

**CAR RENTAL MANAGEMENT SYSTEM**

**Software Design Document**

– Hanoi, August 2019 –

**Table of Contents**

[**I. Overview**](#_heading=h.gjdgxs) **8**

[1. Code Packages/Namespaces](#_heading=h.30j0zll) 8

[2. Database Design](#_heading=h.1fob9te) 8

[**II. Code Designs**](#_heading=h.hcthmehyzkhr) **13**

[1. Manage order car/ View order list](#_heading=h.3dy6vkm) 13

[a. Class Diagram](#_heading=h.1t3h5sf) 13

[c. Class Specifications](#_heading=h.4d34og8) 13

[c. Sequence Diagram(s)](#_heading=h.2s8eyo1) 14

[d. Database queries](#_heading=h.17dp8vu) 14

[2. Manage order car/ View order details](#_heading=h.zfp64d50je5e) 14

[a. Class Diagram](#_heading=h.eqzuqcw79wp1) 14

[b. Class Specifications](#_heading=h.6utlxyijstvj) 14

[c. Sequence Diagram(s)](#_heading=h.ovc2p4wo7z9t) 15

[d. Database queries](#_heading=h.wquyq9lr7bi3) 15

[3. Manage order car/ Extra surcharge](#_heading=h.qlib6yjo74cw) 16

[a. Class Diagram](#_heading=h.wow0n0pjmdhc) 16

[b. Class Specifications](#_heading=h.jnlttn4zpg1k) 16

[c. Sequence Diagram(s)](#_heading=h.wpsb1spkqxoh) 17

[d. Database queries](#_heading=h.e8tuxc7e9rik) 17

[4. Manage order car/ Accept order, Cancel order, History/Cancel booking](#_heading=h.b083cp9kzhi8) 17

