UBER API DOCUMENT

Team No:13

Team Members:
Aparna Kulkarni
Tanvi Shukla
Mounika Muga
Preeti Patil
Hema Begur
Jasdeepsingh Oberoi

API List

Driver Module/Service:

1. **Create a new Driver**

   <u>Function name with inputs:</u>
   function **createDriver(driverID ,firstName ,lastName ,address ,city ,state,zipCode,phoneNumber,email, carDetails, driverImage, Intro_Video)**

   <u>Description:</u>
   Store the Driver details in the database,redirect the user to the HomePage after successful registration.

2. **Delete an existing Driver·**

   <u>Function name with inputs:</u>
   function **deleteDriver()**

   <u>Description:</u>
   Deletes the Driver details from the database.

3. **List all Drivers known by the system**

   <u>Function name with inputs:</u>
   function **listAllDrivers()**

   <u>Description:</u>
   Display the list of all Drivers in the database.

4. **Change a driver's information**

   <u>Function name with inputs:</u>
   function **updateDriverInfo(driverID ,firstName ,lastName ,address ,city, state, zipCode, phoneNumber, email, carDetails,driveImage, introVideo)**

   <u>Description:</u>
   Updates the Driver details in the database.This function will first display the driver details and then updates whatever information driver wants to update.

5. **Search for a driver based on attributes**

   <u>Function name with inputs:</u>

function **searchDriver(driverID, firstName ,lastName ,address ,city ,state, zipCode, phoneNumber, email, carDetails, rating, reviews, driverImage, introVideo, ridesHistory)**

Description:
Driver can be searched using any of the above attributes.Searches and display Driver details from the database.If there are multiple search results the system will display the list for the given criteria.

6.   Display information about a driver

Function name with inputs:
function **driverInfo**()

Description:
This function will be called on a button click where a query will be fired, that will fetch the driver information from the database and display it.

7.   Introduction video of the driver:

Function name with inputs:
function **driverIntroduction ()**

Description:
This function will be called when a user tends to see driver introduction , thus a query will be fired, that will fetch the introduction video of driver from the database and display it.

Customer Module/Service:
8.   Create a new Customer:

Function name with inputs:
function **registerCustomer(custID, firstName, lastName, address, city, state, country, emailId, password)**

Description:
Store the customer details in the database, maintain the registration time in the last login field, redirect the user to the HomePage.

9. Delete an existing Customer:
Function name with inputs:
function **deleteCustomer(custID)**

Description:
Function is called when a customer wants to delete account.

10. List all Customers known by the system:

Function name with inputs:
function **displayAllCustomer()**

Description:
Displays all information of all customers.

11. Generate Bill for a Customer (Every Ride).

Function name with inputs:
function **generate_bill(custID,rideID)**

Description:
This function creates a new bill for every customer for each of his/her ride with the Uber by taking into account the CustomerID. It calls the create_bill(rideId) function which calculates and generates the bill based on the distance of the ride and returns the calculated amount to the generate_bill() function which then assigns the bill amount to the customer.

12. Select driver within 10 miles from Customer location:

Function name with inputs:
function **get_driver(current_loc_cust)**

Description:
This function takes into consideration the current location of the user and shows him/her the location of drivers within 10 miles of distance from him/her.

13. Upload images about events during his ride:

Function name with inputs:
function **upload_image()**

Description:
This function takes the images of events occurred during the ride and saves it in the database. Later in the future these images can be accessed by the user/driver/admin for related purposes.

Billing Module/Service:

14. Create a new Bill for each ride:

Function name with inputs:
function **create_bill(rideID)**

Description:
This function calculates and generates the bill based on the distance of the ride and returns the calculated amount.

15. Delete an existing Bill:

Function name with inputs:
function **delete_bill(billID)**

Description:
This function takes the bill ID of the created bill and performs the delete function on the the row with the passed bill ID as the primary key from the Billing Information table.

16. Search an existing Bill:

Function name with inputs:
function **search_Bill(billID, custID)**

Description:
This function enables the user to search for an existing bill using bill ID and displays the bill information.

Admin Module/Service:

17. Add driver to the system:

Function with inputs:
function **add_Driver(driverName, driverID)**

Description:
This function adds the driver to the system.The parameters that we are accepting are name,driverid.We will keep track of all the drivers added,so that we when we get an uber request we should have a driver to provide the service.

18. Add customer to the system:

Function with inputs:
function **Add_Customer(driverName,custID)**

Description:
The functions add the customer to the system.The parameters that we are accepting are name and customer id.Customers will be added when they will place an uber request.

.

19. Review driver/customer account.·

Function with inputs:
function **Review_Account()**

Description:
 Here we review account and retrieve all the information about the driver or customer as required.

20. Show statistics (revenue/day) and total rides (area wise):

Function with inputs:
function **Show_Statistics(Date)**

Description:
This function will accept the date for which the revenue is to be shown and show the revenue for that day according to area.

21. Show graphs/charts for different rides per area, per driver, per customer:

Function with inputs:
function **showGraphs(asPerArea, areaAddress, custID, driverID)**

Description
This function shows results of rides in the form of charts or graphs based on parameters given. If results are requested per area, area address is passed as parameter, if the results are requested as per driver then driver ID is passed, if the results are to be retrieved as per a customer, customer ID is passed in the parameters.

22. Search for a Bill based on attributes:

Function with inputs:
function **searchBills(billID)**

Description:
This function is used to search bills by providing bill attribute billID as a its parameter. Administrator can use this function to search for bills.

23. Display information about a Bill:

Function with inputs:
function **displayBillInfo(billID)**

Description:
This function displays all the information related to a bill such as billing date, pickup and dropoff location, distance covered, cost of the ride, driver ID and customer ID. Administrator is the authorized user to access this function.

Rides Module/Service:
24. Create a new ride:

Function with inputs:
function **createRide(pickUpLoc, dropOffLoc, dateTime, custID)**

Description:
This function creates a new ride taking pick up location, drop off location, pick up time and customerID as its parameters.

25. Edit an existing ride:

Function with inputs:
function **editRide(rideID, newPickUpLoc, newDropOffLoc, newDateTime)**

Description:
This function looks for an existing ride with the given ride ID and changes the ride details if there's a change from the existing information.

26. Delete an existing ride:

Function with inputs:
function **deleteRide(rideID, custID)**

Description:
This function looks for an existing ride with the given ride ID and customer ID and deletes the scheduled ride.

27.List all rides of a customer:

Function with inputs:
function **allCustRides(custID)**

Description:
This function looks for all rides of a customer with the given customer ID and displays the results.

