

**CAR RENTAL**

**Software Design Document**

– Hanoi, February 2023 –

***Thành viên trong nhóm***

***Lê Trung Đức - SE05390***

***Nguyễn Trí Trường Sơn - HE151167***

***Nguyễn Hữu Minh Đức - HE151166***

[**I. Overview 4**](#_heading=h.gjdgxs)

[1. Code Packages/Namespaces 4](#_heading=h.30j0zll)

[2. Database Design 4](#_heading=h.1fob9te)

[1. Users 6](#_heading=h.2et92p0)

[2. Role 6](#_heading=h.tyjcwt)

[3. Customers 6](#_heading=h.2oyk2p5omlk2)

[4. Rentals 7](#_heading=h.1tcrznj7pvc)

[5. Cars 7](#_heading=h.f1tpn5h3i41v)

[6. Categories 7](#_heading=h.8ryhh8kswrd9)

[7. Transactions 7](#_heading=h.xgxypqgbc3s)

[**II. Code Designs 8**](#_heading=)

[1. Login 8](#_heading=h.ezqf6iijjs6d)

[a. Class Diagram 8](#_heading=h.upuk90fde7y6)

[b. Class Specifications 9](#_heading=h.oqnd66dwap84)

[XYZ Class 9](#_heading=h.vroca7x7cgr0)

[c. Sequence Diagram(s) 9](#_heading=h.xepmg9cwngpx)

[d. Database queries 10](#_heading=h.rg0ly5pnsbqz)

[2. User Register 10](#_heading=h.xfy9vmo9oofj)

[a. Class Diagram 11](#_heading=h.fbzdq7csuxul)

[b. Class Specifications 11](#_heading=h.63fw2fluk7f4)

[XYZ Class 12](#_heading=h.xy3fveci8133)

[c. Sequence Diagram(s) 12](#_heading=h.bgogukwshj4)

[d. Database queries 13](#_heading=h.6k9uglocblnl)

[3. Reset Password 13](#_heading=h.yt2g0rwpbzoi)

[a. Class Diagram 14](#_heading=h.bc68nhyyj309)

[b. Class Specifications 14](#_heading=h.t0ys60y2n7kr)

[XYZ Class 15](#_heading=h.jwit61i3c57)

[c. Sequence Diagram(s) 15](#_heading=h.ya4vswwt4yie)

[d. Database queries 15](#_heading=h.bcbik0nt1wjm)

[4. Public/Car List 16](#_heading=h.8d7epojxy9ga)

[a. Class Diagram 16](#_heading=h.omt4owacggr4)

[b. Class Specifications 16](#_heading=h.gygscju7xd8t)

[XYZ Class 16](#_heading=h.z9dusl5m8puj)

[c. Sequence Diagram(s) 17](#_heading=h.54cu8mem5j0t)

[d. Database queries 17](#_heading=h.y803i1r2rta0)

[5. Public/Car Details 18](#_heading=h.ig47nmbi41wc)

[a. Class Diagram 18](#_heading=h.eod3gdhn6x23)

[b. Class Specifications 18](#_heading=h.iehk7k1x1so9)

[XYZ Class 18](#_heading=h.k7xbmijp2r2m)

[c. Sequence Diagram(s) 19](#_heading=h.eambmtnwvpq)

[d. Database queries 19](#_heading=h.1b4qmsby52sz)

[6. Public/Payment 20](#_heading=h.m1rkpecb9kj9)

[a. Class Diagram 20](#_heading=h.nknm254ayqv3)

[b. Class Specifications 20](#_heading=h.tvd84j4fhax)

[XYZ Class 20](#_heading=h.thv475sunc7e)

[c. Sequence Diagram(s) 21](#_heading=h.tf1m194zjru1)

[d. Database queries 21](#_heading=h.x8txdn50pc)

[7. Admin/Dashboard 22](#_heading=h.bs38qeam81m)

[a. Class Diagram 22](#_heading=h.jltfkos2h72e)

[b. Class Specifications 22](#_heading=h.bo2u4wfe9gx)

[XYZ Class 22](#_heading=h.cxonfzlh5zem)

[c. Sequence Diagram(s) 23](#_heading=h.sfgh023qwgm4)

[d. Database queries 23](#_heading=h.v4ogmmv839zf)

[8. Admin/ListCar 24](#_heading=h.lfe7mjm9y3vl)

[a. Class Diagram 24](#_heading=h.jbv6g36kr41)

[b. Class Specifications 24](#_heading=h.bq074cimc4vs)

[XYZ Class 24](#_heading=h.nsc90rutt09k)

[c. Sequence Diagram(s) 25](#_heading=h.6wft06fypnnb)

[d. Database queries 25](#_heading=h.gxdb6y9ejktd)

[9. Admin/CarDetail 26](#_heading=h.avq5dbfpgo6q)

[a. Class Diagram 26](#_heading=h.w5u9l6poluh4)

[b. Class Specifications 26](#_heading=h.vamhrh422rc2)

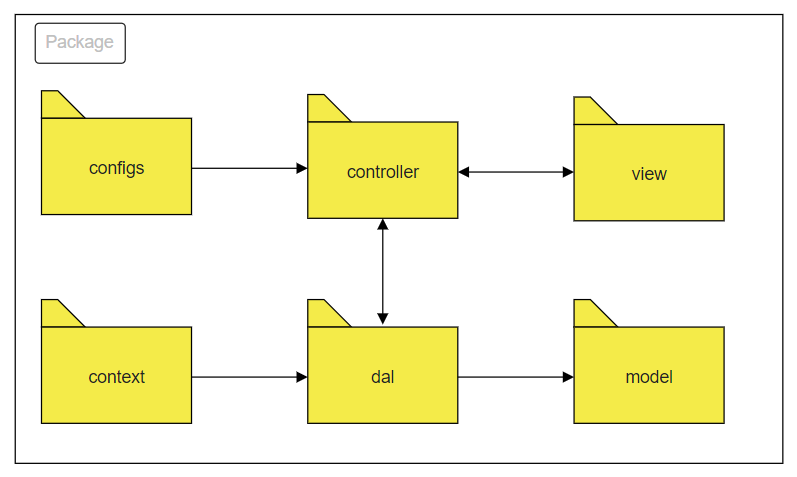
[XYZ Class 26](#_heading=h.ex5cn9nnrtyz)

