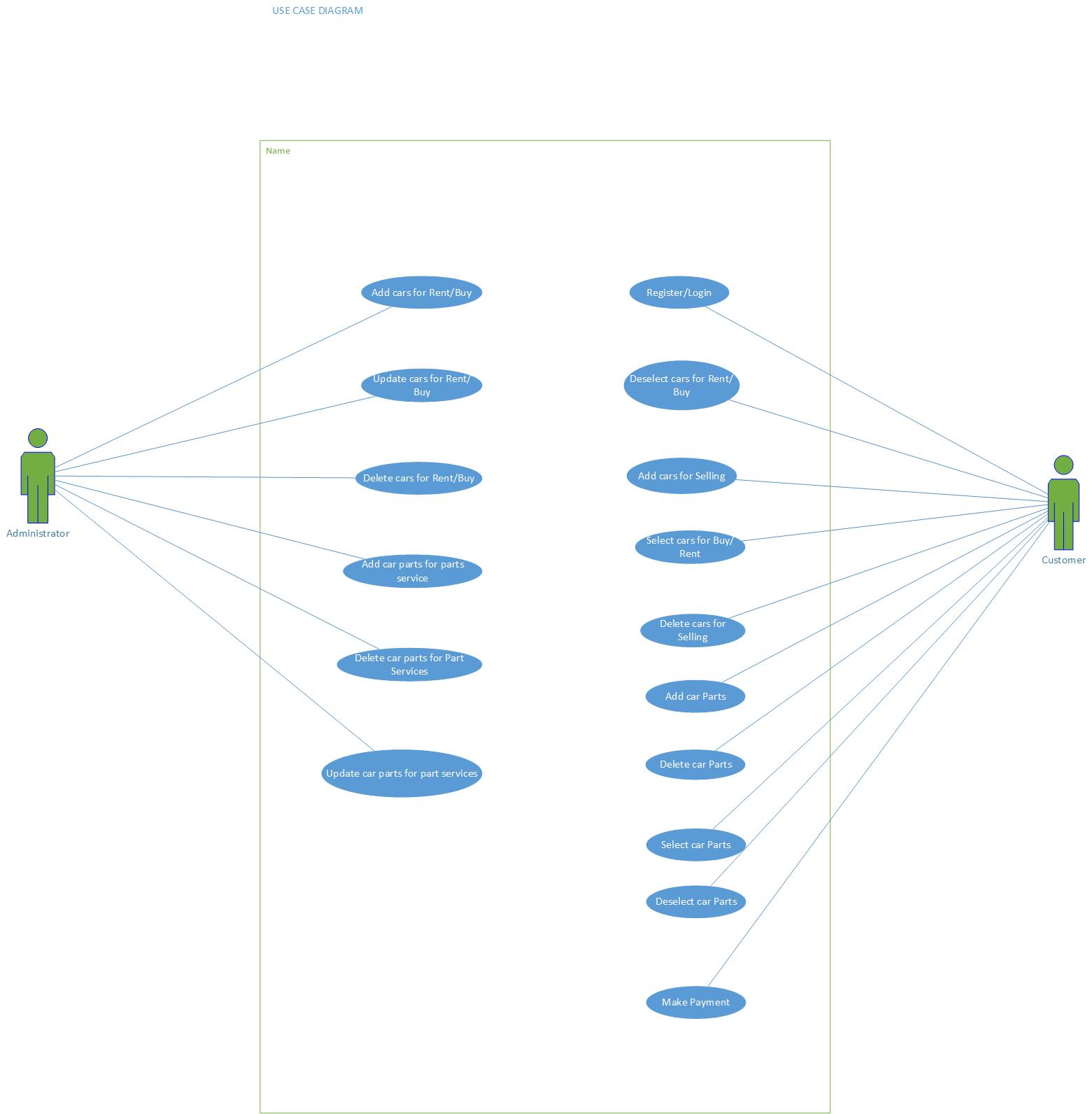
PROJECT CAR REQUEST SERVICES – PHASE1

**Requirements and Hight Level Design**

* 1. Requirement Specification Document (RSD)

