Carpooling System

Problem Statement:

Carpooling has become a practical, cheap and stress-free way to move around. This project presents the requirement, design and implementation of an enterprise-class application for carpooling.

The added features, compared to available applications, are different kinds of trips, a check-in system and social media integration. The two kinds of trips are normal trips, which are trips belonging to the regular category and frequent trips which are the ones that commuters do every day. The check-in system enables users to check in at meeting points and notify all users about that. Users can also share their activities on the application thanks to social media integration. Application is also a real-time application: any person taking part of a trip can check in the meeting point to let the other persons know he/she has arrived at the meeting point.

Functional Requirements:

- Registration screen where new users can register themselves.
- Log in screen that allows registered users to log in and create trips or register for trips.
- Profile modification screen where a user can edit any of his information.
- Search trip screen that gives the choice to either search a single or frequent trip.
- Create trip screen that gives the choice to either create a single or frequent trip.
- Rate user screen that shows all the recent users that the user has taken a trip with. Then it gives him the ability to rate those users.
- Check in screen where the user can check in the trip when he reaches the meeting point.

Key Features Required:

1. System should allow user to register on the website and book the Car.
2. System should generate different notifications as per requirement.
3. System should maintain the bookings records.
4. System should be integrated with the Google Map and GPS.

Technologies to be Used :

- UI Layer : AngularJS, BootStrap UI with Spring MVC
- DB : MySQL
- JDK : 1.8
- Spring Framework
- Logging : Slf4j with Log4j
- Build Tool : Apache Maven

Types of System Users :

Actor	Description
Passenger	Responsible for managing user activities such as search for trip, join trip,etc.
Driver	Responsible for posting the trips (Regular/Frequent), share the trip,etc.

Epics of Car Pooling System :

Epic Id	EPIC Name
EP-CPS-1	User Management
EP-CPS-2	Trip Management
EP-CPS-3	Social Media Sharing
EP-CPS-4	Notification Management

USER STORIES

User Story **: EP1-US-CPS-1**

User should be able to register or login to the system

Business Rules:

- For each new user unique ID should be generated.
- Username and Password of the user should be validated.

Solution

- Username and password cannot be left blank.
- Application should allow the driver/passenger to register with the system.
- New driver/passenger has to register first in order to book a seat.
- User should be able to use the functionalities provided by the application.

User Story : **EP1-US-CPS-2**

User should be able to update/edit the profile information.

Business Rules :

- Only registered users should be allowed to edit profile information.
- System should allow user to provide valid information only.

Solution:

- User should be able to see his/her existing information
- System must allow driver/passenger to update his existing information.
- Validation of fields should be done accordingly

User Story : **EP1-US-CPS-3**

User should be able to search for trips.

Business Rules :

- User should be able to search trips based on the Destination, type, origin, departure, date/time.
- User's search preferences should be taken into consideration.

Solution:

- User should be able to select starting point and ending point.
- User should be able to enter preferences
- User should be able to browse through the posted trips

User Story : **EP2-US-CPS-4**

Driver should be able to create/post a trip.

Business Rules :

- Driver should be able to create a new trip.
- Driver should be be able to provide the preferences.
- Unique Trip-ID should be generated for newly added trip

Solution :

- Driver should be able to select the type of trip
- Driver should be prompted to provide the trip preferences and related information

User Story : **EP2-US-CPS-5**

Driver should be able to provide Terms & Conditions

Business Rules:

- The User or Driver must be particular with the conditions and terms before creating the trip, and so the passengers who can travel depends upon either the rating of his/her before trips or the acceptance to the posted terms and conditions

Solutions:

- Without the prior agreement from the passengers the driver will not initiate the trip once he/she have created the trip.

User story : **EP2-US-CPS-6**

Driver should be able to select Meeting point

Business Rules:

- Only the Driver who creates or starts the trip can decide the meeting point selection.
- There can be multiple meeting points respectively according to the driver's choice.