[c. Sequence Diagram(s) 27](#_heading=h.eijwxnycwjo2)

[d. Database queries 27](#_heading=h.uhsrrq7s03w3)

# I. Overview

## 1. Code Packages/Namespaces

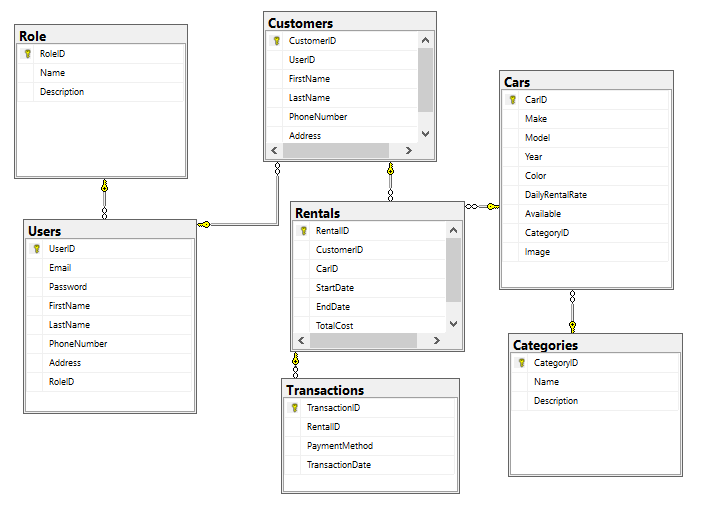


***Package descriptions & package class naming conventions***

| **No** | **Package** | **Description** |
| --- | --- | --- |
| *01* | *controller* | *Acts as a go-between for the service layer and the Front-end, passing on different input requests*  *<Class naming convention>* |
| *02* | *dal* | *This component is used to get data directly FROM the database*  *<Class naming convention>* |
| *03* | *configs* | *This component helps define configuration*  *<Class naming convention>* |
| *04* | *context* | *This component helps connect database*  *<Class naming convention>* |
| *05* | *model* | *This component includes interfaces used to receive data retrieved FROM the database and classes used to receive data*  *<Class naming convention>* |
| *05* | *view* | *This component display front-end*  *<Class naming convention>* |

## 2. Database Design

**2.1 Database Schema**

****

***Table descriptions & package class naming conventions are as below***

| **No** | **Table** | **Description** |
| --- | --- | --- |
| *01* | *Users* | *This table contains information about the users of the system, including their email address, password, name, phone number, address, and role.*  *- Primary keys: UserID*  *- Foreign keys: RoleID* |
| *02* | *Role* | *This table contains information about the various roles that users can have within the system, such as "admin" or "customer service representative".*  *- Primary keys: RoleID* |
| *03* | *Customers* | *This table contains information about the customers of the rental service, including their name, phone number, and address.*  *- Primary keys: CustomerID*  *- Foreign keys: UserID* |
| *04* | *Rentals* | *This table contains information about the car rentals in the system, including the customer who rented the car, the car that was rented, the rental dates, and the total cost.*  *- Primary keys: RentalID*  *- Foreign keys: CustomerID, CarID* |
| *05* | *Cars* | *This table contains information about the cars available for rent, including the make, model, year, color, daily rental rate, availability status, category, and image.*  *- Primary keys: CarID*  *- Foreign keys: CategoryID* |
| *06* | *Categories* | *This table contains information about the various categories of cars in the system, such as "economy" or "luxury".*  *- Primary keys: CategoryID* |
| *07* | *Transactions* | *This table contains information about the financial transactions associated with rentals, including the rental ID, payment method, and transaction date.*  *- Primary keys: TransactionID*  *- Foreign keys: RentalID* |

**2.2 Database table**

## 1. Users

| **#** | **Field name** | **Type** | **Size** | **Unique** | **Not Null** | **PK/FK** | **Notes** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | UserID | int |  | v | v | PK |  |
| 2 | Email | String | 255 |  | v |  |  |
| 3 | Password | String | 255 |  | v |  |  |
| 4 | FirstName | String | 50 |  |  |  |  |
| 5 | LastName | String | 50 |  |  |  |  |
| 6 | PhoneNumber | String | 20 |  |  |  |  |
| 7 | Address | String | 255 |  |  |  |  |
| 8 | RoleId | int |  |  |  | FK |  |

## 2. Role

| **#** | **Field name** | **Type** | **Size** | **Unique** | **Not Null** | **PK/FK** | **Notes** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | RoleID | int |  | v | v | PK |  |
| 2 | Name | String | 50 |  | v |  |  |
| 3 | Description | String | 255 |  |  |  |  |

## 3. Customers

| **#** | **Field name** | **Type** | **Size** | **Unique** | **Not Null** | **PK/FK** | **Notes** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | CustomerID | int |  | v | v | PK |  |
| 2 | UserID | int |  |  | v | FK |  |
| 3 | FirstName | String | 50 |  |  |  |  |
| 4 | LastName | String | 50 |  |  |  |  |
| 5 | PhoneNumber | String | 20 |  |  |  |  |
| 6 | Address | String | 255 |  |  |  |  |

## 

## 

## 4. Rentals

| **#** | **Field name** | **Type** | **Size** | **Unique** | **Not Null** | **PK/FK** | **Notes** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | RentalID | int |  | v | v | PK |  |
| 2 | CustomerID | int |  |  | v | FK |  |
| 3 | CarID | int |  |  | v | FK |  |
| 4 | StartDate | Date |  |  |  |  |  |
| 5 | EndDate | Date |  |  |  |  |  |
| 6 | TotalCost | double |  |  |  |  |  |

## 5. Cars

| **#** | **Field name** | **Type** | **Size** | **Unique** | **Not Null** | **PK/FK** | **Notes** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | CarID | int |  | v | v | PK |  |
| 2 | Make | String | 50 |  | v |  |  |
| 3 | Model | String | 50 |  | v |  |  |
| 4 | Year | int |  |  |  |  |  |
| 5 | Color | String | 50 |  |  |  |  |
| 6 | RentalRate | double |  |  |  |  |  |
| 7 | Available | boolean |  |  |  |  |  |
| 8 | CategoryID | int |  |  |  | FK |  |
| 9 | Image | String | Max |  |  |  |  |

