Suppliers:** Gas stations

**Businesses we rely on**: bank credit card (uber gas discount credit cards)

**Transactions:** Multiple client requests responded to immediately, continuous and simultaneous flow of various driver/client before and after data transactions to store.

Money made per transaction: Base fare + Time + Distance

Data Storage System: Data Warehouse

**Fast Phases**

**Scope Definition**

Intent of project: To preserve time of drivers and money of clients, being as efficient as can be to use more if not all the space in the vehicle to satisfy more than 1 customers’ needs heading to the same direction at about the same time.

**WHAT:** What the new system is designed to do is, adding this extra feature would allow potential clients to drive with another passenger riding to a relatively similar destination.

**WHY:** This would not only save passengers’ money making it more desirable for them to use us, but also make potential employees(drivers) eager to join us as we are to develop a feature not yet promoted by any other vehicle service company. Furthermore, it would create a more efficient environment where the quantity of requests is satisfied to benefit the passenger, driver and ultimately UBER.

**HOW:** Since our employee fee will remain 20% per completed ride, our mission is to have more drivers and requests for a ride.

Time frame: 1 Year (estimated in the budget section)

**User interface**

The Web / App modification shouldn’t take very long as it is the simplest part of the system, taking from a week – 2 weeks for each modification (App and Web) to not only be designed but hooked up to our new system. In total about a month in time for this part of the system.

**Software**

The most complex and complicated part of our system, first the mathematical experts would need to design an algorithm that connects 2 or more passenger requests to a relatively similar destination, and create the calculation of transaction to generate more profit for Uber and the driver, and deduct expenses for the passenger(s). This would require at least 6 months of creating and testing until the satisfactory result is achieved. At the same time, our programming team would need to cooperate with the mathematicians to understand their necessities, which ultimately depend on the success of this feature. However, without the software to drive these computations and calculations nothing would work so I estimate the same 6 months of time for this part of the project. Overall, I would predict that this would take from 7-8 months to link, test and make it function.

**Data Storage**

Our programmers and analytics would be discussing the software and hardware requirements, but ultimately how data storing would operate and queried. This part of the project is not the challenging part but it isn’t the easiest either, after deciding what soft/hard ware we intend to use, creating a data flow diagram, tables, and collective data queries would take approximately from 2-3 months with the included time of how we would implement it into the system.

**Budget**

**4 Mathematical Experts:**(125k x 4 = 500k x 8/12months = 335k) for algorithm, computations and calculations.

**4 Senior Software Engineers**: (4 x 135k = 540k x 8/12months =360k) for the building and linking of functionality from user interface through the software to our database.

**2 College grad. Software Engineers:** (100k a year x 2 = 200k x 8/12months = 133k) (college grads. for the Web/App user interface).

**3 Data Analysists**: (65k a year x 8/12months = 42900$ per Analyst x 3 = 130k$) To interpret data, analyze results using their statistical techniques and provide reports; identify trends, filter and clean data and develop/implement data collection systems to optimize efficiency and data quality.

**2 Analytics Managers**: (100k salary) As Full time employers to our corporation, but for the project time ONLY (for 8-9 months) we would pay 75k. (for technical and organizational skills to identify and improve business structure / solutions for complications).

**1 Scheduling Manager**: (Salary from 30k-50k) We would hire for them for 8 months, paying the full 50k salary. (50k x 8/12months = 33K$) (For scheduling of actions throughout project and to see through that the project is going accordingly with respect to time).

Software and Hardware components = 100k

**TOTAL:** 335k + 360k + 133k + 130k + 75k + 100k= $1,113,000

**People in business**: Owners, managers, drivers, programmers, and analysts

# of People in on the project: 16

**People required for project**: Senior and college graduates in Software Engineering, Data Analysts, Project Managers and Mathematicians.

**Problem Analysis**

**Current and future OS**: The current system operates by starting with the request of a client, which pulls and displays their location and nearby drivers via GPS, they are able to look at a photo of the driver and their rating. After deciding on who, our tracking system records the driver, user, time and location. After arriving our system records the time and destination. Finally, arriving at the destination our system takes note of time elapsed, distance and amount to pay. With our new system coming, the improvements are relatively similar in a way they adapt more than 1 passenger, more than 1 location and more than 1 destination, and a transaction for each of the passengers computed with their tangent discount.

