

Team S4, Andrew Wang, Jeremy Lee, Rick Huang, Akash Kejriwal

Our Uber Story: Uber seeks to reinvent the transportation industry by offering peer-to-peer transportation services by opening business opportunities for drivers and users. Uber aims to work in the same marketplace that taxis and public transportation operate in, by offering users inexpensive and personalized transportation to unique destinations and the opportunity for drivers to supply this transportation in exchange for market demand compensation. Uber supplies the platform to connect these users and charges a marginal amount for each transaction.

User Stories:

User One : Driver - One who drives the Uber car, must have one Uber car.

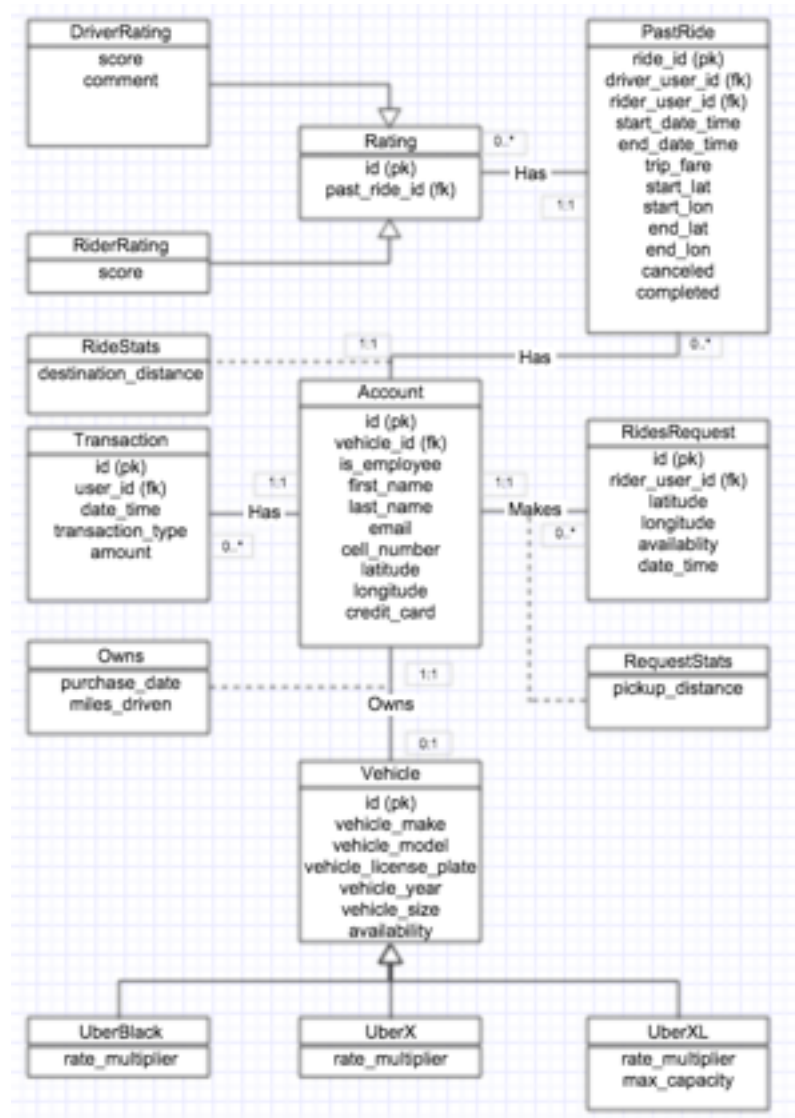
User Two : Rider - One who uses the Uber application and/or rides the Uber car.

User Three : Customer Service Rep - One who handles customer calls and contacts including account inquiries, complaints, or support calls.

User Stories

#	I want <role>	I want <goal>	So that <reason>
1	Driver	to see all the users locations that have requested rides	find new riders.
2	Driver	to find the closest ride to my current location and automatically accept it	complete a trip and earn a fare.
3	Driver	turn on and off availability	choose when I work.
4	Driver	to see the final destination	I can deliver the rider.
5	Driver	to find the most profitable location to pickup riders at	I can maximize my earnings.
6	Rider	rate drivers	filter good/bad drivers.
7	Rider	select available car types	so that I can choose the specific car for my purpose.
8	Rider	see a history of my rides	remember where I went.
9	Customer Rep	query ratings for a user/ driver	filer users who are irresponsibly abusing the system.
10	Customer Rep	see date and time for rides for a particular user	I can relay accurate information.
11	Customer Rep	to know the number of requests at the most popular pickup location of radius x	I can recommend to Uber drivers and administration what areas should have a higher influx of Uber drivers.

Initial UML Diagram :



Relational Schema:

Entities

Account (id (pk), vehicle_id (fk), is_employee, first_name, last_name, email, cell_number, latitude, longitude, credit_card)

Transaction (id (pk), user_id (fk), date_time, transaction_type, amount)

Vehicle (id (pk), uber_type, vehicle_make, vehicle_model, vehicle_license_plate, vehicle_year, vehicle_size, purchase_date, miles_driven, max_capacity, rate_multiplier, availability)

RidesRequest (id (pk), rider_user_id (fk), latitude, longitude, availability, date_time, pickup_distance)

RiderRating (id (pk), past_ride_id (fk), score)

DriverRating (id (pk), past_ride_id (fk), score, comment)

PastRide (id (pk), driver_user_id (fk), rider_user_id (fk), start_date_time, end_date_time, trip_fare, start_lat, start_lon, end_lat, end_lon, canceled, completed, destination_distance)

Functional Dependency:

Vehicle

Bad

vehicle_model -> uber_type, vehicle_make, vehicle_size

Fixed dependency

id -> vehicle_model_id (fk), vehicle_year, vehicle_license_plate, availability

id -> vehicle_model_name, uber_type, vehicle_make, vehicle_size

Normalized Schema:

Vehicle (id (pk), vehicle_model_id (fk), vehicle_license_plate, vehicle_year, purchase_date, miles_driven, max_capacity, rate_multiplier, availability)

VehicleModel (id (pk), vehicle_model_name, uber_type, vehicle_make, vehicle_size)