## 6. Categories

| **#** | **Field name** | **Type** | **Size** | **Unique** | **Not Null** | **PK/FK** | **Notes** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | CategoryID | int |  | v | v | PK |  |
| 2 | Name | String | 50 |  | v |  |  |
| 3 | Description | String | 255 |  | v |  |  |

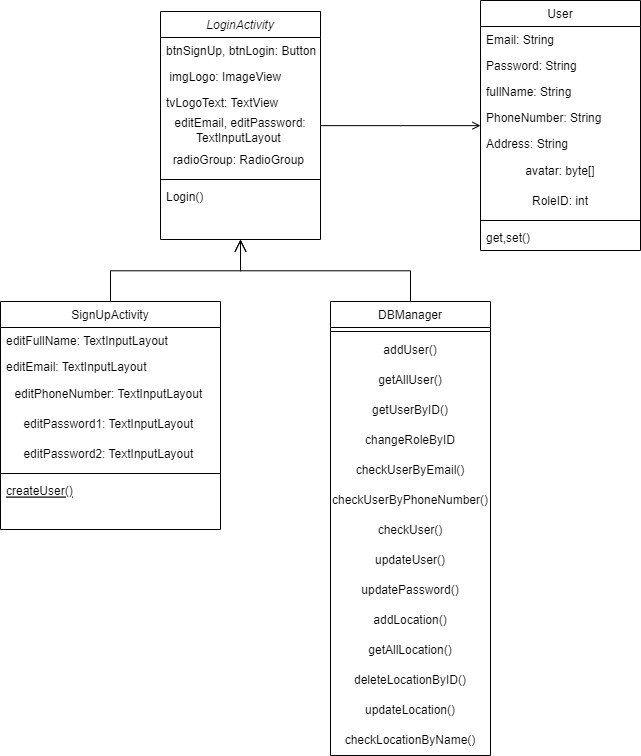
## 7. Transactions

| **#** | **Field name** | **Type** | **Size** | **Unique** | **Not Null** | **PK/FK** | **Notes** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | TransactionID | int |  | v | v | PK |  |
| 2 | RentalID | int |  |  | v | FK |  |
| 3 | PaymentMethod | String | 50 |  |  |  |  |
| 4 | TransactionDate | Date |  |  |  |  |  |

# II. Code Designs

## 1. Login

### a. Class Diagram



### b. Class Specifications

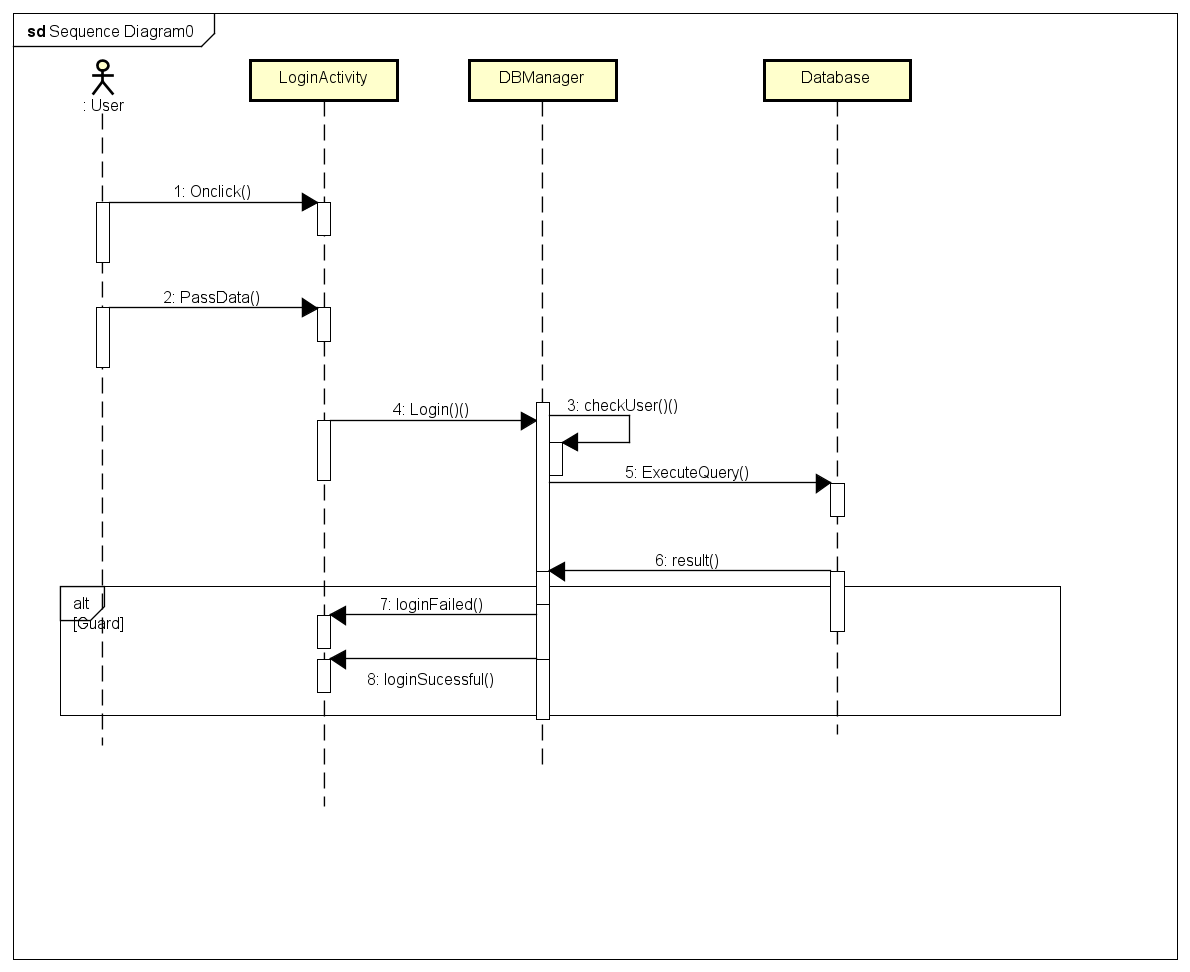
#### XYZ Class

#### 

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *Login* | *Login user into system* |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