**5 System Problems**

1) The current system does not have a section devoted to this new system. Create option for driver to be part of the new proposed multi-ride system, reconfigure and add this additional feature as an option for clients using the app or online website.

2) There is no way for the current system to intake the new flow of information needed to analyze the status of the proposed system. An additional directory/storage device to adapt and archive the new flow of data.

3) Currently our system only allows for the simple transaction of satisfying a client from location to destination. We require new programs, algorithms and procedures for calculating locations, distances and prices. How would the algorithm give 100% efficiency / effectiveness for potential rides of client choices to their current location and their destination?

4) Our system would not support an instance where two clients request same ride, and on the way to 2nd client the request is canceled.

5) There isn’t an option for drivers to either be forced or free to decide if they’d be a multi driver. What if there are no close ride or it would take too long for one to arrive?

**5 System Improvement Objectives**

1) Re-configure the app and website to fit in the new feature option dimensions.

2) Create a relatively similar but more adaptive directory that suits our new operations means

3) Effectiveness is simply to hail the closest available driver that is not multi-riding; Efficiency on the other hand would save the client money but use more of their time. Extra time taken on tangent path is divided by 2 giving us x, then x is deducted from all boarded passengers costs after the completion of their ride.

4) In that case, the moment the request is canceled the driver would return back on course and from the time taken off tangent until the destination will be reduced from the final cost payment.

5) We would allow the driver to decide whether they would like to participate as a Multi-rider or not. Show GPS locations of other Uber drivers and locations that require attention/hotspots/common places of requests

**Business Process Requirements:**

**Passenger**

1) Download free app/ use online web and log into your account

2) Choose vehicle type: black car, Taxi, UberX, SUV, LUX. (or use nearby drivers)

3) Mark your location and wait in front of given address

4) Payment processed through credit card information once ride is complete

**Driver**

After logging in, choose car type and submit which will activate your GPS location and you’re in business.

**Data Storage**

I believe a Data warehouse is required due to needing a tremendous amount of data organized by rows. In addition, the data has to be up to date and we need to keep all historical data as well. Need to store user account information from both drivers and clients.

**Requirements Analysis**

**Signing up for Uber**

**Client**

1) Visit Uber website create account requiring name, photo, mobile number, email, billing information and provide a username and password.

2) Read terms and policies, accept and activate account through email.

**Driver**

Provide their information; a name, email, mobile phone number, give a password and the city you’re located in, and tell us where did you hear about Uber. Furthermore, you would provide the car type you drive.

**Functional Requirements**

**Internal**

**Admins/ Managers:**

1) Retrieve information (from data analysts) to display for various participants in need of precise information such as success and failure rate of the new operations in place (Revenue/losses).

2) Be able to login and have the authenticated privileges of banning/unbanning drivers and/or their privileges.

**Analytics:**

1) Gather raw data (query data base) and create organized collective data/ charts, graphs to display for various participants in need of precise information.

2) Figure out new trends and best practices for the company, for instance you can’t hail an Uber cab currently but we might be able to make it possible.

**External**

Clients:

1) Provide feedback after the ride about the driver/rating, the passengers and their experience with the operating system.

2) See driver info AND accompanying passengers of the ride with their requested destination.

Suppliers:

1. Gas stations: Provide Uber drivers with a specific credit card discounts on gas.
2. Drivers: Respond to requests, get location of client and travels to destination

**List of Entities**

1) The owner(s)/founder(s)

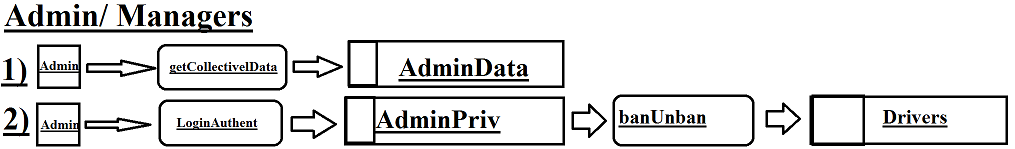
2) The Admins, managers, business analytics and programmers