28.List all rides of a driver:

Function with inputs:
function **allDriverRides(driverID)**

Description:
This function looks for all rides of a driver with the given driver ID and displays
     the results.

29.Show rides statistics as per location:

Function with inputs:
funtion **rideStatistics(pickUpLoc,DropOffLoc)**

Description:
This function looks for all rides corresponding to a location with pickup and drop off locations and displays the ride statistics.

30.Display location of drivers within 10 miles of radius from customer location

Function with inputs:
funtion **driverNearBy(custID, driverID, pickUpLoc, driverLoc)**

Description:
This function looks for all the drivers whose location is within 10 miles radius from the pickup location of the customer and displays the results.

31.Login for all the users:

Function with inputs:
function **rideStatistics(userId, password)**
Description:This function provides authentication enables the user to log in to their account using username and password.

Database Schema:
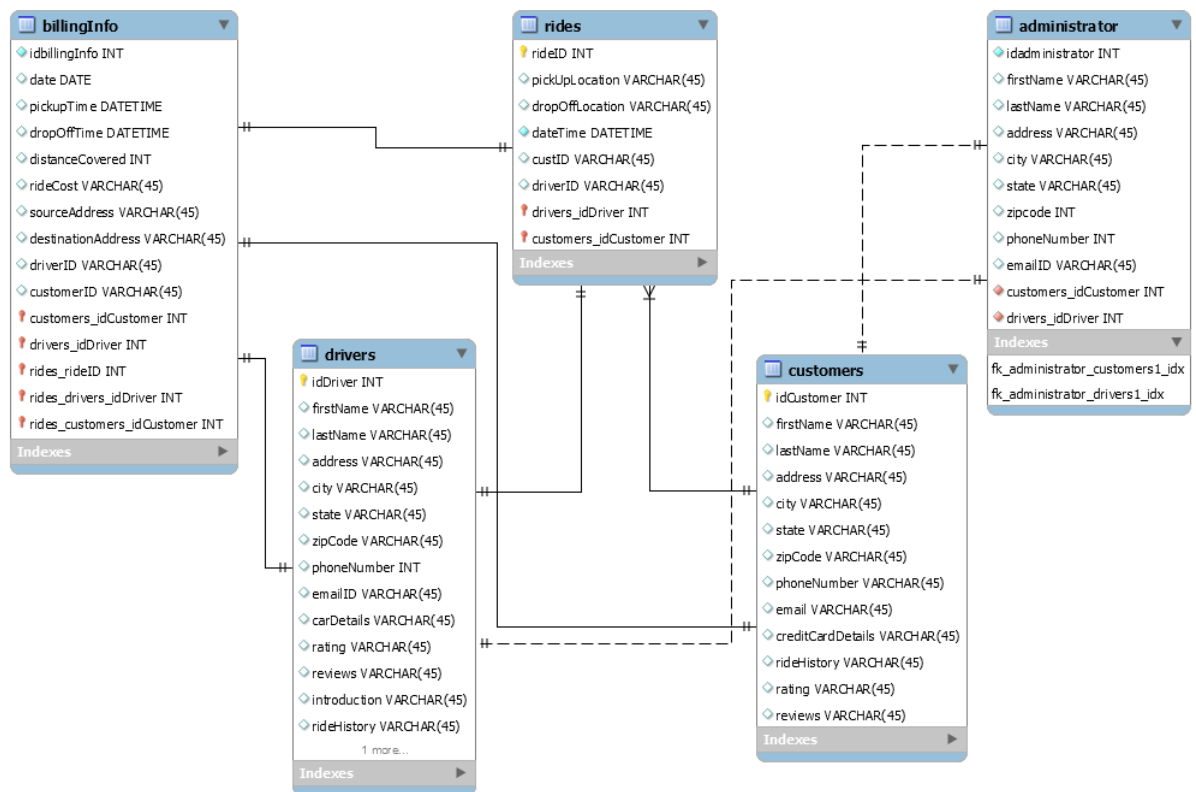
Relationships between Entities:

a. Customer can book a ride:
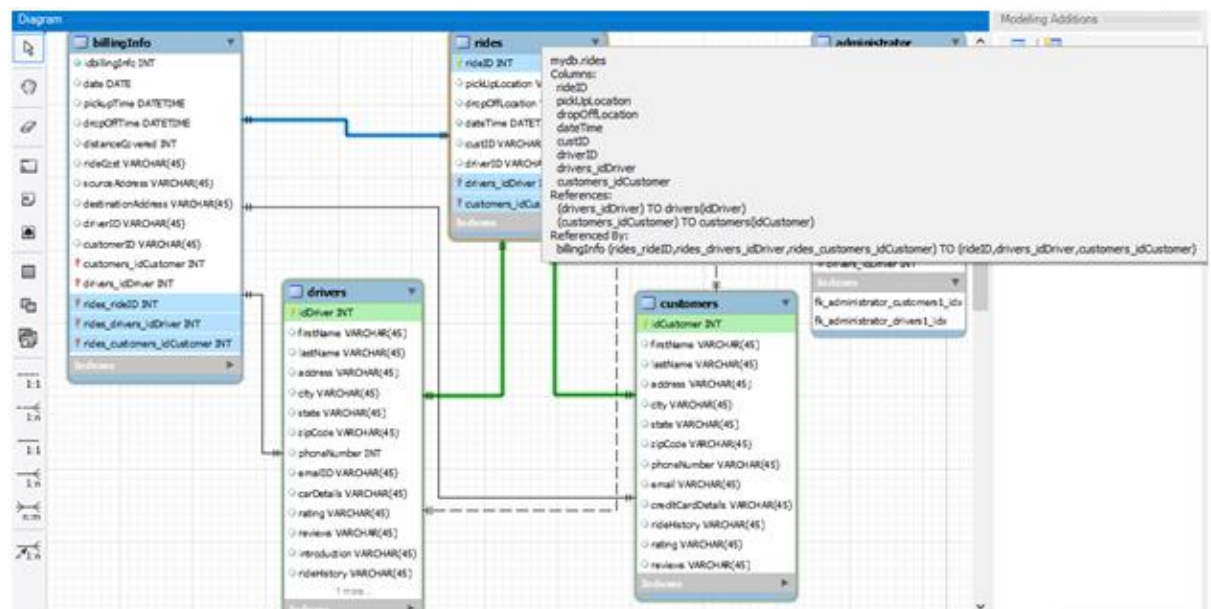
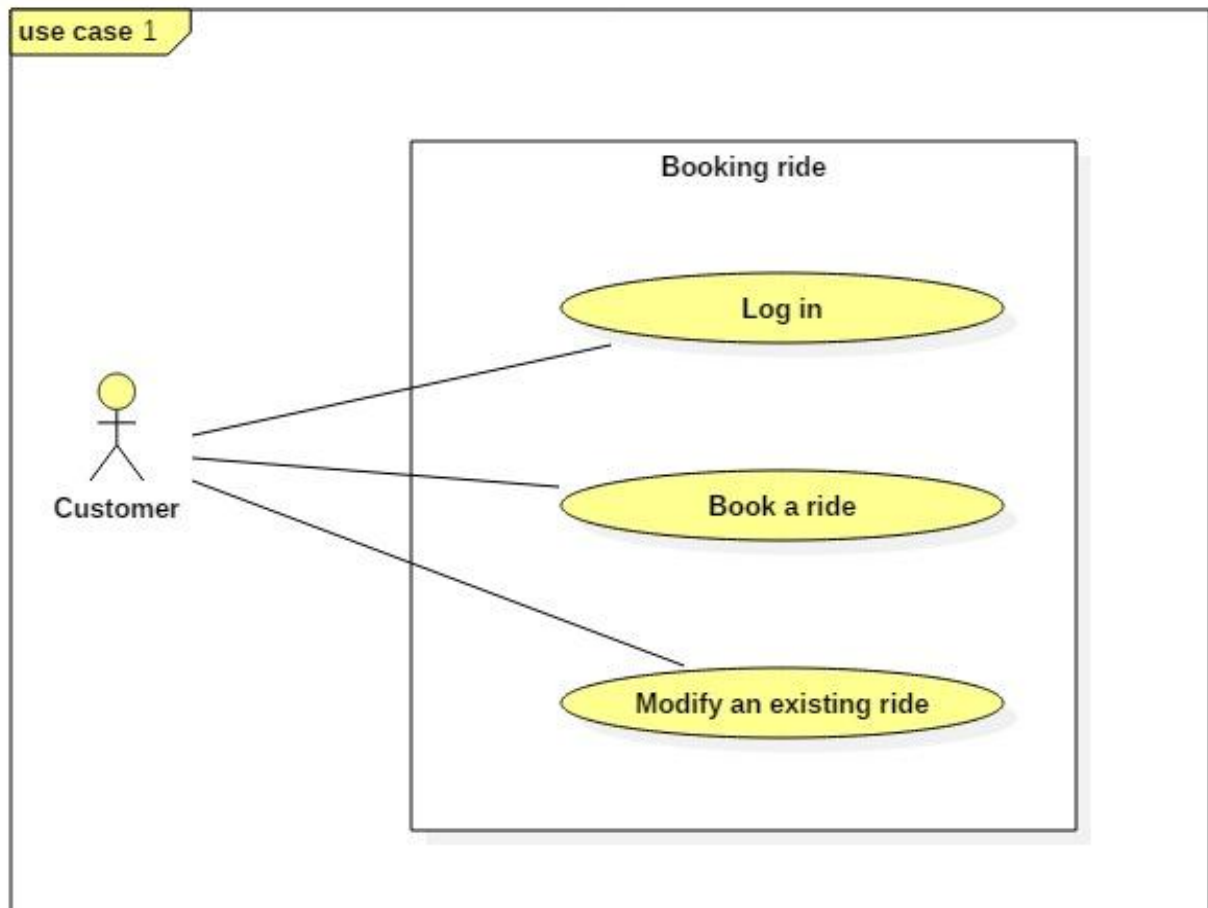### b. Admin can add customers or drivers:
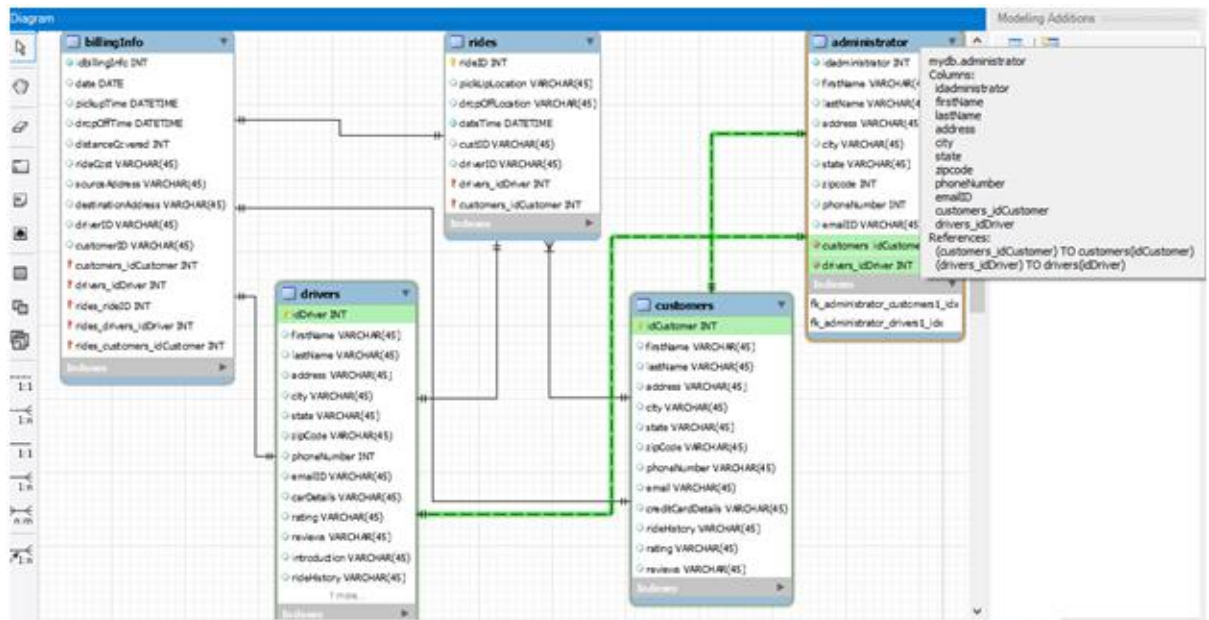