UML Use Case diagram:



|  |  |  |  |
| --- | --- | --- | --- |
| **Req. ID** | **Description** | **Version** | **Priority** |
| 1 | Select the car from the given list | 1.0 | High |
| 2 | Check availability for the selected car | 1.0 | High |
| 3 | Select rental period if renting a car | 1.0 | Medium |
| 4 | Booking the selected vehicle | 1.0 | Medium |
| 5 | Make payment transaction | 1.0 | High |
| 6 | Order summary | 1.0 | Medium |
| 7 | Select car to buy | 1.0 | High |
| 8 | Place order and make the payment | 1.0 | High |
| 9 | Give the car details to sell and submit | 1.0 | High |
| 10 | Select part to buy | 1.0 | High |
| 11 | Place order and make the payment | 1.0 | High |

Requirement Table:

1.2 Design Phase

MVC Architecture:

Model:

* CustomerDetails
* RentalCarDetails
* DealersCarDetails
* PartsDetails
* OrderDetails
* TransactionDetails

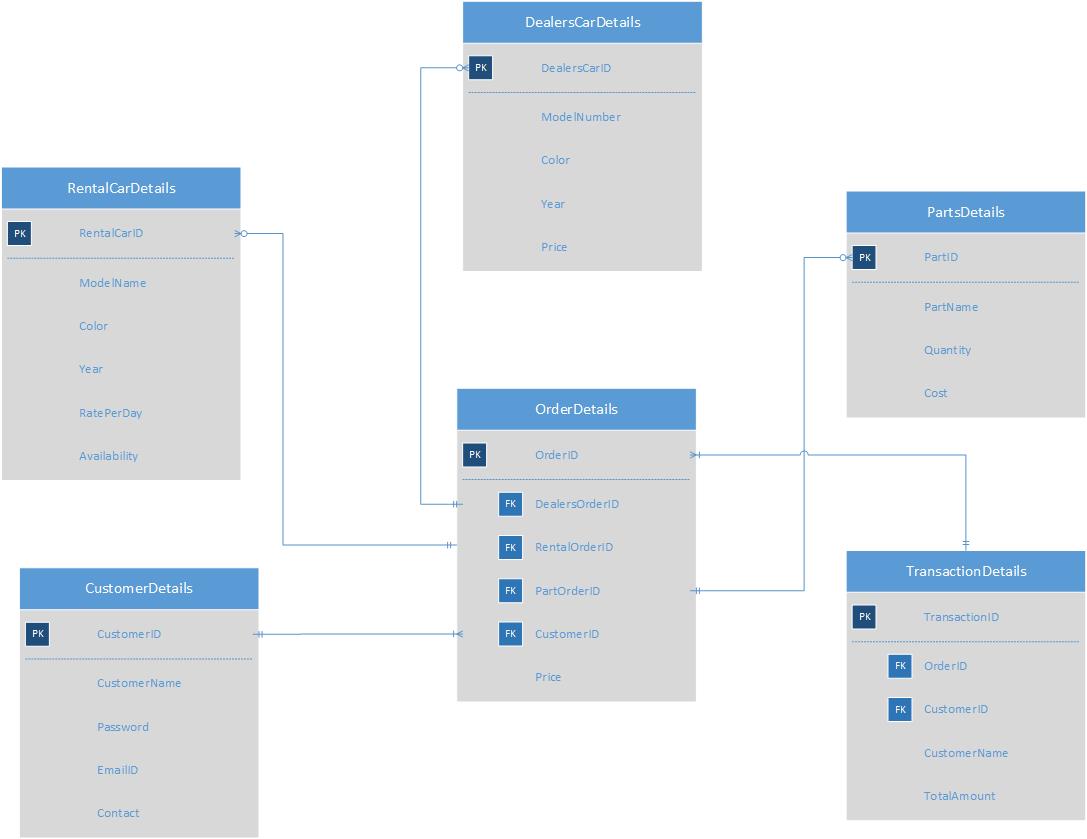
View:

* Main Page to select User (Admin/Customer)
* Admin Login page
* Customer Login Page
* Admin Home Page
  + RentalCar
  + DealersCar
  + PartsServices
  + ViewTransactions
* Customer Home Page
* RentCar
* DealCar
* PartServices
* ViewOrder
* Admin Rental Car
* AddCar
* DeleteCar
* UpdateCar
* Admin Dealers Car
* AddCar
* DeleteCar
* Admin Parts Services
* AddParts
* DeleteParts
* UpdateParts
* Admin View Transactions
* View all transactions till date
* Customer Rent Car
* Select car
* Customer Dealers Car
* Sell Car from list
* Sell a car
* Customer Parts Services
* Select Parts
* Sell Parts
* Customer Order
* Make Payment

Controller:

* Main controller
* Admin login page controller
* Customer login page controller
* Customer Registration controller
* Admin home page controller
* Customer home page controller
* Admin Rental car controller
* Admin Dealers car controller
* Admin Parts details controller
* Customer Rental car controller
* Customer dealer’s car controller
* Customer parts car controller
* Customer View orders controller
* Admin View Transaction controller
* Place orders controller
* Admin Add car controller
* Admin Update car controller
* Customer Add car controller
* Customer delete car controller

Entity Relationship Diagram:



SQL Database queries:

1. CustomerDetails table:

CREATE TABLE customerDetails (customerID number(10) not null primary key,

CustomerName varchar(20) not null,

Password varchar(20) not null,

emailID varchar(30) not null,

contact number not null);

1. RentalCarDetails table:

CREATE TABLE RentalCarDetails (RentalCarID number(10) not null primary key,

ModelName varchar(20) not null,

Color varchar(20) not null,

year number(4) not null,

RatePerDay decimal not null,

Availability Boolean not null);

1. DealersCarDetails table:

CREATE TABLE DealersCarDetails (DealersCarID number(10) not null primary key,

ModelName varchar(20) not null,

Color varchar(20) not null,

year number(4) not null,

Price decimal not null);

1. PartsDetails table:

CREATE TABLE PartsDetails (PartID number(10) not null primary key,

PartName varchar(20) not null,

Quantity number(4) not null,

Cost decimal not null);

1. OrderDetails table:

CREATE TABLE OrderDetails (OrderID number(10) not null primary key,

DealersOrderID number(10) ,

RentalOrderID number(10) ,

PartOrderID number(10) ,

CustomerID number(10) not null,

Price decimal not null,

FOREIGN KEY (DealersOrderID) REFERENCES DealersCarDetails(DealersOrderID),

FOREIGN KEY (RentalOrderID) REFERENCES RentalCarDetails(RentalOrderID) ,

FOREIGN KEY (PartOrderID) REFERENCES PartsDetails(PartOrderID),

FOREIGN KEY (CustomerID) REFERENCES CustomerDetails(CustomerID));

