Carpooling App

This project focuses on designing a database for a carpooling app to help map users with other users to carpool with. The aim of the carpooling app is to help students, staff and faculty members of Syracuse University commute in and around the city. Carpooling app would help the commuters of Syracuse University take more informed decisions which would help them reduce expenses, save time, make new friends and help the environment as it cuts down the number of cars on the road. Fewer the number of cars on the road, lesser will be the rate of carbon emission, fuel consumption, noise pollution and traffic.

The proposed system will store information about ride offerors, ride takers, route, etc. The total fare includes a base price, waiting charge and service fee. Base price - Carpool app calculates a base price that makes commuting more affordable for both riders and drivers. It is based on the distance of travel the number of riders in the carpool. Waiting charge is applicable if a rider delays by over 3 minutes. Service fee - Service fees are currently waived. The assumptions for the proposed database system are that all cars offered for the ride are in good condition. The distance of the route is measured in miles.

People who own their own cars travel with vacant seats and often spend too much on fuel. Whereas people who don't, make use of cabs or buses. By utilizing the availability of seats for one party and the requirement of seats for another, the project aims at bridging the gap, thereby reducing traffic, pollution and expenses. To commute around Syracuse, students, staff, faculty members have to travel either by cab, bus or by their own cars. Traveling by cabs every day becomes heavy on the pocket. Likewise traveling in personal cars becomes costly as people have to pay for fuel as well as the maintenance of the car. Buses have a fixed schedule, so it becomes difficult to match this schedule with that of the commuters. Also, for commuters who stay far away from bus stops, it becomes even more difficult to travel through buses as they have to leave early in order to catch a bus. Traveling by buses also consumes a lot of time as the bus has a fixed route with a high number of stops.

For this we need to structure a database that acts as a central repository for all the data pertaining to ride offerors and ride takers. Drivers and riders are mapped automatically whenever they want to commute in Syracuse. This system will display different views for the riders and the drivers, so that each view retrieves information relevant to the type of user. This provides access control and prevents misuse of private information. The purpose of this system is to reduce expenses, save natural resources and save time.

The analysis, functional dependencies and relationships between the various entities and attributes is shown in the entity relationship diagram(ERD). The business rules are also mentioned in the report. The report also addresses major data questions that that can be used to improve the app in later versions so that it increases the satisfaction of users.

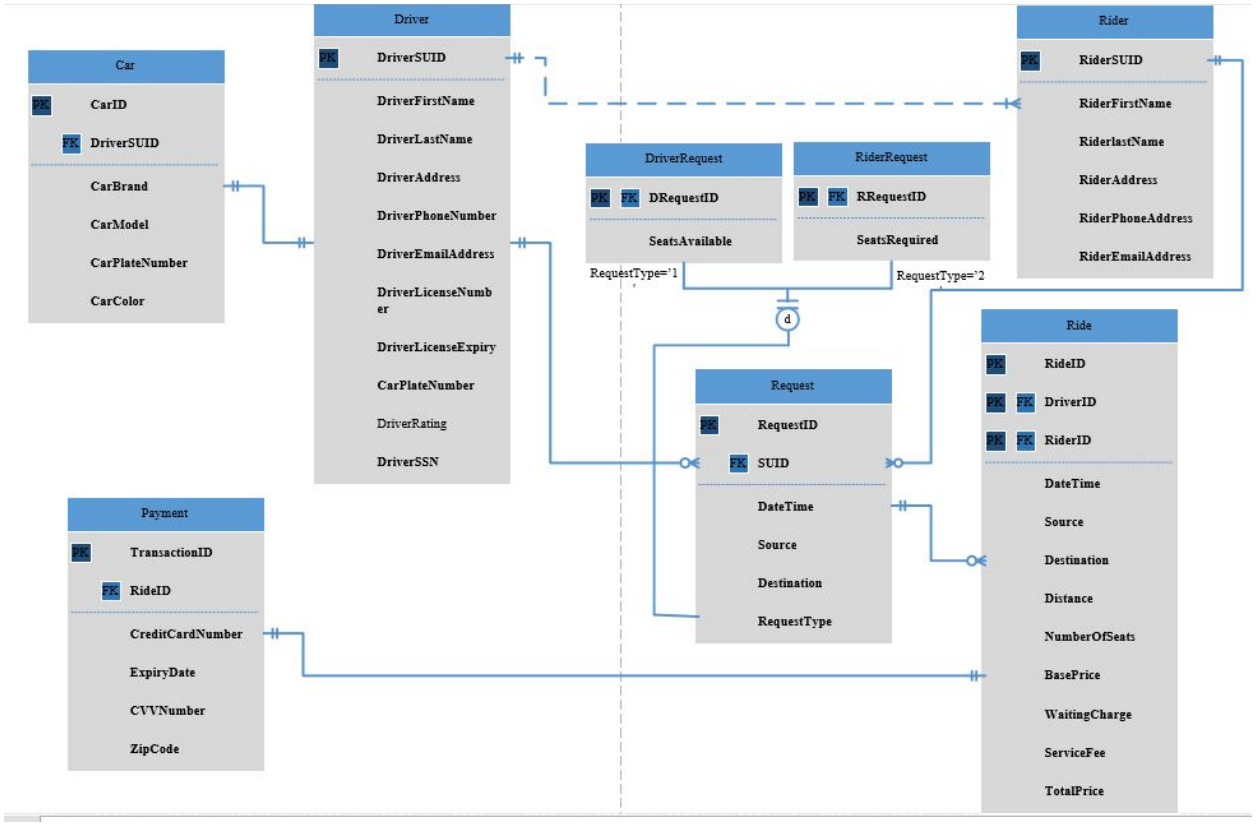
Entities and their Attributes:

Objects	Description
1) Driver	**A person who wants to offer ride**
Driver SUID	Primary key/ Unique identifier for the driver
DriverFirstName	First name of the driver
DriverLastName	Last name of the driver
DriverAddress	Address of the driver
DriverPhoneNumber	Phone Number of the driver
DriverEmailAddress	Email address of the driver
DriverLicenseNumber	Driver license number
DriverLicenseExpiry	Driver license expiry
CarPlateNumber	Driver's car plate number
DriverRating	Driver Rating
DriverSSN	SSN of Driver
2) Rider	**A person who wants to take ride**
RiderSUID	Primary key/ Unique identifier for the rider
RiderFirstName	First name of the rider
RiderLastName	Last name of the rider
RiderAddress	Address of the rider
RiderPhoneNumber	Phone Number of the rider
RiderEmailAddress	Email address of the rider

3) Payment	**Payment of each ride**
TransactionID	Primary key/ Unique identifier for the transaction
RideID	Foreign key/ Unique identifier for the ride
CreditCardNumber	Credit Card Number of the payer
ExpiryDate	Expiry date of the credit card of the payer
CVVNumber	CVV Number of the credit card of the payer
ZipCode	Zip code of the payer
4)Car	**Car offered for carpooling**
CarID	Primary key/ Unique identifier for the car
DriverSUID	Foreign key/ Unique identifier for the driver
CarBrand	Brand of the car
CarModel	Model of the car
CarPlateNumber	Driver's car plate number
CarColor	Color of the car
5)Request	**Request made by driver or the rider**
RequestType	Identifies who has requested the ride: rider(2) or the driver(1)
RequestID	Primary key/ Unique identifier for the request
SUID	Foreign key identifier of the rider/driver
DateTime	Date and time at which the request was made
Source	Source location of the requestor
Destination	Destination of the requestor

6) DriverRequest	**When RequestType='1'**
DRequestID	Primary/Foreign key identifier of the driver
SeatsAvailable	Number of seats available in car
7) RiderRequest	**When RequestType='2'**
RRequestID	Primary/Foreign key identifier of the rider
SeatsRequired	Number of seats requested by the rider
8) Ride	**Details about the ride**
RideID	Primary key identifier for the ride
DriverID	Foreign and primary key identifier for the driver
RiderID	Foreign and primary key identifier for the rider
DateTime	Date and time at which the ride started
Source	Source location of the ride
Destination	Destination of the ride
Distance	Total distance between the source and destination
NumberofSeats	Number of seats taken
BasePrice	Base fare of the ride
WaitingCharge	Waiting charge of the ride
ServiceFee	Service fee of the ride
TotalPrice	Total price of the journey

ERD:


Car
CarID
DriverSUID
CarBrand
CarModel
CarPlateNumber
CarColor
Driver
DriverSUID
DriverFirstName
DriverLastName
DriverAddress
DriverPhoneNumber
DriverEmailAddress
DriverLicenseNumber
DriverLicenseExpiry
CarPlateNumber
DriverRating
DriverSSN
DriverRequest
DRequestID
SeatsAvailable
RiderRequest
RRequestID
SeatsRequired
RequestType='1'
RequestType='2'
d
Rider
RiderSUID
RiderFirstName
RiderlastName
RiderAddress
RiderPhoneAddress
RiderEmailAddress
Request
RequestID
SUID
DateTime
Source
Destination
RequestType
Ride
RideID
DriverID
RiderID
DateTime
Source
Destination
Distance
NumberOfSeats
BasePrice
WaitingCharge
ServiceFee
TotalPrice
Payment
TransactionID
RideID
CreditCardNumber
ExpiryDate
CVVNumber
ZipCode

Business Rules:

- The users of the system are either riders or drivers.
- A requestor must have a SUID.
- Driver must have valid driver's license.
- Driver car must have car registered in his name.
- Request type identifies driver and rider requests.
- One driver can accommodate more than one rider request if number of seats available is greater than or equal to number of seats requested.
- Waiting charge is applicable if a rider delays by over 3 minutes.

Major Data Questions:

- What is the ranking order of drivers based on rating?
- What is the shortest route?
- What is the longest route?
- Which car is more comfortable?
- What are the peak hours for usage of the system?
- What are the popular areas of pickup?
- What are the popular areas of drop?
- Does supply meet demand?