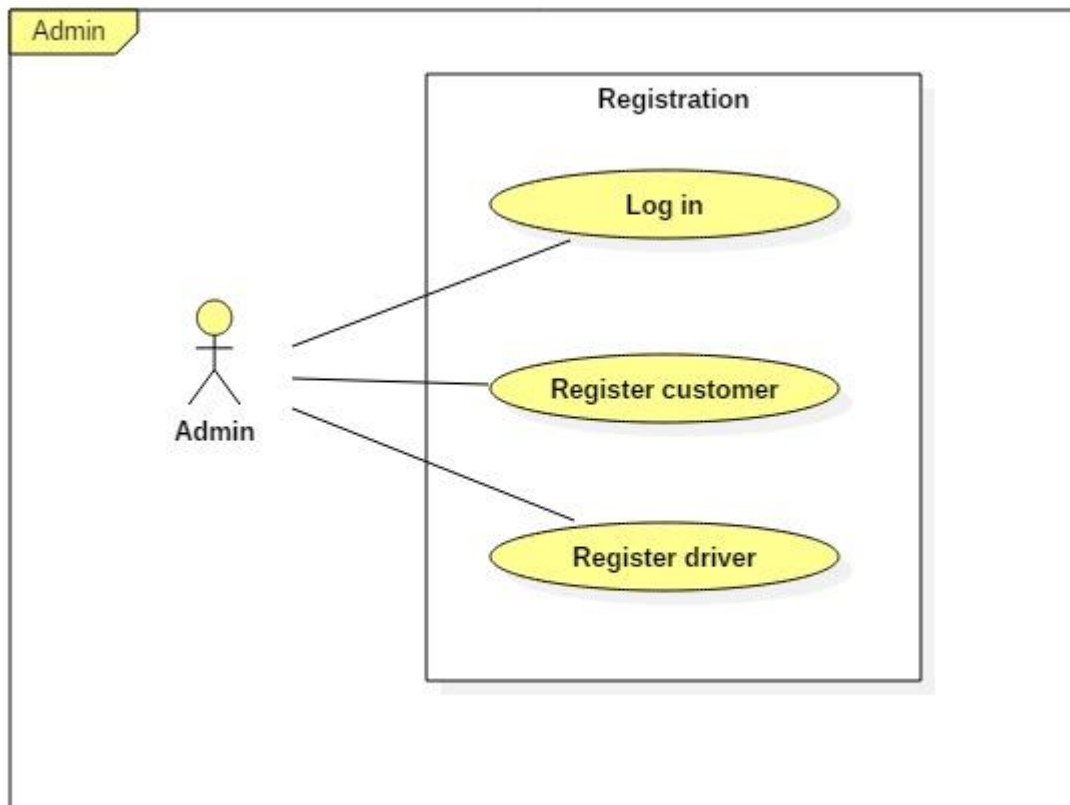
c. Billing Info for a Ride:
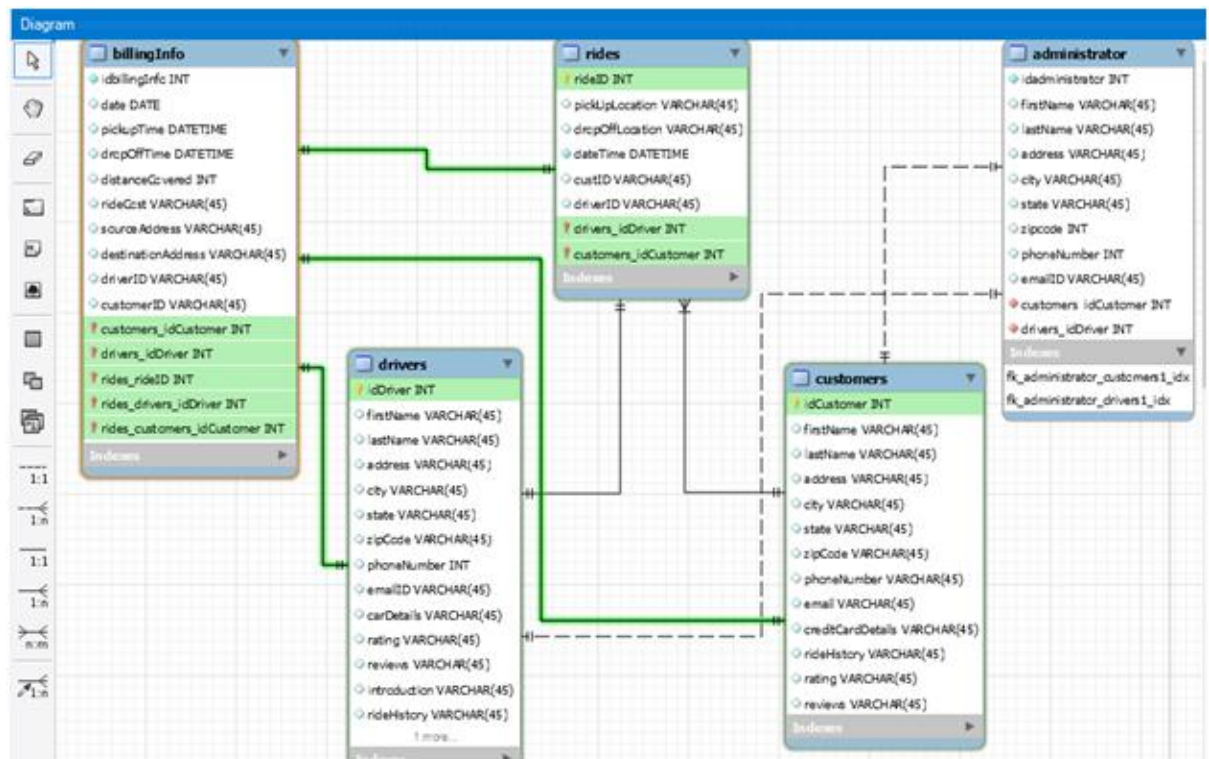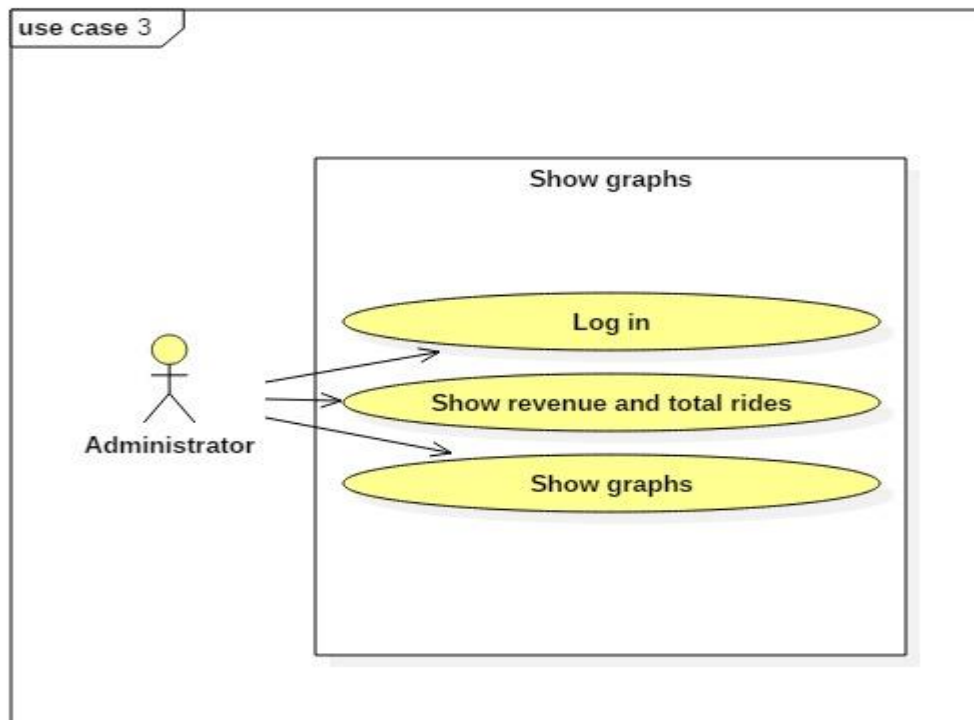
Use case diagrams:

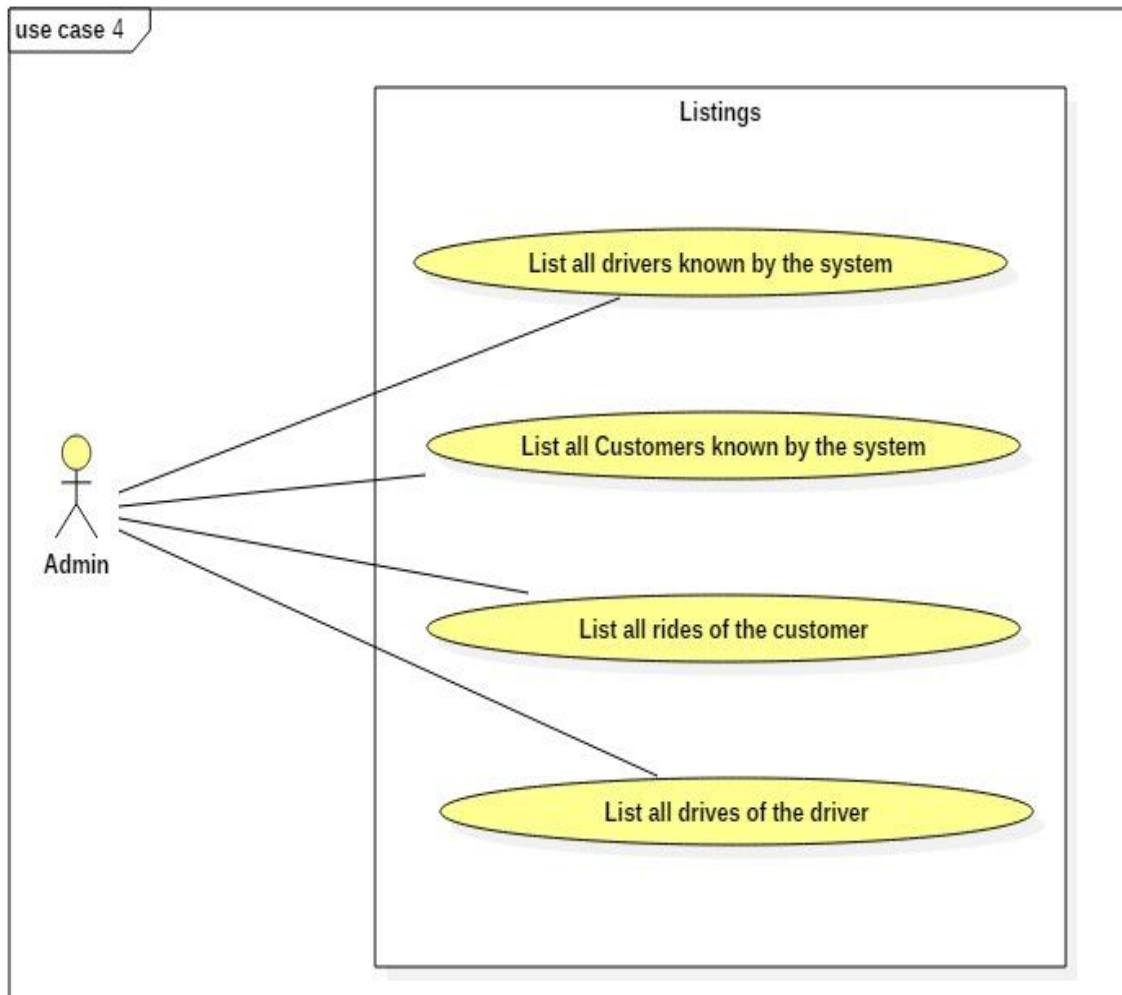1. A customer using Uber system can log in and book a ride. Also an existing ride can be modified.