1. TransactionDetails table:

CREATE TABLE TransactionDetails (TransactionID number(10) not null primary key,

OrderID number(10) not null,

CustomerID number(10) not null,

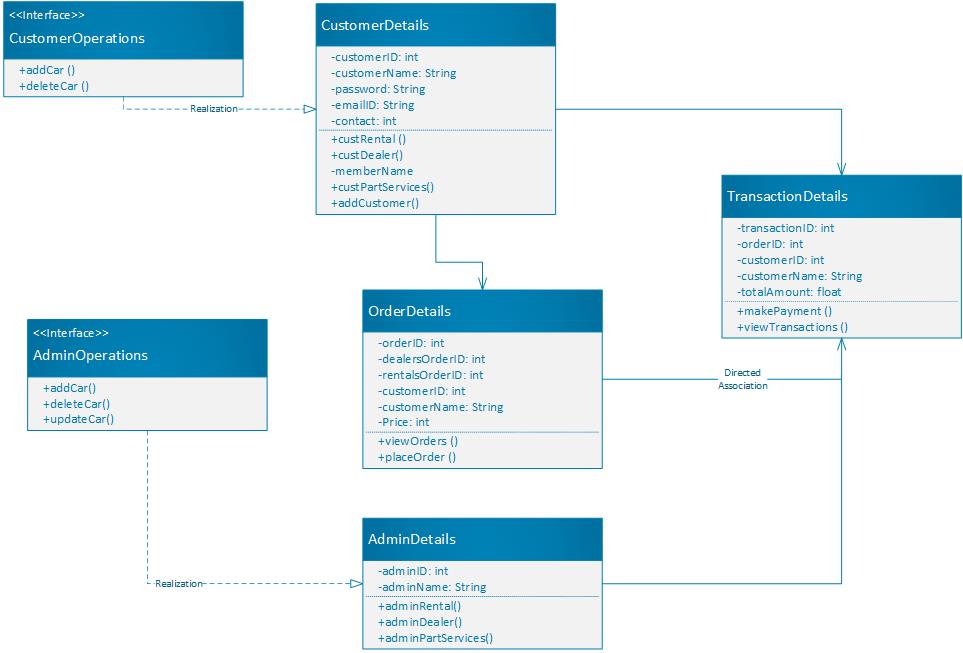
CustomerName varchar(20) not null,

TotalAmount decimal not null,

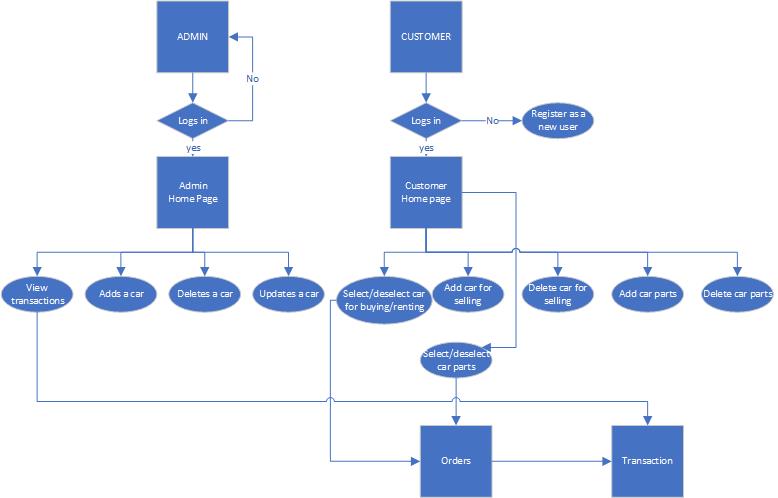
FOREIGN KEY (OrderID) REFERENCES OrderDetails(OrderID),

FOREIGN KEY (CustomerID) REFERENCES CustomerDetails(CustomerID));

UML Class Diagram:



High Level Architecture Diagram:



1.3 Web services API:

1. ArrayList<String>getCustomerByStreet(String street);
2. ArrayList<String>getPartsCount(int parts);
3. ArrayList<String>getDealersCarByPrice(int DealersCarID);
4. ArrayList<String>getRentalCarByPrice(int RentalCarID);
5. ArrayList<String>getPartsCarByPrice(int PartID);
6. ArrayList<String>getAllCustomerNames(String CustomerName);
7. ArrayList<String>getFrequentlyRentedCars(String ModelName);
8. ArrayList<String>getLastTransactionID(int TransactionID);
9. ArrayList<String>getRentalCarCount(int rental);
10. ArrayList<String> getDealersCarCount (int dealer);