### 

### c. Sequence Diagram(s)



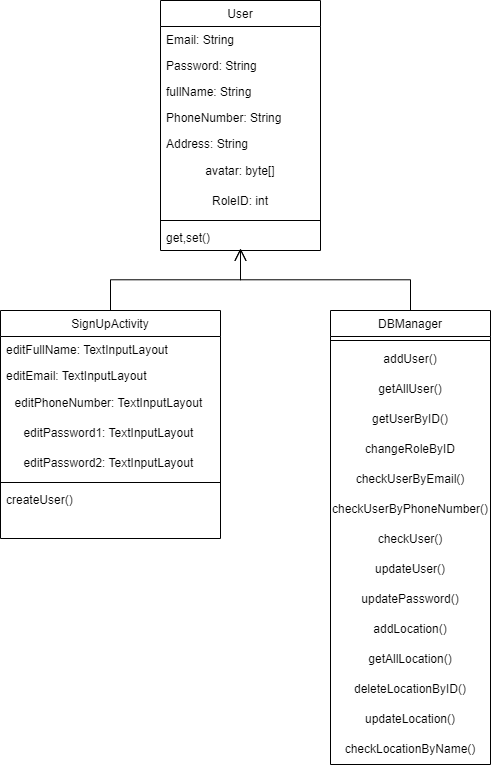
### d. Database queries

SELECT \* FROM Users

WHERE email = ? and password = ?

## 2. User Register

### a. Class Diagram



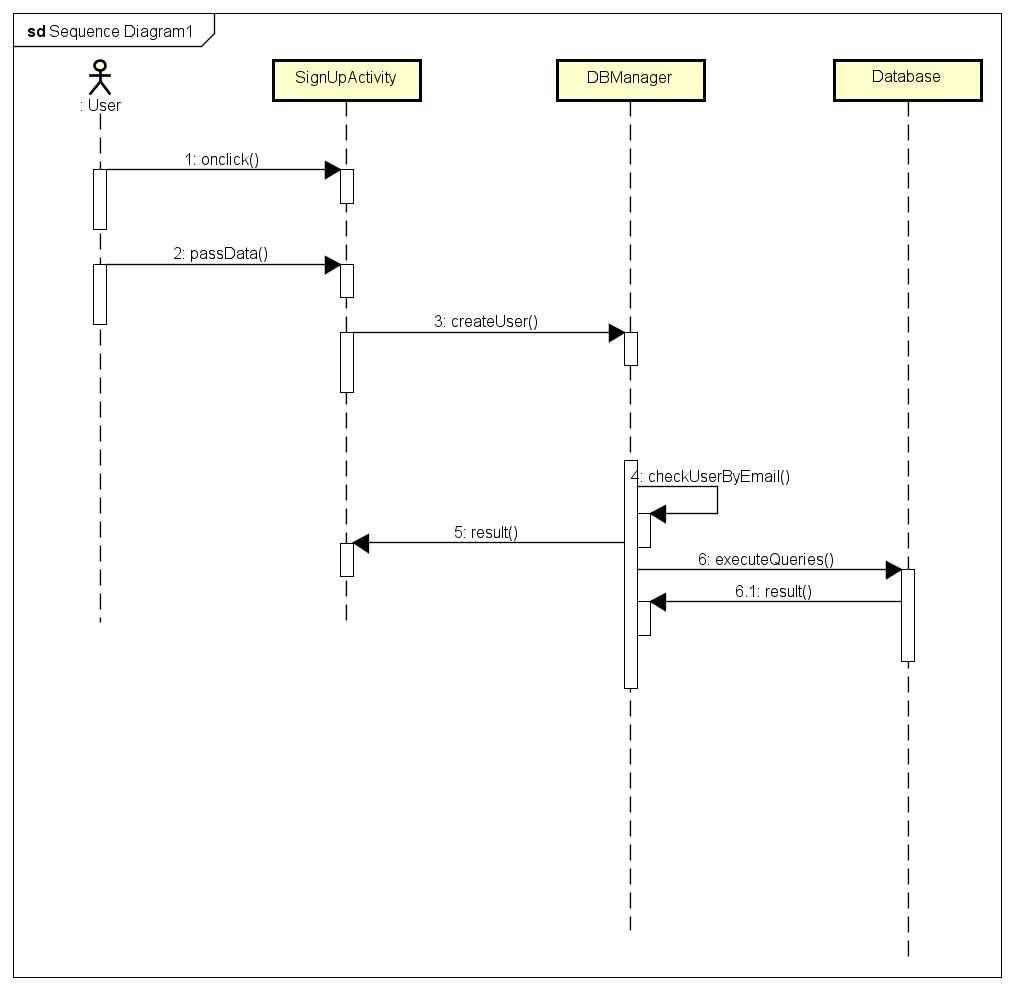
### b. Class Specifications

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *createUser* | *Add new user into database* |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

### 

#### XYZ Class

### c. Sequence Diagram(s)



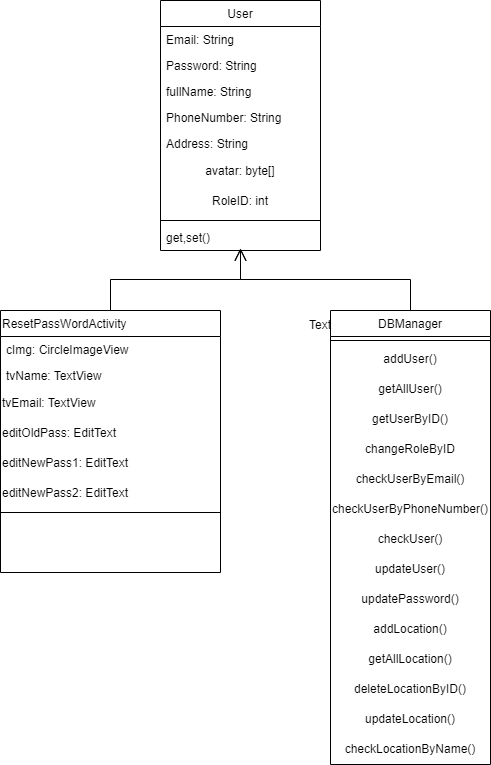
### d. Database queries

INSERT INTO Users (Email,password,FirstName,LastName,PhoneNumber,Address,RoleID)

VALUES(?,?,?,?,?,?,?)