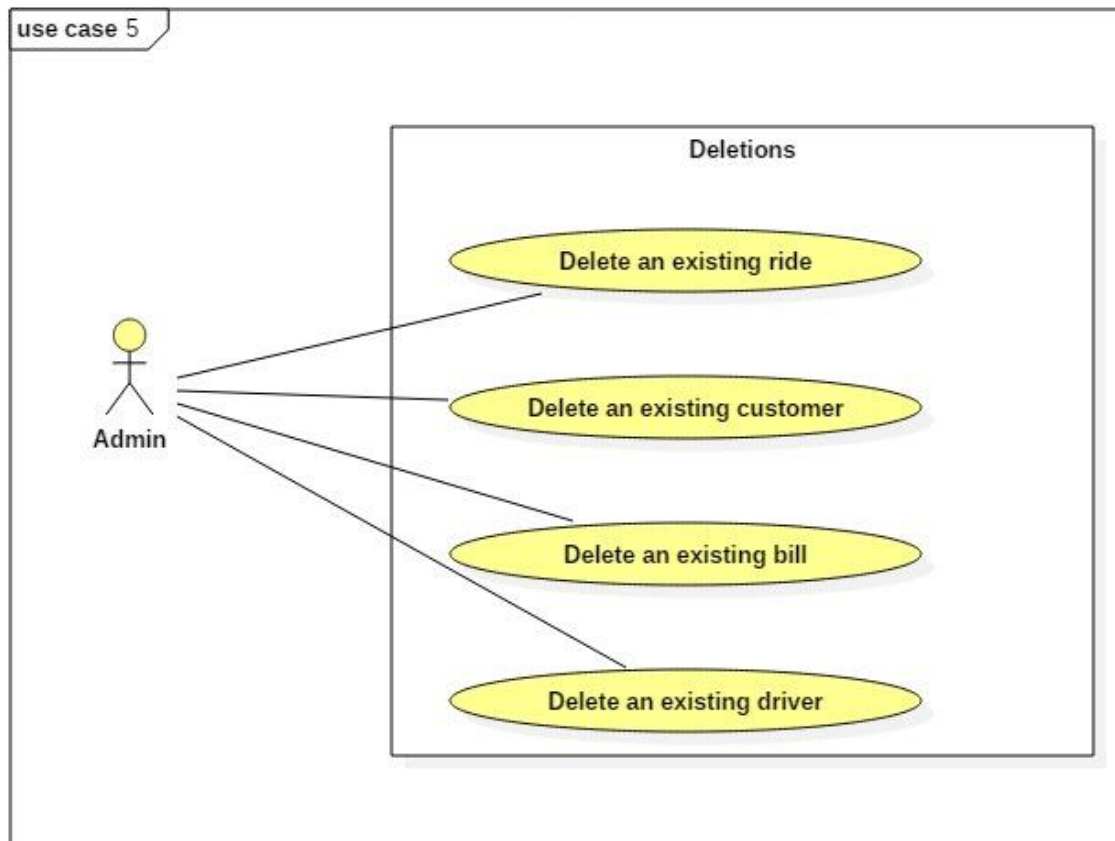**2.** Administrator accesses Uber system to register new drivers and customers.