Solutions:

- Meeting point can be fixed by driver from the source point surroundings.
- Multiple meeting points can be made depending on the driver's choice.

User story : **EP2-US-CPS-7**

Passengers should be able to join a trip

Business rules:

- Passenger should be able to choose starting and ending point
- Passenger can provide preferences like having luggage, allowing to smoke, pets,etc.

Solution:

- Passenger selects the location from where he have to start and end the end the journey.

- Passenger can select the preferences whether he want driver who allows smoking or not, whether he wants to carry luggage or not, what could be the bag size, pets, etc.

User story : **EP1-US-CPS-8**

Passengers as well as drivers should be able to rate each other.

Business rules:

- Passenger needs to know about the driver as it is a matter of safety.
- Driver needs to know what kind of passenger is going to join him.

Solution:

- Passenger can rate the driver on certain basis, which will help other passengers to know about the driver.
- Driver will also rate the passenger, and that will in turn be helpful to other drivers to have some information about passenger who is going to join him.

User story : **EP1-US-CPS-9**

Driver can withdraw the trip at any point of time.

Business rules:

- Driver no longer wants to continue with the journey.
- Driver is not even starting the journey.

Solution:

- If driver is not willing to continue with the journey than he can withdraw the trip.
- Driver can also withdraw if he does not wish to start the journey. (In these cases, user will be notified and he may have to arrange his trip with some other driver.)

User Story : **EP2-US-CPS-10**

User should be able to check seat availability and cancel the booked seat

Business Rules:

- The passengers who are willing to join the trip must be allowed to book the seat which are available of his/her own choice.
- Once the passengers book the available seats in the car, he/she can also cancel the booking of seats.

Solutions:

- In the Web Application the passengers or the users are allowed to select their placement according to their choice from the available seats in car.
- Passengers are also provided with the option of cancelling the booked seat, in case of emergency.

User story : **EP3-US-CPS-11**

Drivers should be able to share their activity on social media after posting the trip information.

Business rules:

- A suggestion for sharing trip's creation,registration should pop-up.
- Application should be connected to user's social media account.

Solution:

- Application should allow Drivers to share the trip data on social media
- Social media sharing can be done using Oauth API
- Sharing should be authorised by user

User story : **EP2-US-CPS-12**

Driver should be able to see the request for joining the trip and be able to approve

Business Rules :

- Driver should be informed about new spot registration.
- Driver can not approve the requests more than the car capacity
- Driver should be able to deny the joining request

Solution :

- Driver should get the notification if he/she receives any request for joining the trip
- Driver should be able to approve or deny the joining request based upon ratings for passenger

User Story: **EP4-US-CPS-13**

System should notify the users when the driver or passenger reaches the meeting point.

Business rule :

- As soon as the the user reaches the meeting point, other users should be notified.

Solution:

- Check-in function in the system will notify the user that driver or passenger has reached the meeting point.
- Every other passenger involved in carpooling will get a notification that someone is waiting at meeting point.
- If the GPS location of the user matches with the meeting point then only the Check-in function can be used.

User Story:**EP4-US-CPS-14**

Driver should get a notification about reserving/cancelling the seats.