## 3. Reset Password

### a. Class Diagram

****

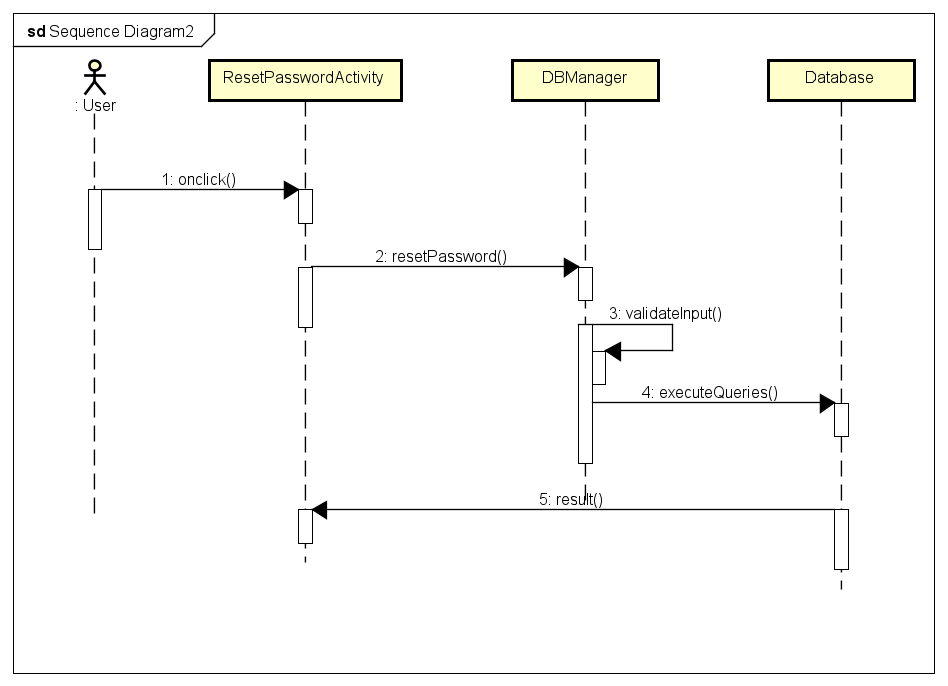
### b. Class Specifications

#### XYZ Class

#### 

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *updatePassword* | *update new password to database* |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

### c. Sequence Diagram(s)

**

### d. Database queries

1.SELECT \* FROM Users

WHERE UserID=?

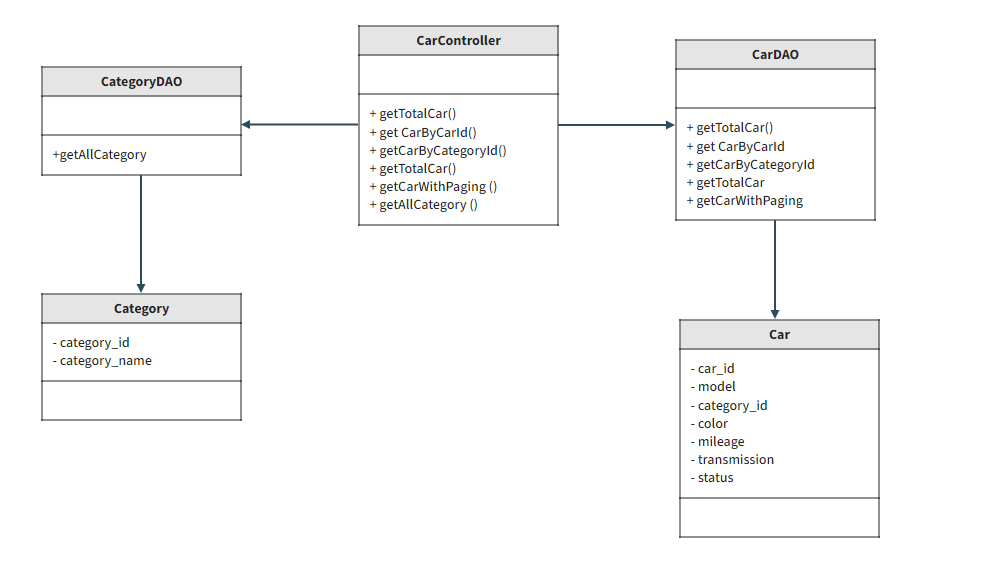
2.UPDATE Users

SET Password

WHERE UserID=?

## 4. Public/Car List

### a. Class Diagram

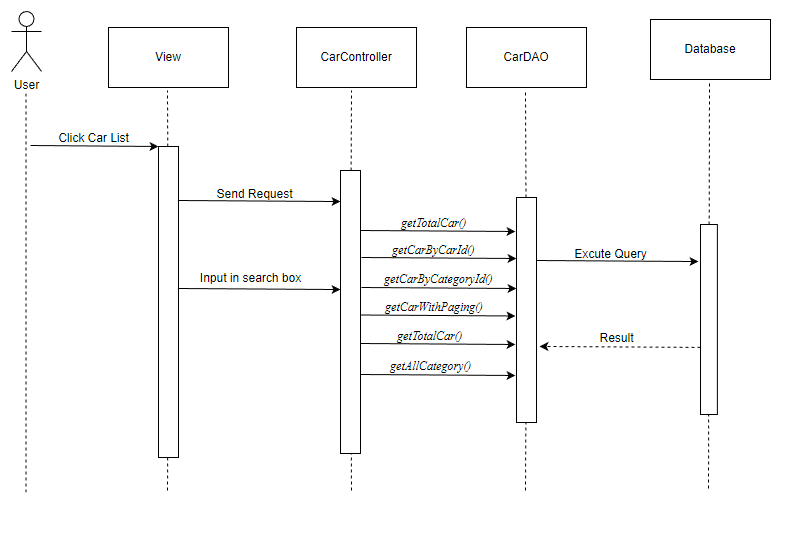


### b. Class Specifications

#### XYZ Class

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *getTotalCar* | *get all car* |
| *02* | *getCarByCarId* | *get car by car id* |
| *03* | *getCarByCategoryId* | *get car by category id* |
| *04* | *getCarWithPaging* | *git car with paging* |
| *05* | *getTotalCar* | *get total car* |
| *06* | *getAllCategory* | *get all category* |

### c. Sequence Diagram(s)



### d. Database queries

Select \* from Car where category\_id = ?

