Capstone Project
Cab Booking application

1   ROAD MAP

Application will be developed in following

1. Spring Boot + Rest Controller + Spring Data JPA

2   PROBLEM STATEMENT

To Create an Application for Online Cab Booking

Customer should be able to view the list of available cabs.

Customer should be able to book cab for desired location from a certain location.

Admin should be able to perform below operations:

- admin can add/update/delete/view the cabs

- admin can add/update/delete/view the driver

- admin can view and cancel the bookings

- admin can add/update/delete/view the customer

Customer should be able to perform below operations:

- Customer can register
- Customer can update their profile
- Customer can view all the available cabs
- Customer can book and cancel the booking
- Customer can view the booking history.

Customer should be able to perform below operations:

- Driver can view the booking details
- Driver can view the booking history

3   MODULES

User Module

Driver Management Module

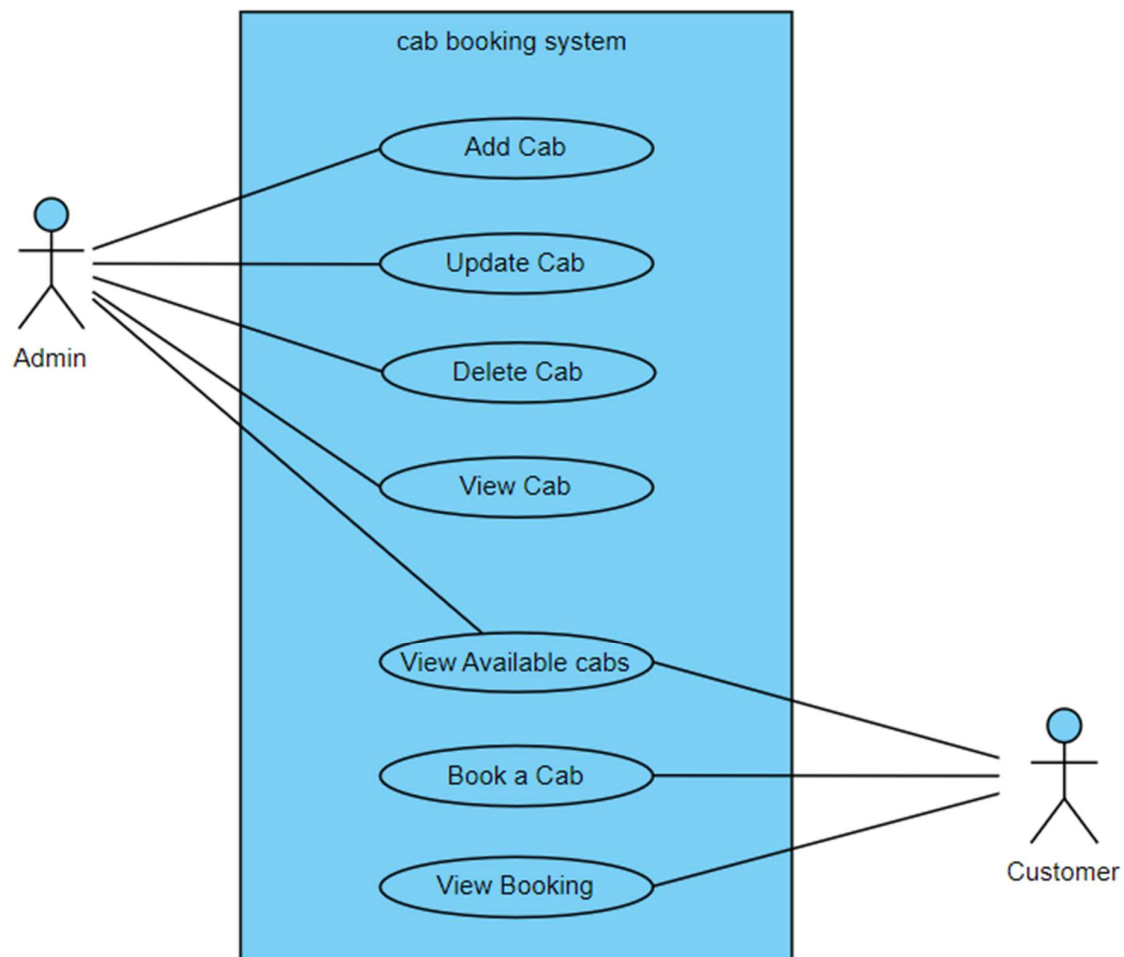Cab Management Module

Booking Management Module
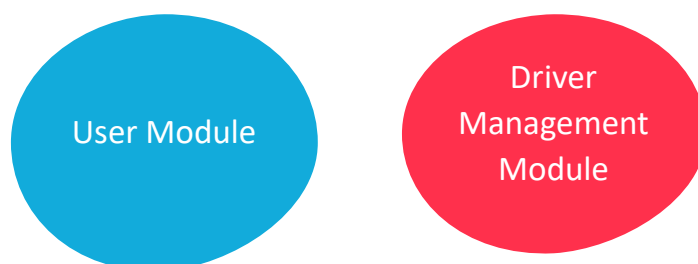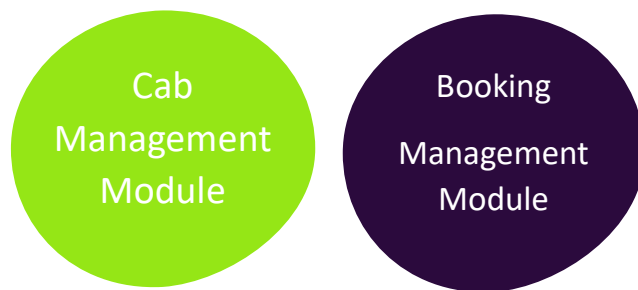
4 USE CASE DIAGRAM

Participants have to create use case diagram based on problem statement

For your reference sample use case is added.
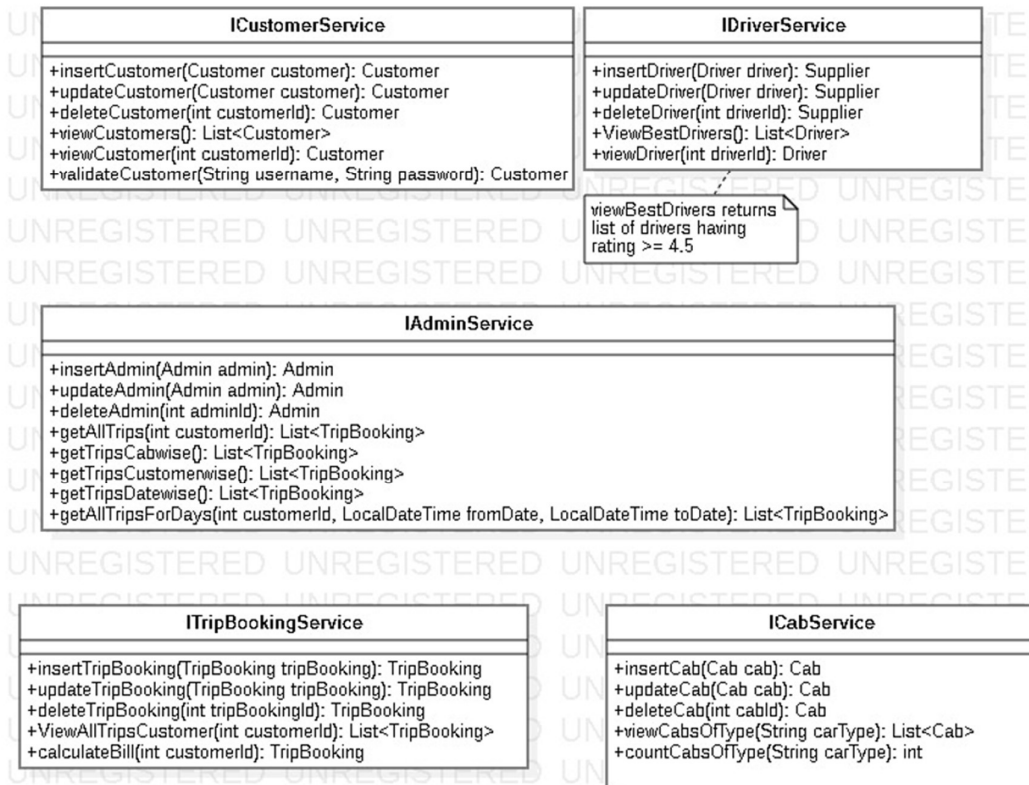
Class Design