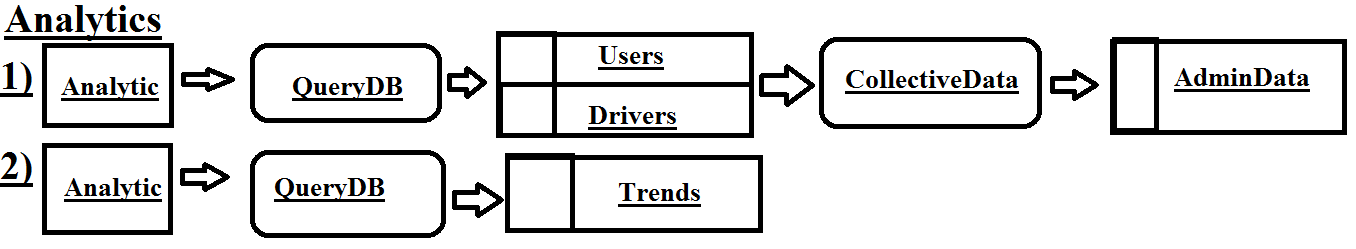
3) The Clients

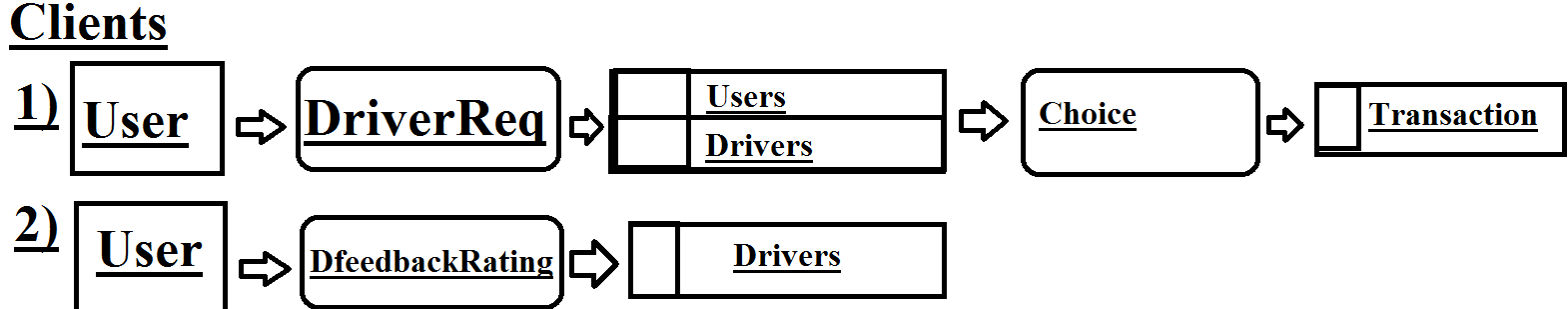
4) The Drivers

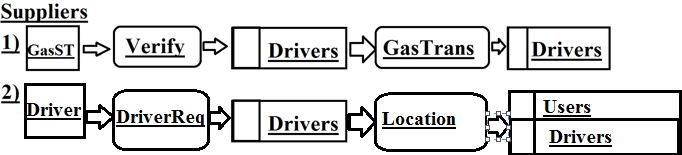
5) The Gas station(s)