Select \* from Car where car\_id = ?

Select count(\*) from Car "

+ "where category\_id " + categoryId + " and status = 1 and name like N'%" + searchKey + "%'\n

select \* from Car\n"

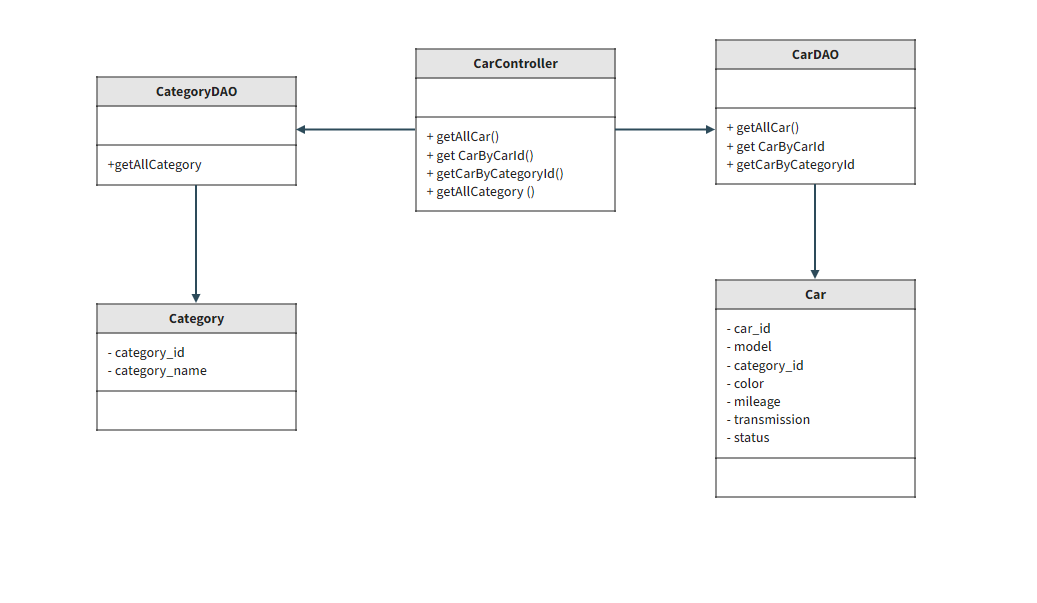
+ "where category\_id " + categoryId + " and status = 1 and name like N'%" + searchKey + "%'\n"

+ " order by " + value + " " + type + " offset (?-1)\*? row fetch next ? row only

Select \* from Category and status = 1

## 5. Public/Car Details

### a. Class Diagram



### b. Class Specifications

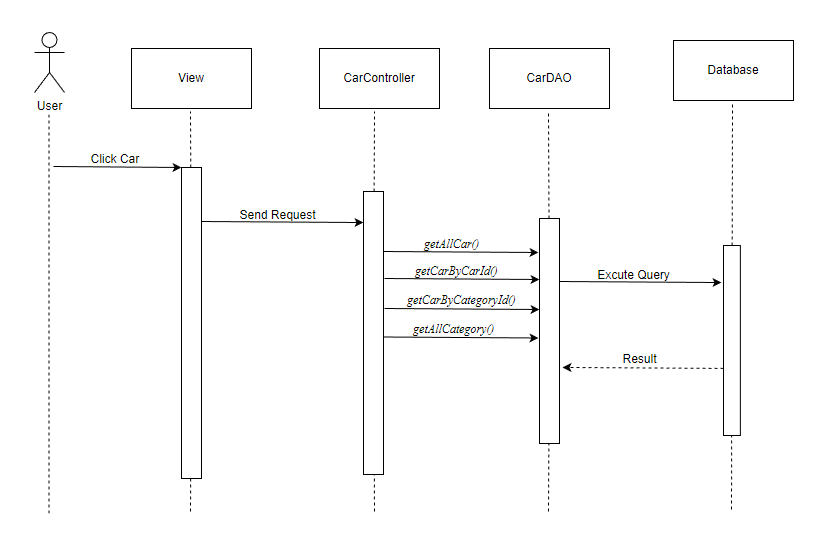
#### XYZ Class

#### 

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *getAllCar* | *get all car* |
| *02* | *getCarByCategoryId* | *get car by category id* |
| *03* | *getCarByCarId* | *get car by carid* |
| *04* | *getAllCategory* | *get all category* |

### 

### c. Sequence Diagram(s)



### d. Database queries

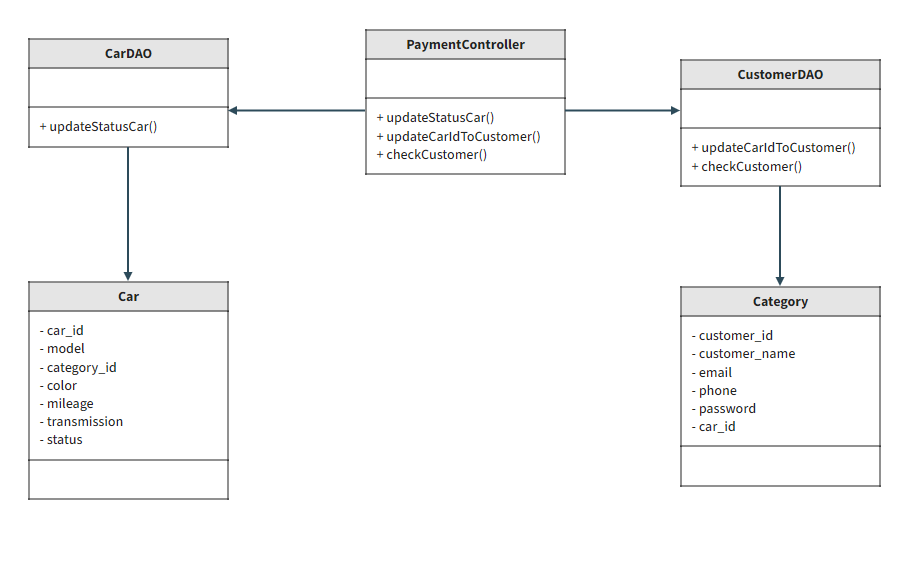
Select \* from Car where category\_id = ?