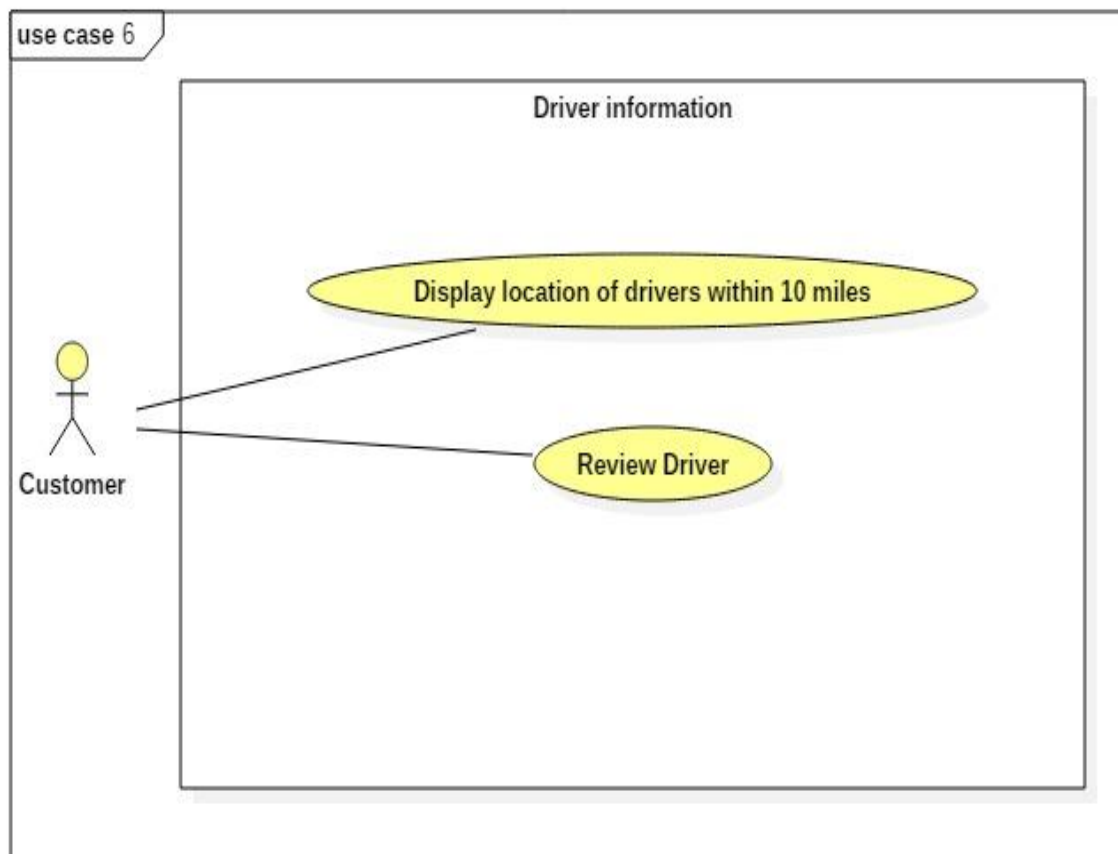**3**. Administrator can use the system to see the revenue per day and number of rides per area.

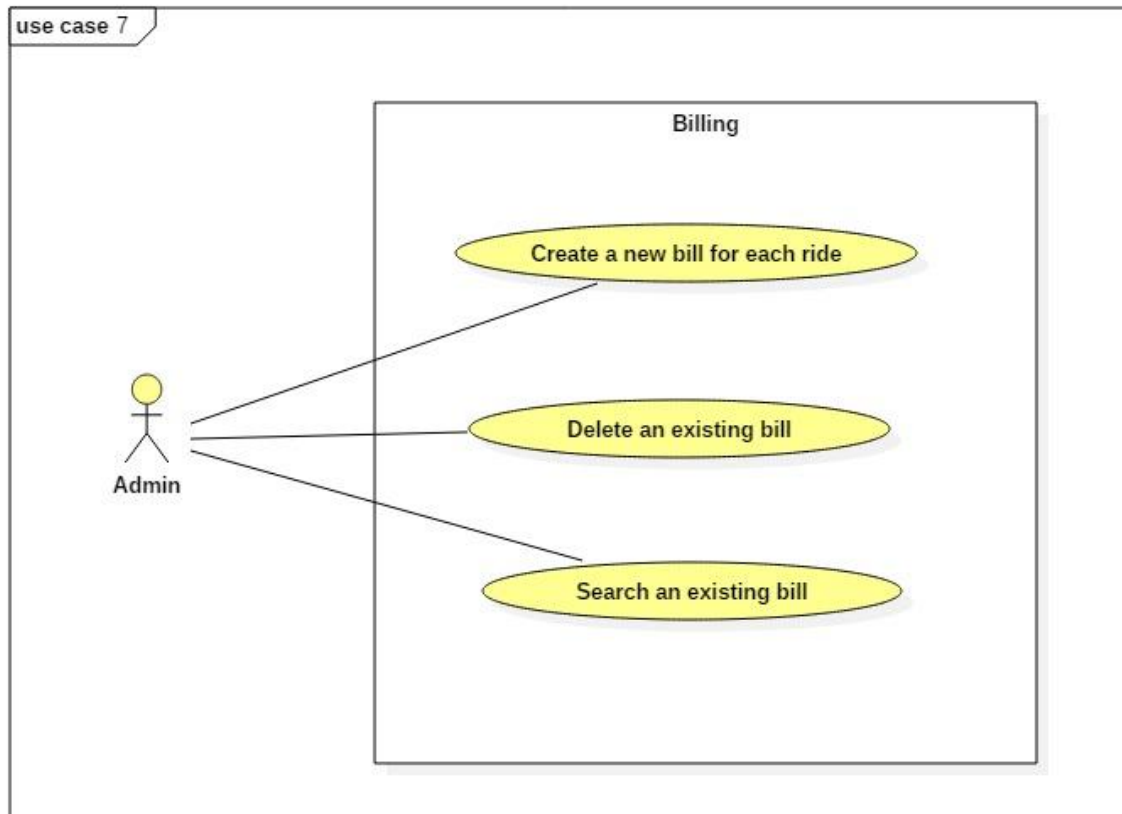**4:**Admin can list all drivers,customers,rides and drives of the system.