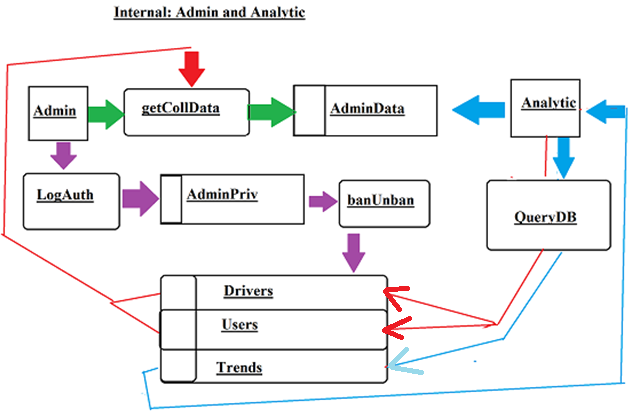
**Logical Design**

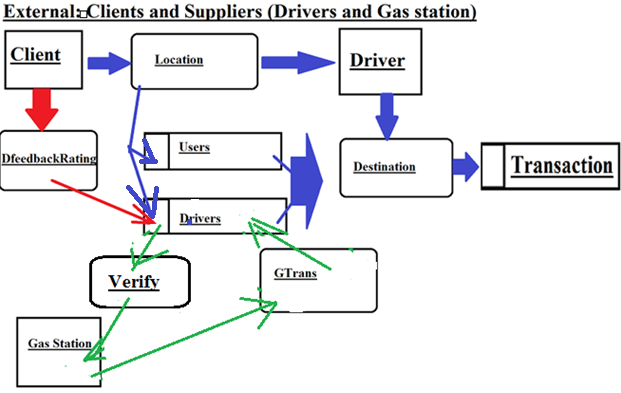












**Decision Analysis**

Technical Feasibility

We require some math specialists in order to come up with an algorithm so that it not only radar the nearby uber drivers to the client but compute which one would be most efficient / or most effective. It is also essential to have some educated programmers to interact the algorithm with a program that processes options for the client, their choice, etc. As previously stated, the project should take up to a year if not more to complete. It will not be easy, but it is entirely possible.

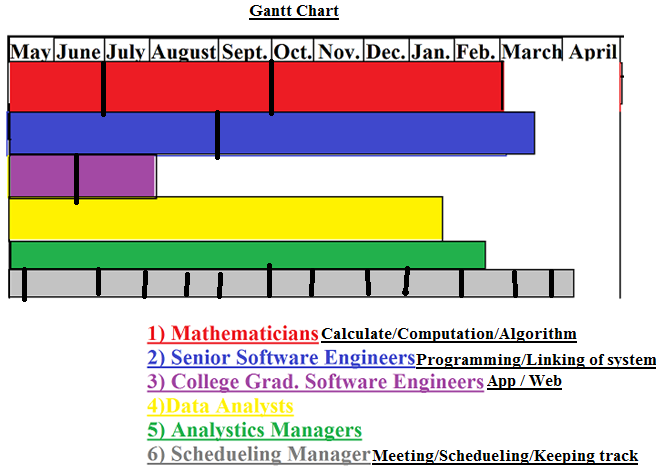
Economic Feasibility

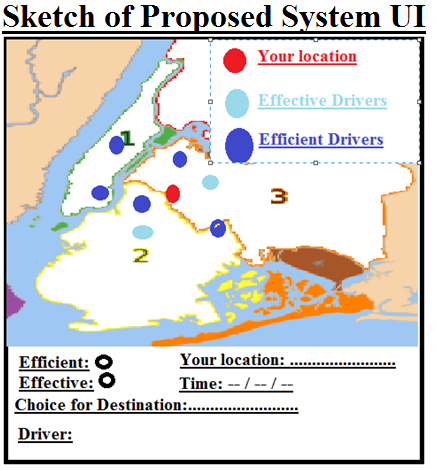
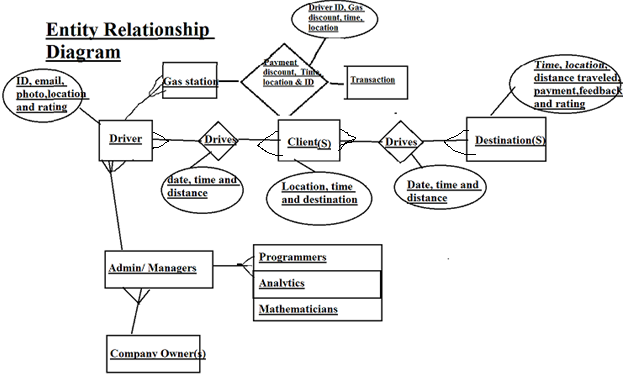
The Information in the scope definition is more or less accurate and remains the same.