Select \* from Car where car\_id = ?

Select \* from Category and status = 1

## 6. Public/Payment

### a. Class Diagram

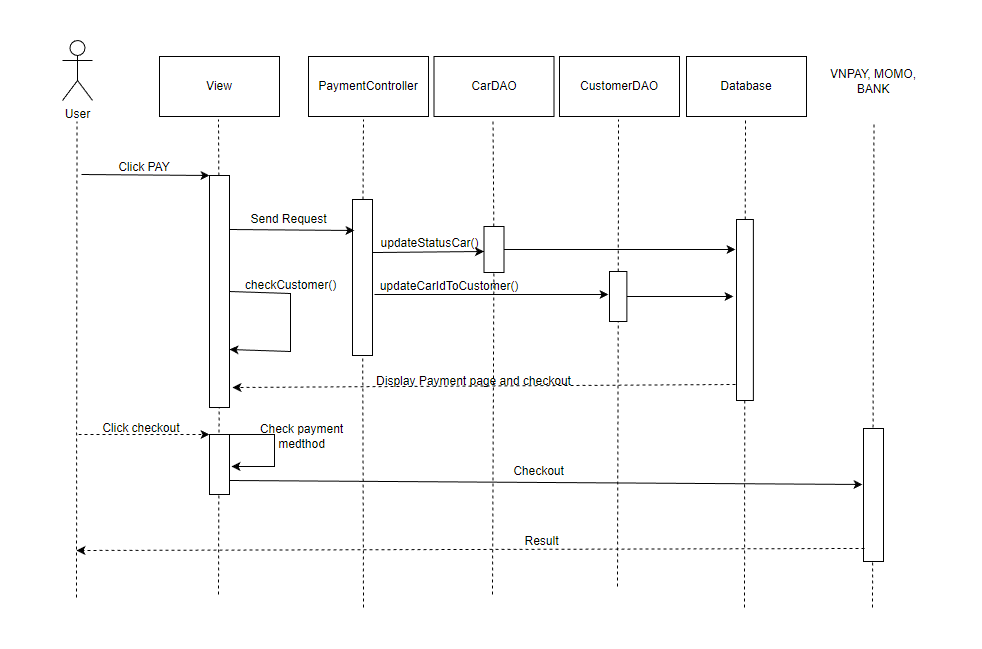


### b. Class Specifications

#### XYZ Class

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *updateStatusOrder()* | *Update status for order selected* |
| *02* | *updateCarIdToCustomer()* | *Update car id for customer selected* |
| *03* | *checkCustomer()* | *Check customer existed* |

### c. Sequence Diagram(s)



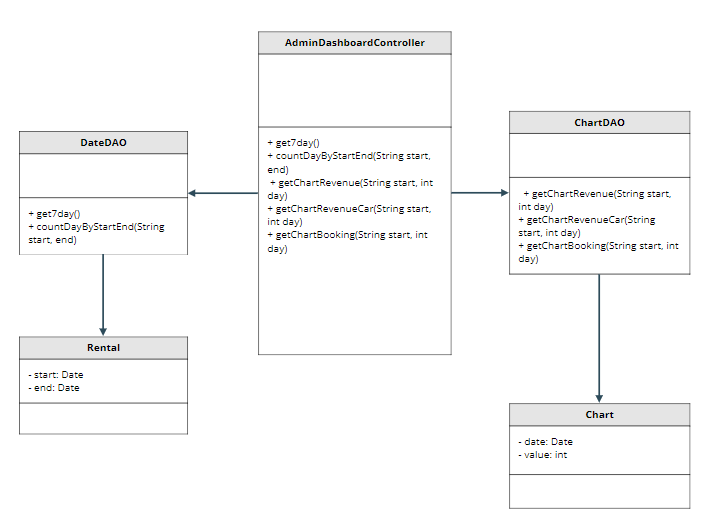
### d. Database queries

* UPDATE Car SET status = ? WHERE car\_id = ?
* UPDATE Customer SET car\_id ? WHERE customer\_id = ?
* SELECT customer\_id, customer\_name, email, mobile FROM Customer

WHERE customer\_name = ? and email = ? and mobile = ?

## 7. Admin/Dashboard

### a. Class Diagram

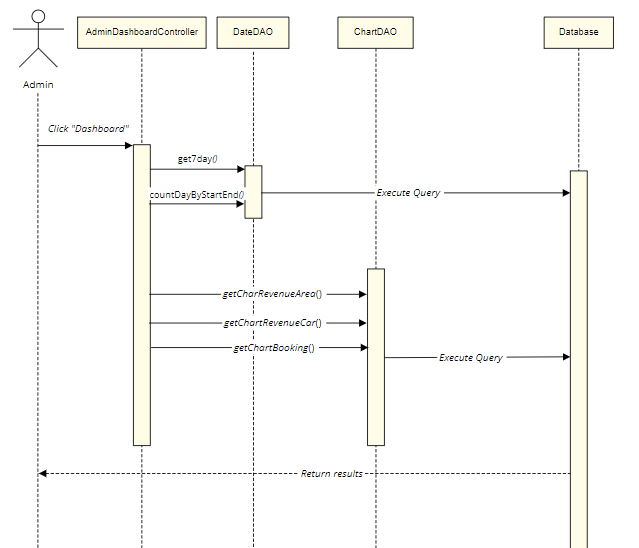


### b. Class Specifications

#### XYZ Class

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *get7day()* | *Get last seven days* |
| *02* | *countDayByStartEnd()* | *Count the number of days difference between start and end* |
| *03* | *getRevenue()* | *Get statistics revenue chart from start date to end date of all* |
| *04* | *getChartRevenueCar()* | *Get statistics revenue chart from start date to end date of each car* |
| *05* | *getChartBooking()* | *Get statistics revenue chart from start date to end date of each customer* |