**5**:Admin can delete an existing bill,customer,driver and ride.


.

6:Display driver information

7:Admin can create a new bill,delete and even search existing bill.

use case 7

Billing

Create a new bill for each ride

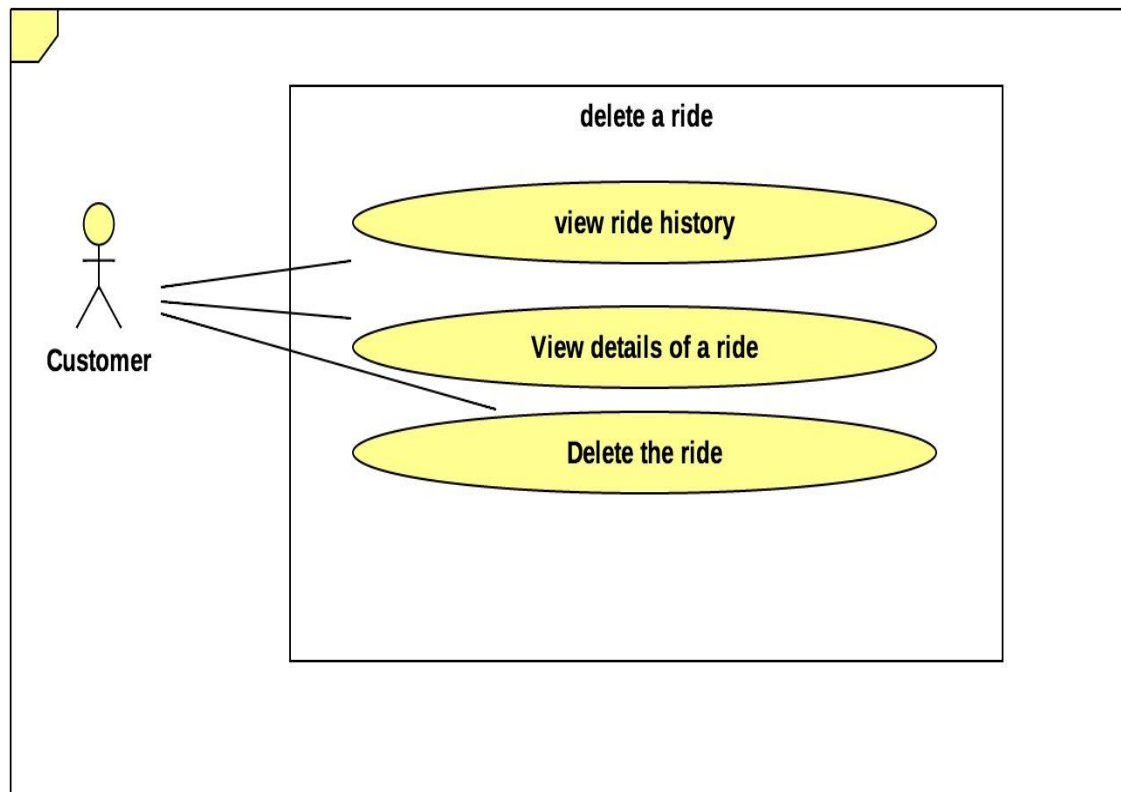Delete an existing bill

Admin

Search an existing bill

8. A customer can search for driver information and an admin can search for a bill. Following use case shows the relation between the se use cases and actors involved.
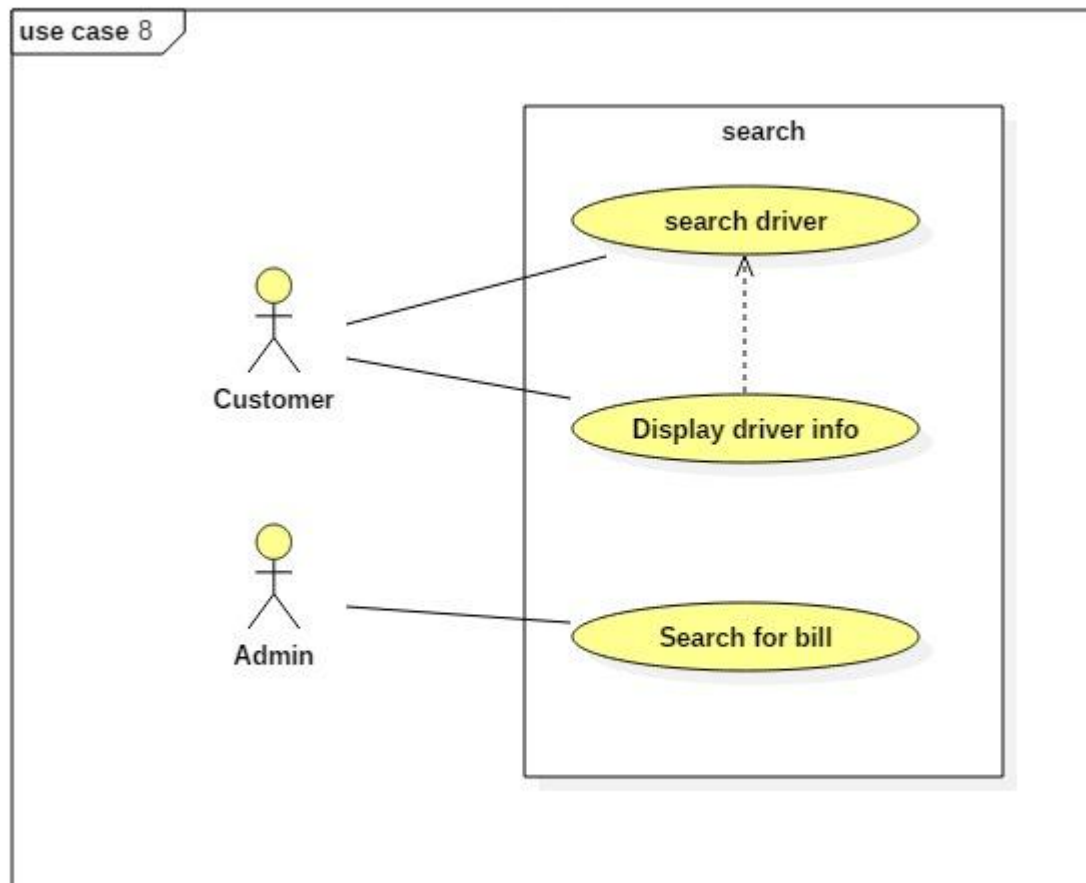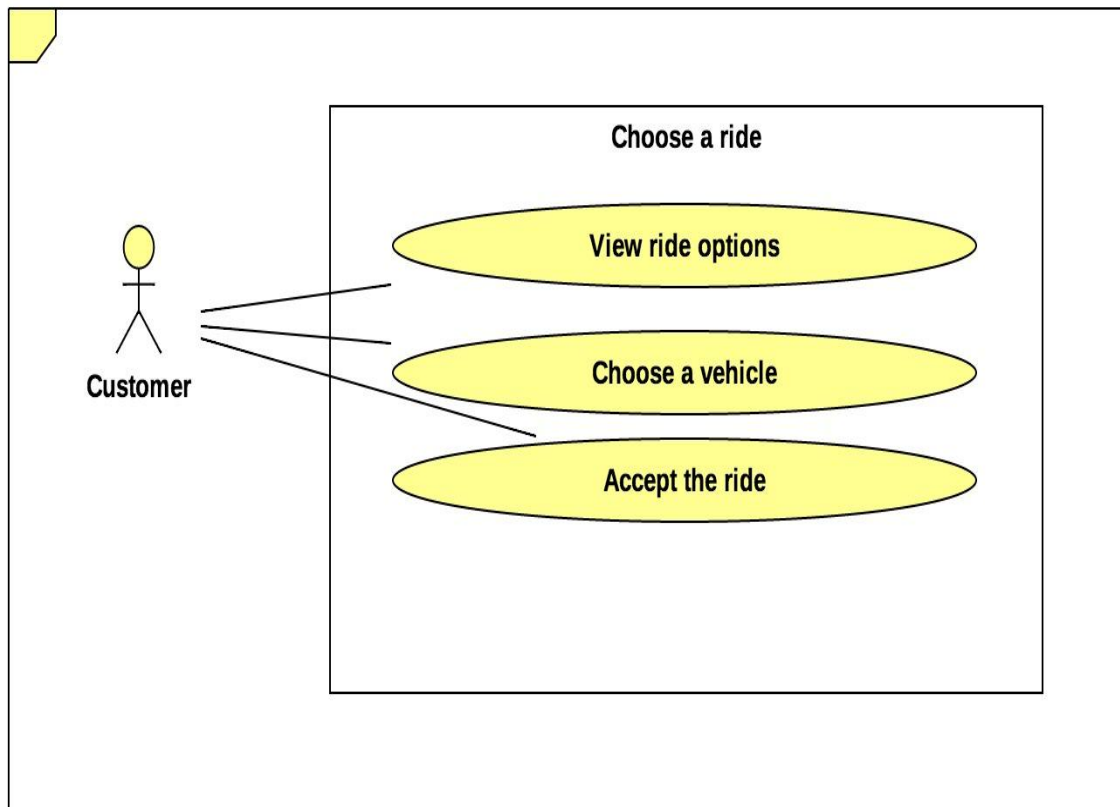
9. Customer can view ride history and review details of a ride.
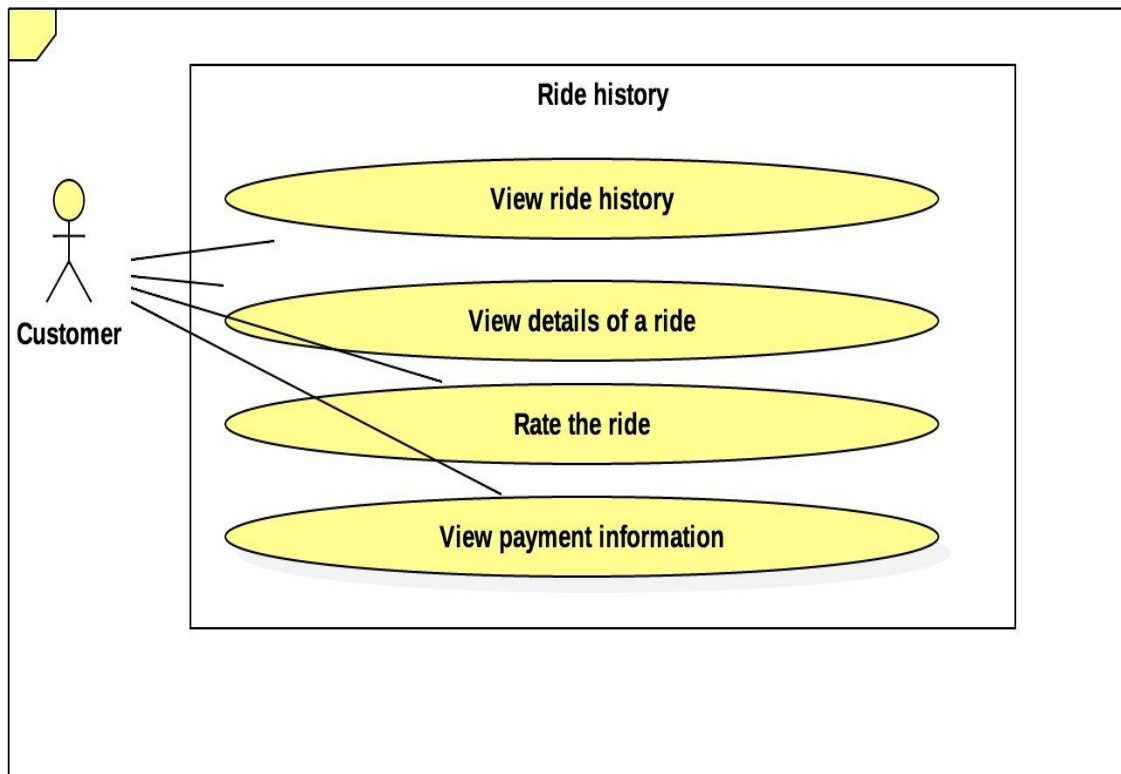
10. A customer can also delete an existing ride from ride history.

11. Customer can choose a ride and type of vehicle among the available rides in 10 miles radius from the pickup point.

12. Customer can cancel the ride