POJO Classes:
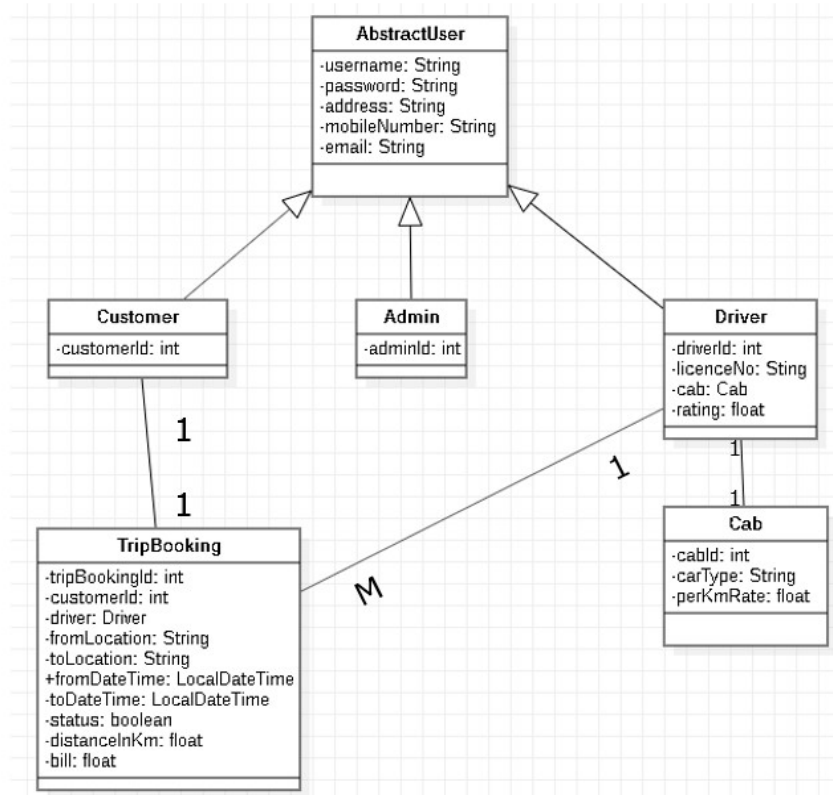
Class Design
Service Interface:

ICustomerService

+insertCustomer(Customer customer): Customer
+updateCustomer(Customer customer): Customer
+deleteCustomer(int customerId): Customer
+viewCustomers(): List<Customer>
+viewCustomer(int customerId): Customer
+validateCustomer(String username, String password): Customer

IDriverService

+insertDriver(Driver driver): Supplier
+updateDriver(Driver driver): Supplier
+deleteDriver(int driverId): Supplier
+ViewBestDrivers(): List<Driver>
+viewDriver(int driverId): Driver

viewBestDrivers returns list of drivers having rating >= 4.5

IAdminService

+insertAdmin(Admin admin): Admin
+updateAdmin(Admin admin): Admin
+deleteAdmin(int adminId): Admin
+getAllTrips(int customerId): List<TripBooking>
+getTripsCabwise(): List<TripBooking>
+getTripsCustomerwise(): List<TripBooking>
+getTripsDatewise(): List<TripBooking>
+getAllTripsForDays(int customerId, LocalDateTime fromDate, LocalDateTime toDate): List<TripBooking>

ITripBookingService

+insertTripBooking(TripBooking tripBooking): TripBooking
+updateTripBooking(TripBooking tripBooking): TripBooking
+deleteTripBooking(int tripBookingId): TripBooking
+ViewAllTripsCustomer(int customerId): List<TripBooking>
+calculateBill(int customerId): TripBooking

ICabService

+insertCab(Cab cab): Cab
+updateCab(Cab cab): Cab
+deleteCab(int cabId): Cab
+viewCabsOfType(String carType): List<Cab>
+countCabsOfType(String carType): int

6   PROJECT DELIVERABLES

1. User stories based on the problem statement.
2. Use case Diagrams
3. Source code
4. Junit Testcases

Note:

You can feel free to add new features and create or update design diagrams if it is required.