### c. Sequence Diagram(s)

****

### d. Database queries

SELECT \* FROM cars WHERE availability = 'available'

SELECT \* FROM rentals WHERE customer\_id = <customer\_id>

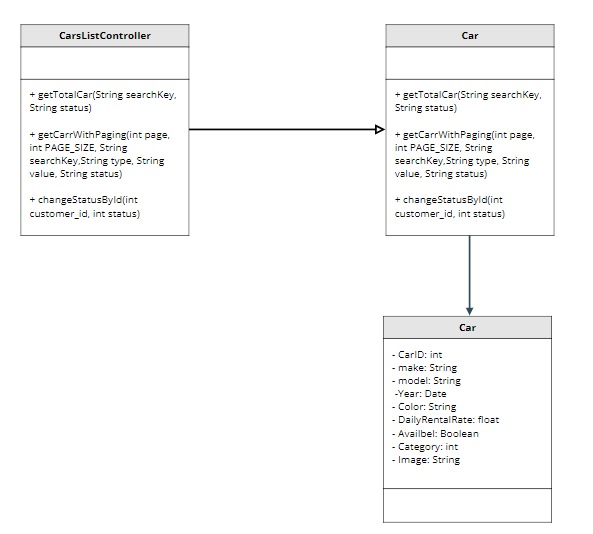
SELECT SUM(total\_cost) FROM rentals

SELECT category, COUNT(\*) AS count FROM rentals JOIN cars ON rentals.car\_id = cars.car\_id GROUP BY category ORDER BY count DESC LIMIT 1

UPDATE cars SET availability = 'unavailable' WHERE car\_id = <car\_id>

## 8. Admin/ListCar

### a. Class Diagram

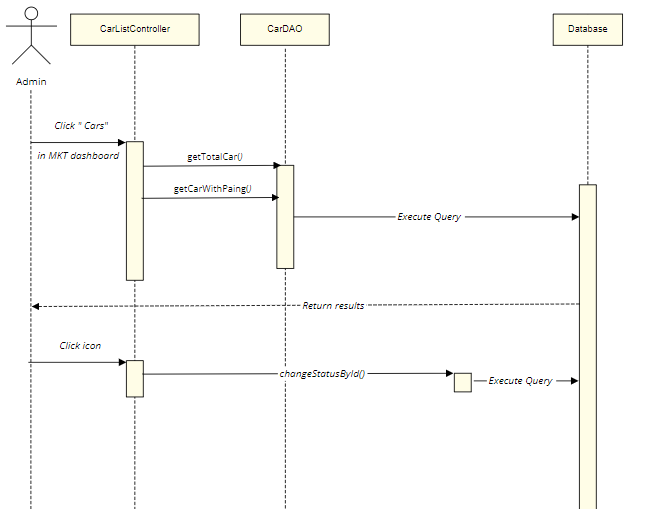


### b. Class Specifications

#### XYZ Class

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *getTotlaCar()* | *Get total car in system* |
| *02* | *getCarWithPaging()* | *Get all car list is paginated* |
| *03* | *changeStatus()* | *Change status of specific car* |

### c. Sequence Diagram(s)

****

### d. Database queries

UPDATE Car SET status = ? WHERE car\_id = ?

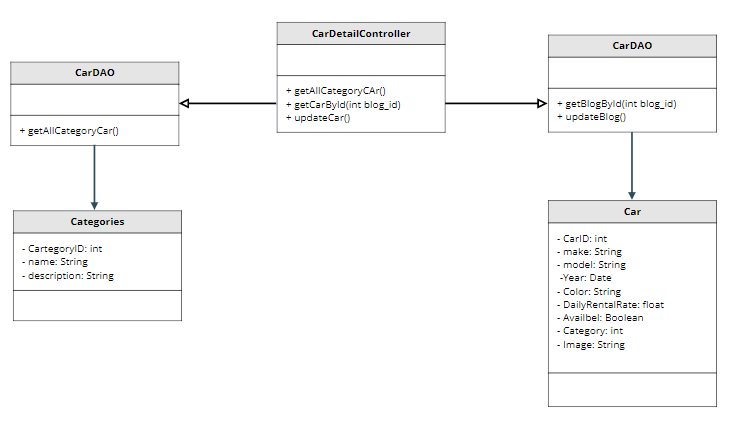
Select \* from Car where car\_id = ?

select \* from Car where category\_id " + categoryId + " and status = 1 and name like N'%" + searchKey + "%'\n"

+ " order by " + value + " " + type + " offset (?-1)\*? row fetch next ? row only

## 9. Admin/CarDetail

### a. Class Diagram

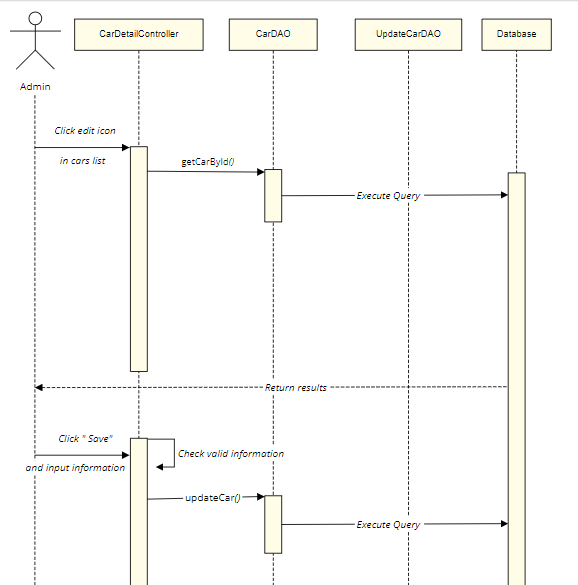


### b. Class Specifications

#### XYZ Class

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *getAllCategoryCar()* | *Get all category of car in system* |
| *02* | *getCarById()* | *Get specific car by id* |
| *03* | *updateCar()* | *Update specific car with new value* |

### c. Sequence Diagram(s)

****

### d. Database queries

UPDATE Car SET specific = ? WHERE car\_id = ?

Select \* from Car where car\_id = ?

select \* from Category