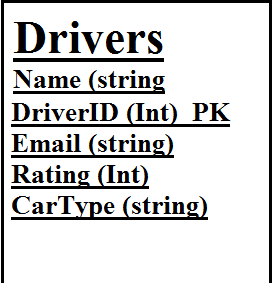
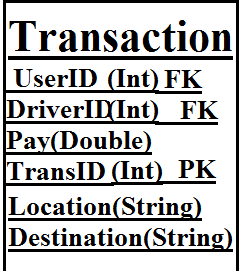
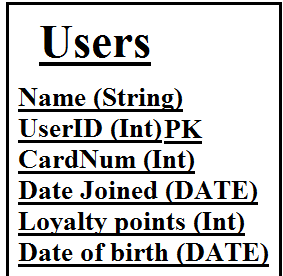
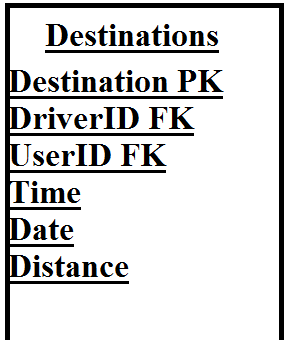
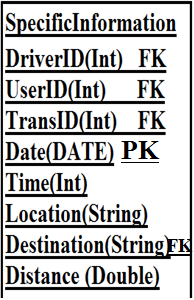
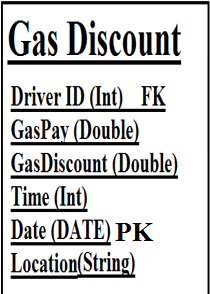
Operational Feasibility

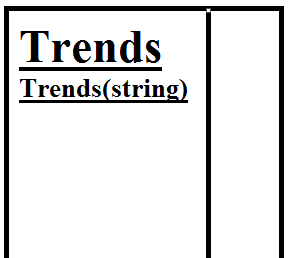
The system is definitely improved; no other cab company can do the unique platform we have created for multiple clients to be supported by one vehicle in a common destination. On top of this, the Uber and the driver make more money while saving the client money. Instances such as driver using less gas to get more people satisfied and the people are satisfied by paying less.

Scheduele feasibility



**Physical Design & Integration**



**Method Signatures**

**Admins/ Managers**

1) Object getCollectiveData (Int TransactionID)

Input data type: For all the orders associated with this new feature.

Output data type: For admin/ managers to analyze results.

2) Object LoginAuthent (DriverID)

Input data type: To track down the particular driver.

Output data type: To see the specific qualities that determine the action to be taken upon the driver.

**Analytics**

1) Object QueryDB(Int UserID, DriverID, TransID CardNum, DateJoined, Dateofbirth, Time, Rating)

Input data type: A lot of specific qualities that make up a transaction.

Output data type: Organized information to be analyzed

2) String QueryDB(String Location, Destination, Car type, Email, Name)

Input data type: To have characters that identifies a request

Output data type: List of information

3) String Trends(String Trends)

Input data type: To have flexibility of what are the specifics that contribute to a trend

Output data type: Retrieval of that trend to discuss about.

**Clients**

1) Object DriverReq (Int DriverID, UserID, Time, Rating)

Input data type: Driver details

Output data type: Display characteristics of potential choice of ride

2) String DriverReq(String Name, Location)

Input data type: A string of charecters to make up name and location

Output data type: Detailed information of request

**Suppliers**

**1)** Double GasTrans (Int DriverID, GasPay, GasDiscount)

Input data type: Transaction criteria

Output data type: May be a decimal value which INT can’t handle

2) String Location(String Location, Date, Time)

Input data type: String the transaction that took place

Output data type: Store it as a string.

**Construction & Testing**

**Analytics**

1) After a click of a button the system queries with the method to acquire desired information, test passes.

2) If Analytics can retrieve various versions of trends they have added and modified in the past, test passes.

**Admins/Managers**

1) If the press of a button the Admins receive the collective data stored by the Analysts, the test passes

2) If authentication is secure and distinct for each admin, after logging in and having the privalages of banning, unbanning or modifying drivers status, the test passes.

**Clients**

1) See if Driver information and passanger information comes up everytime, the test passes

2)See if Option for feedback on the new operating system pops up after completion of ride, if so test passes.

**Suppliers**

1) If gas discounts are accounted for and processing functionally then the test passes.

2) If the driver is signaled by a requesting passenger and is given the location(s) of passenger(s), the test passes.

**Installation & Delivery**

In conclusion, potential requests can be diverted by more efficient drivers as to preserve gas and money. It will cost time for both the passenger and driver. Initially the client would select the efficient option which would display the available drivers, from there they would input their destination and the request would match up best driver that is heading to the most common spot and more likely than not will be the choice of the client. Displaying the time the drive would take and the cost deduction (how much they save) they would wait, be picked up and driven to their destination. If truly successful in the region of New York, we would make corrective adjustments to make the system more flaw-proof and be launched throughout all currently active locations filled with uber drivers.