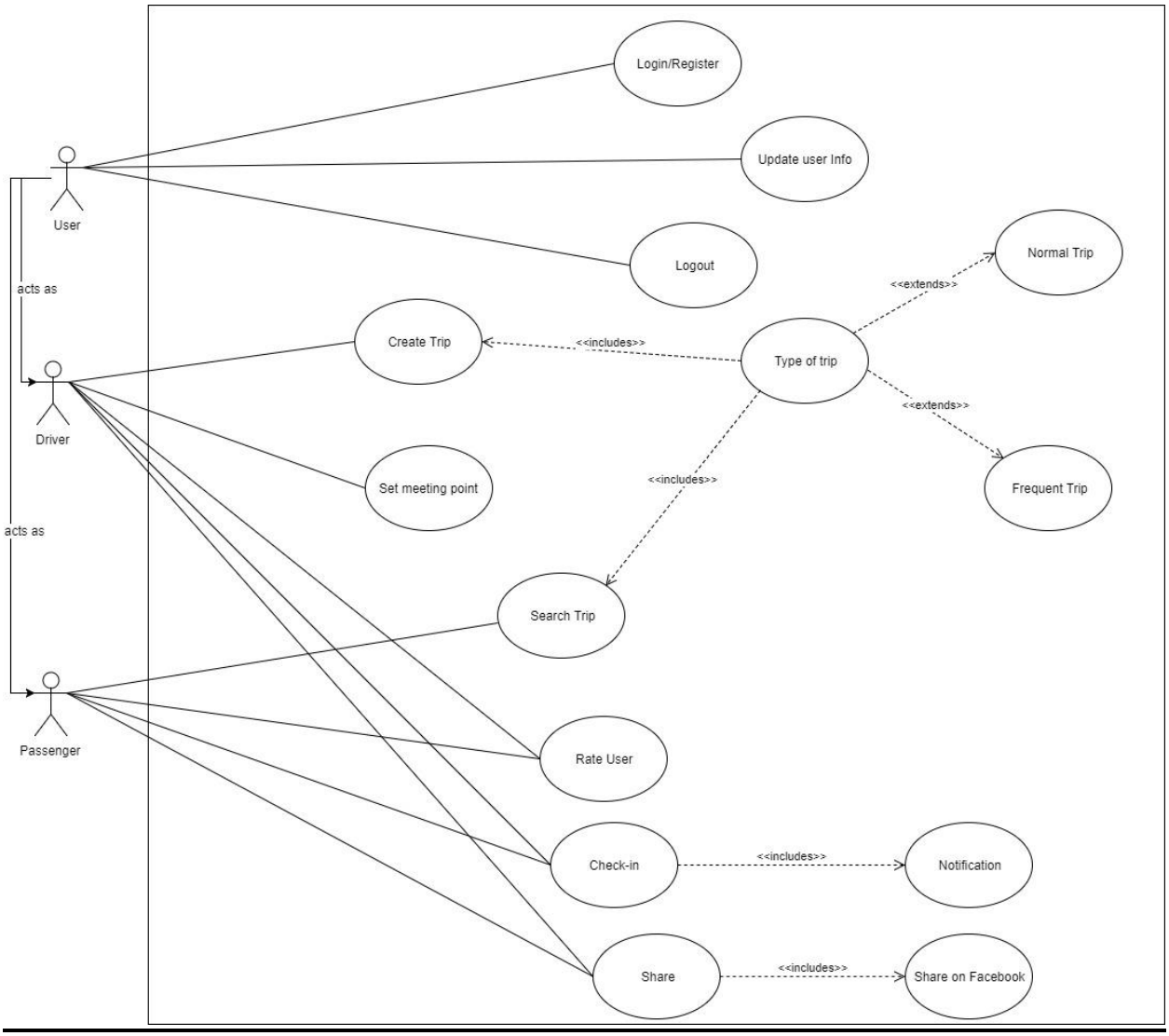
Business Rule:

- The amount of bookings should not exceed the limit of the vehicle.
- Cancelled seat should vacant and should be visible to other users.

