DAMG6210 Data Management and Database Design
Group 18
Carpooling

Members:
Sutiksh Verma: 002122052
Ashish Shethia: 002776691
Varun Yepuri 002771801
Ratnesh chimnani 002795124
Krishanand Jha 002768931

Mission Statement:
This project demonstrates the use of database to solve the problem of individuals facing difficulty to find the most efficient and cost effective rides to their destination for eg Companies, University or Planned Trips.

Mission Objectives:
Centralized system that consists of all the rides data, will be accessible by any individual who is enrolled in the university for eg students, faculty or staff members

Determining when and where to organize a car ride based on the set threshold (a pre-decided minimum number of people decided by the car owner).

A passenger can post a request the details about the ride (Source, destination and time, and if other people are tagging along)

A person can post about their trip to a pre-destined location and other people can join in by putting up a request and they will be selected on a first come first serve basis.

An accurate database will help car owners to organize and manage their day to day rides and cut down their cost of gas and efficiently plan their route so they can cater to every request.

The database system can be accessed only by the authorized personnel, leading to a high security environment for sensitive user information.

The cost will be decided in two factors:
1. If someone owns a car: The number of miles the passenger is travelling and the fuel cost to ride that distance
2. If someone is renting a car: Basic rent divided among the number of people with extra fuel charges if applicable
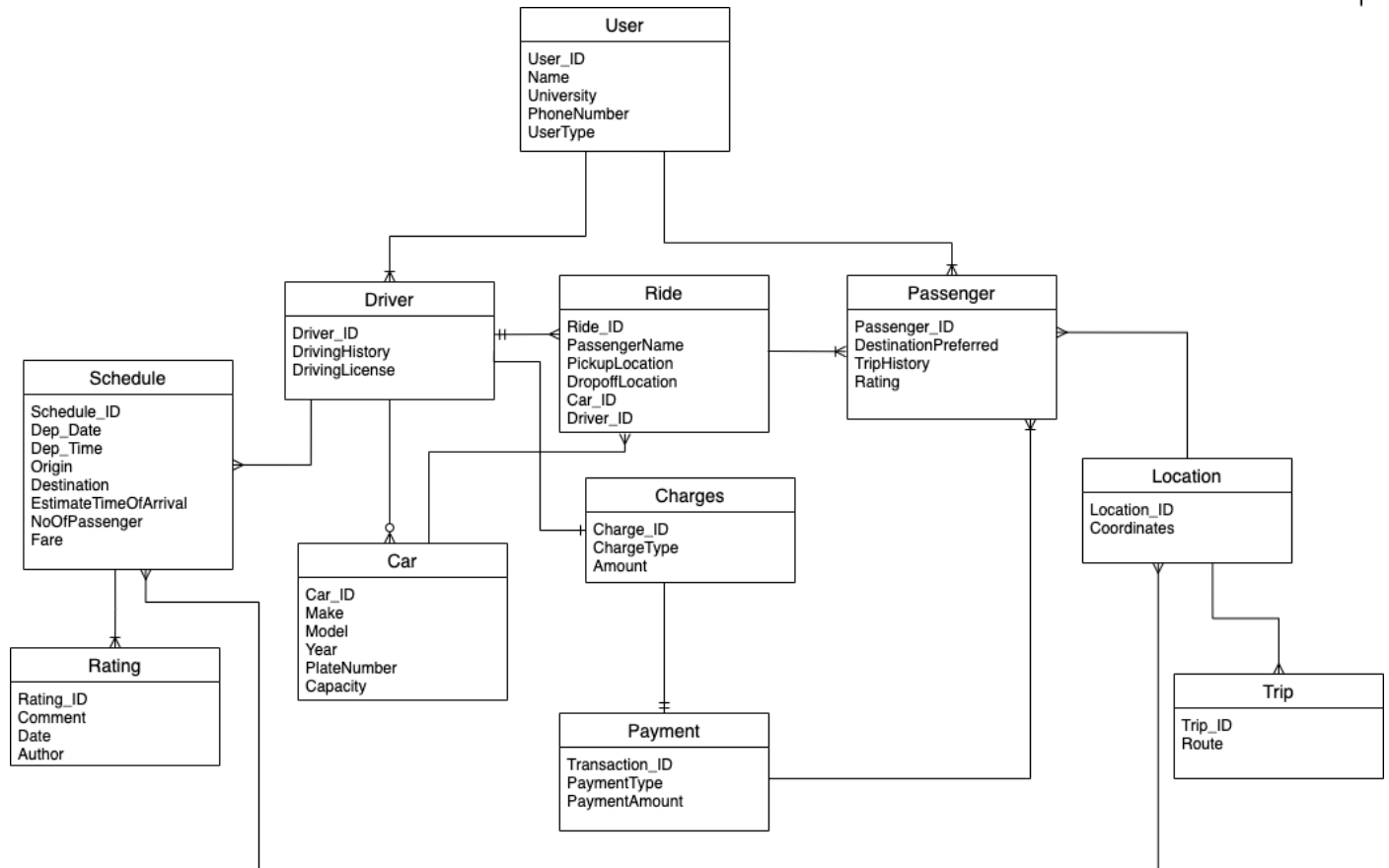
While implementing this database, there are relationships must be followed. Few of them are as follows:

- Firstly There are two types of users that are Driver and Passenger
- Driver can Schedule all his trips in advance. One Driver can Schedule many trips but one schedule can be made by only one driver indicating one to many relationship
- One Schedule can have multiple ratings but a single rating will be given to one schedule indicating one to many relationship
- One driver can have many cars but one car is owned by a single driver indicating one to many relationship
- One driver will organize many Rides, but one ride must be organized by a single driver indicating one to many relationship
- Driver must release a single Charge statement and one charge statement is generated by only one driver indicating one to one relationship
- Charges are converted to Payments, one charge statement must be converted to a single payment and one payment is made out of one charges indicating one to one relationship.
- Payments are further split into different passengers, so many passengers will make their individual payment and payments will be done by many passengers indicating one to many relationship
- Passengers can select their respective drop off locations. One location can have many passengers but one passenger will be only going to one location. indicating many to one relationship.
- Finally one Location can be included in multiple trips but one trip will have only one specific drop of location indicating one to many relationship

To design a database for this system, we have identified the following entities must be:

- User
- Driver
- Passenger
- Schedule
- Ride
- Car
- Charges
- Payment
- Location
- Trip
- Rating

EER:

User
- User_ID
- Name
- University
- PhoneNumber
- UserType

Driver
- Driver_ID
- DrivingHistory
- DrivingLicense

Ride
- Ride_ID
- PassengerName
- PickupLocation
- DropoffLocation
- Car_ID
- Driver_ID

Passenger
- Passenger_ID
- DestinationPreferred
- TripHistory
- Rating

Schedule
- Schedule_ID
- Dep_Date
- Dep_Time
- Origin
- Destination
- EstimateTimeOfArrival
- NoOfPassenger
- Fare

Car
- Car_ID
- Make
- Model
- Year
- PlateNumber
- Capacity

Charges
- Charge_ID
- ChargeType
- Amount

Location
- Location_ID
- Coordinates

Rating
- Rating_ID
- Comment
- Date
- Author

Payment
- Transaction_ID
- PaymentType
- PaymentAmount

Trip
- Trip_ID
- Route

Reference data:

Reference data refers to the static data that provides a common understanding of the Car pooling and the entities involves in it. This type of data is used to establish a common understanding of the Car pooling and to support decision-making processes.

For example, Reference data will include :

- Users
- Drivers
- Passengers
- Schedule
- Cars
- Location
- Trip
- Ride

Transactional data:

Transactional data refers to the data is generated as a result of the day-to-day activities in the car pooling. This type of data provides a detailed view of the activities that are taking place in the car pooling and is used to track the flow of charges and payments, also the ratings given by the user

For example, Transactional data will include:

- Charges
- Payment
- Rating