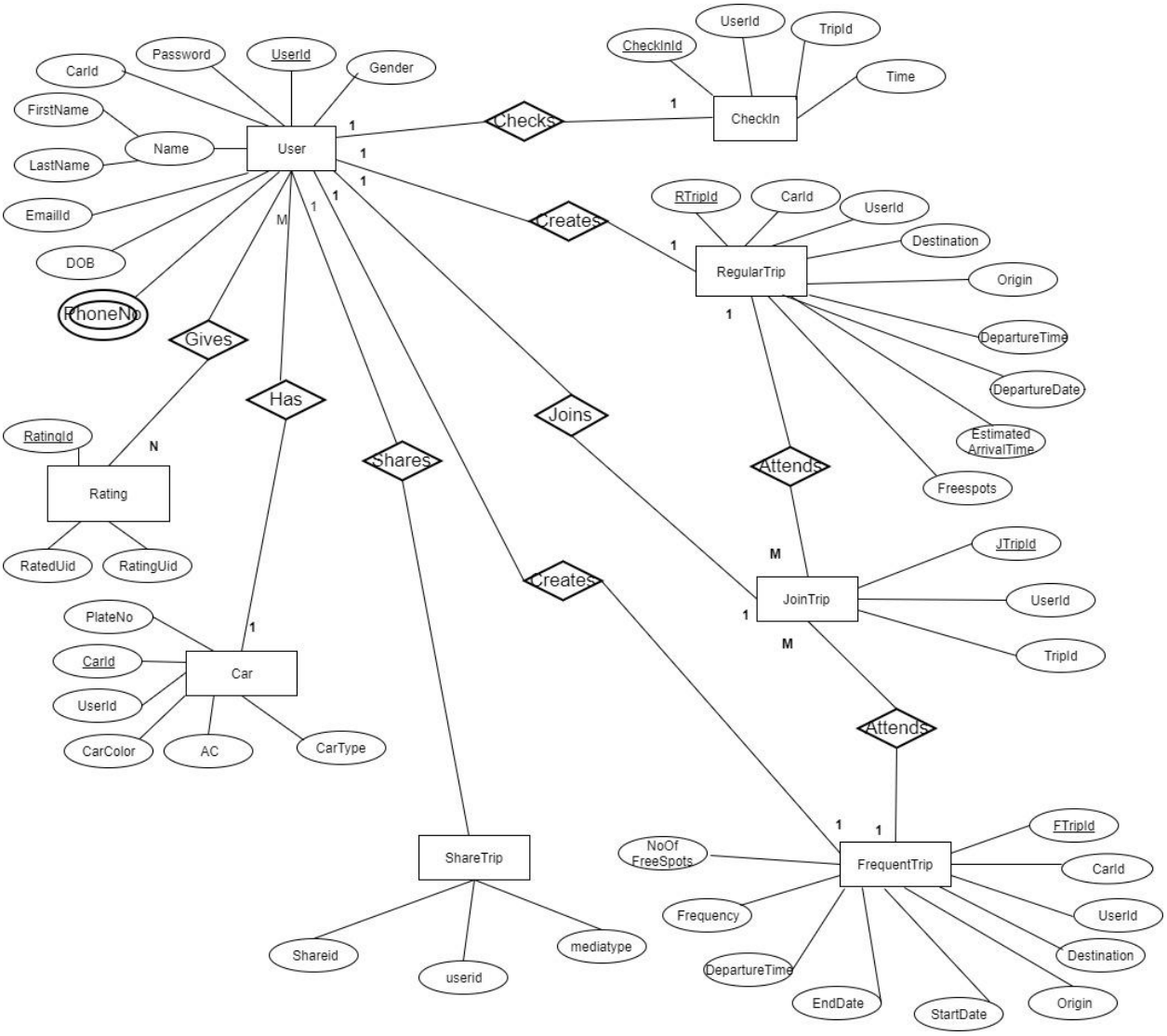
Solution:

- System will notify the driver whenever user books a seat.
- Driver will be notified whenever any user cancels the seat.

Usecase Diagram:

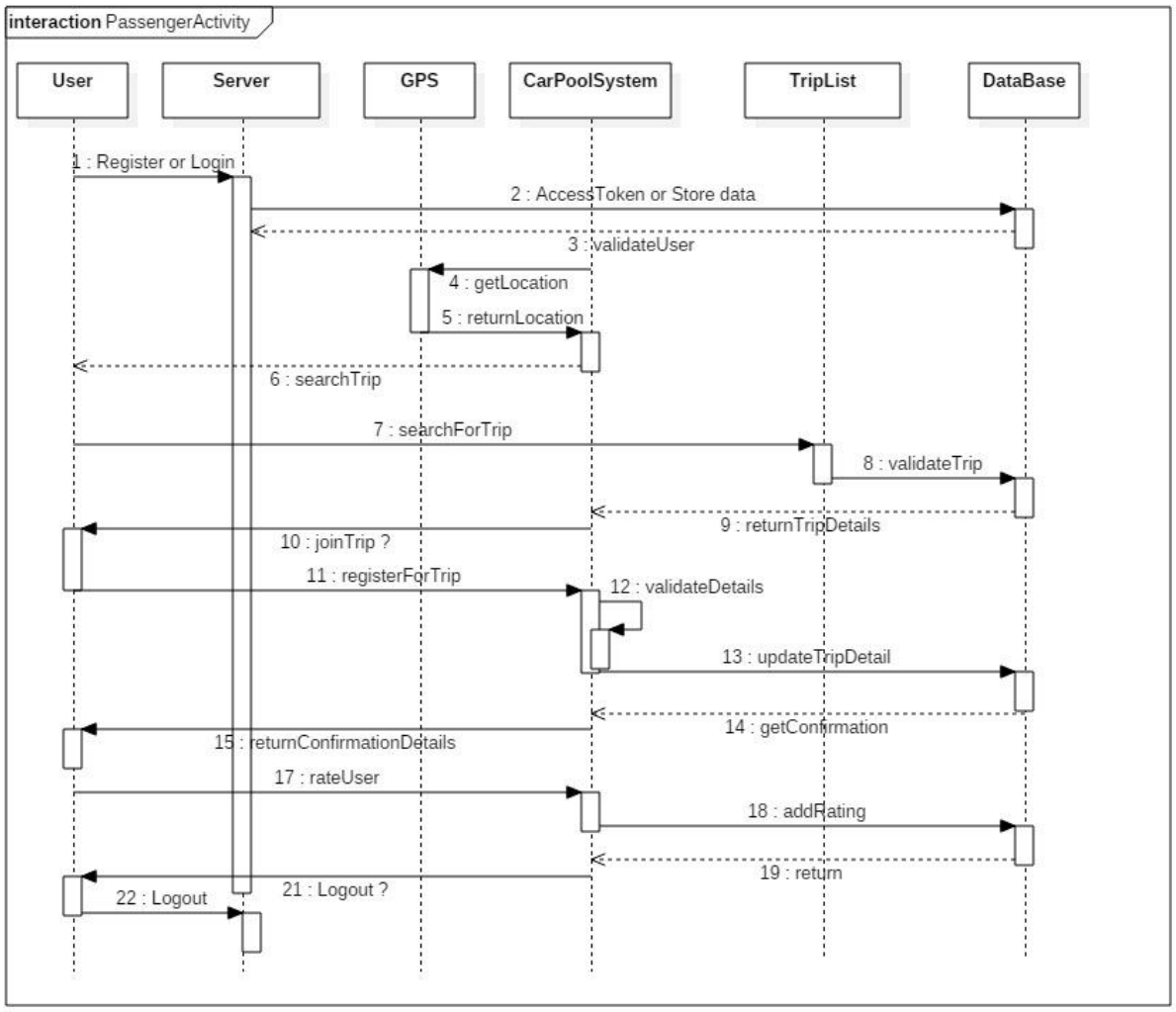


ER Diagram:

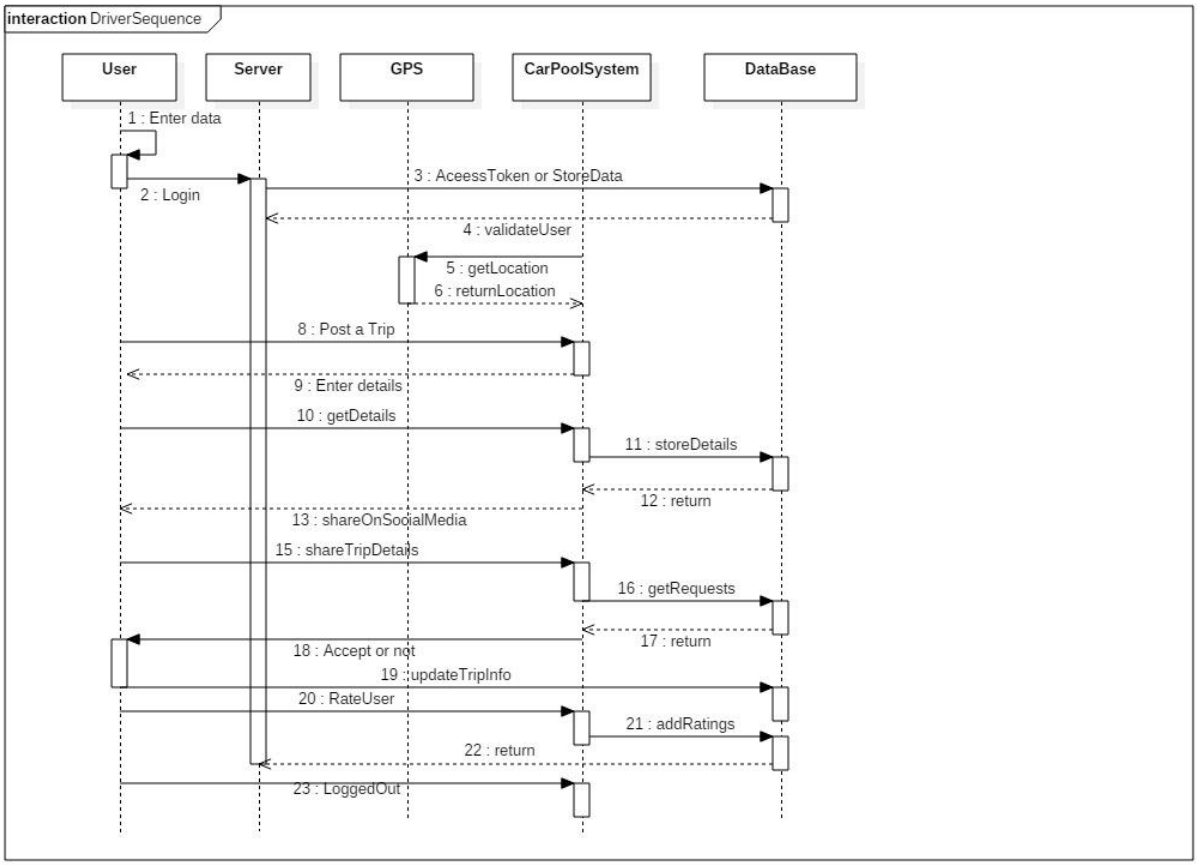


Sequence Diagram:

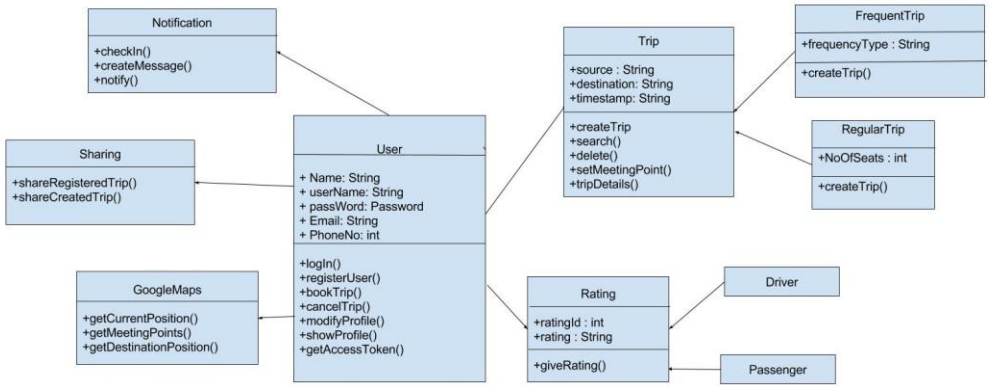
1) Passenger sequence Diagram


interaction PassengerActivity
User
Server
GPS
CarPoolSystem
TripList
DataBase
1 : Register or Login
2 : AccessToken or Store data
3 : validateUser
4 : getLocation
5 : returnLocation
6 : searchTrip
7 : searchForTrip
8 : validateTrip
9 : returnTripDetails
10 : joinTrip ?
11 : registerForTrip
12 : validateDetails
13 : updateTripDetail
14 : getConfirmation
15 : returnConfirmationDetails
17 : rateUser
18 : addRating
19 : return
21 : Logout ?
22 : Logout

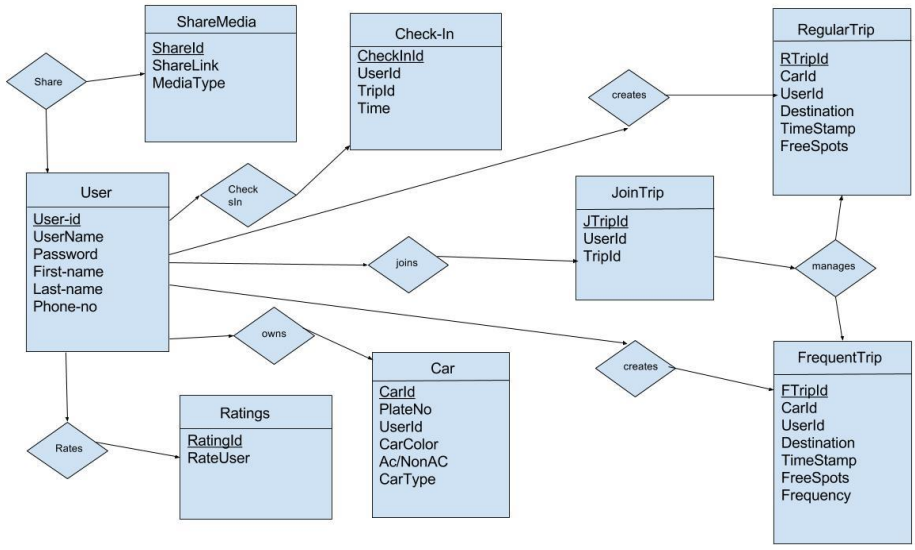
2) Driver sequence diagram:



Class Diagram:


Notification
+checkIn()
+createMessage()
+notify()
Trip
+source : String
+destination: String
+timestamp: String
+createTrip
+search()
+delete()
+setMeetingPoint()
+tripDetails()
FrequentTrip
+frequencyType : String
+createTrip()
RegularTrip
+NoOfSeats : int
+createTrip()
Sharing
+shareRegisteredTrip()
+shareCreatedTrip()
User
+ Name: String
+ userName: String
+ passWord: Password
+ Email: String
+ PhoneNo: int
+logIn()
+registerUser()
+bookTrip()
+cancelTrip()
+modifyProfile()
+showProfile()
+getAccessToken()
GoogleMaps
+getCurrentPosition()
+getMeetingPoints()
+getDestinationPosition()
Rating
+ratingId : int
+rating : String
+giveRating()
Driver
Passenger

DB Schema:


ShareMedia
ShareId
ShareLink
MediaType
Check-In
CheckInId
UserId
TripId
Time
RegularTrip
RTripId
CarId
UserId
Destination
TimeStamp
FreeSpots
Share
creates
User
User-id
UserName
Password
First-name
Last-name
Phone-no
Check sIn
JoinTrip
JTripId
UserId
TripId
joins
manages
owns
creates
Car
CarId
PlateNo
UserId
CarColor
Ac/NonAC
CarType
FrequentTrip
FTripId
CarId
UserId
Destination
TimeStamp
FreeSpots
Frequency
Rates
Ratings
RatingId
RateUser