[a. Class Diagram](#_heading=h.ua3s51ufl3c5) 17

[b. Class Specifications](#_heading=h.h7h8gzee79ci) 17

[c. Sequence Diagram(s)](#_heading=h.x11vbnvyadu) 18

[d. Database queries](#_heading=h.uvn3yr1boz9r) 18

[5. History/ View list booked car](#_heading=h.5sbbx732nrl6) 18

[a. Class Diagram](#_heading=h.dte37d6geih1) 18

[b. Class Specifications](#_heading=h.365qkefanbpm) 19

[c. Sequence Diagram(s)](#_heading=h.dulgsbug23kb) 19

[d. Database queries](#_heading=h.su8dqqdc5oa3) 19

[6. History/ View booking details](#_heading=h.q55b7ae3ny08) 20

[a. Class Diagram](#_heading=h.r3ynlrmcl93u) 20

[b. Class Specifications](#_heading=h.3f8skfhnopsp) 20

[c. Sequence Diagram(s)](#_heading=h.ko1nbunf5jb1) 21

[d. Database queries](#_heading=h.xfzfvshz7b7q) 21

[7. History/ Edit booking](#_heading=h.sc9nq1feq5j4) 21

[a. Class Diagram](#_heading=h.uu102gc68o1o) 21

[b. Class Specifications](#_heading=h.w60rd76zp0ps) 21

[c. Sequence Diagram(s)](#_heading=h.b3zhos5w6lfx) 22

[d. Database queries](#_heading=h.5mcomvsnqf81) 22

[8. History/ Feedback](#_heading=h.fpaauq25ydil) 23

[a. Class Diagram](#_heading=h.y5ydf172zafr) 23

[b. Class Specifications](#_heading=h.9hp6c988qclj) 23

[c. Sequence Diagram(s)](#_heading=h.7syc7mmh35b9) 24

[d. Database queries](#_heading=h.ym6eme3lzaqj) 24

[9. Car rental/ Find car](#_heading=h.byoo5ssq0pg) 24

[a. Class Diagram](#_heading=h.j2dmincbga6n) 24

[b. Class Specifications](#_heading=h.na2nbnfz7k5z) 24

[c. Sequence Diagram(s)](#_heading=h.g3dievi6l0ff) 25

[d. Database queries](#_heading=h.52tid2tvsfu3) 25

[10. Car rental/View list car](#_heading=h.b732o6aem6xj) 25

[a. Class Diagram](#_heading=h.ul29syqdjuy9) 25

[b. Class Specifications](#_heading=h.oi4mk7lp0de4) 25

[c. Sequence Diagram(s)](#_heading=h.w6e50bgriz8w) 26

[d. Database queries](#_heading=h.plulz6xz3bl9) 26

[11. Car rental/ View detail car](#_heading=h.q2iz2q8ttnhl) 26

[a. Class Diagram](#_heading=h.tf7lpxpqskjq) 26

[b. Class Specifications](#_heading=h.56vnhp51evao) 26

[c. Sequence Diagram(s)](#_heading=h.gl25pystp9h3) 27

[d. Database queries](#_heading=h.arm5g0pj9zzi) 27

[12. Car rental/ Car rental](#_heading=h.waocu0uaso2m) 27

[a. Class Diagram](#_heading=h.5ujhfh9qhjvl) 27

[b. Class Specifications](#_heading=h.o3lstg9h6g3e) 27

[c. Sequence Diagram(s)](#_heading=h.kmtdcvr2g7zu) 28

[d. Database queries](#_heading=h.mgj11vjl8u62) 28

[13. Car rental/ View feedback](#_heading=h.b4y26xu1g27) 28

[a. Class Diagram](#_heading=h.sxdb37epgxd6) 28

[b. Class Specifications](#_heading=h.r8ot3pne55w3) 28

[c. Sequence Diagram(s)](#_heading=h.as5hbq587bk) 29

[d. Database queries](#_heading=h.wa1kjlc17qwb) 29

[14. Car rental/ Add driving license](#_heading=h.ii5ejir7wjsz) 29

[a. Class Diagram](#_heading=h.wq8sb3fy7fux) 29

[b. Class Specifications](#_heading=h.5ubtcm7eaq7u) 29

[c. Sequence Diagram(s)](#_heading=h.4bsmvxao78di) 30

[d. Database queries](#_heading=h.qp8l86qn8il5) 31

[15. Individual](#_heading=h.ys2iov8xr67j) 31

[a. Class Diagram](#_heading=h.sbtsibenyroy) 31

[b. Class Specifications](#_heading=h.41jskdhkz2an) 31

[c. Sequence Diagram(s)](#_heading=h.jlmps1q2gcf) 32

[d. Database queries](#_heading=h.1diu3nyzjmi3) 32

[16. View address](#_heading=h.7j1a6yqfbxef) 32

[a. Class Diagram](#_heading=h.h4bcexu2pfqj) 32

[b. Class Specification](#_heading=h.t8hrd0gfd1nv) 32

[c. Sequence Diagram(s)](#_heading=h.ayu4n84zblb6) 33

[d. Database queries](#_heading=h.a2e5uektulvd) 33

[17. Add address](#_heading=h.vawcs3fi0zh) 33

[a. Class Diagram](#_heading=h.x1kedsd43jju) 33

[b. Class Specification](#_heading=h.1wn0gelyd968) 34

[c. Sequence Diagram(s)](#_heading=h.6hbrygp4f6e) 34

[d. Database queries](#_heading=h.1kvueqqg6vr) 35

[18. Account information](#_heading=h.r3oqeni527gj) 35

[a. Class Diagram](#_heading=h.2et92p0) 35

[b. Class Specification](#_heading=h.tyjcwt) 35

[c. Sequence Diagram(s)](#_heading=h.49ugbaa6bc7p) 36

[d. Database queries](#_heading=h.senjpaej3iyh) 36

[19. Verification driver’s license](#_heading=h.g476no6jlkj3) 36

[a. Class Diagram](#_heading=h.maxnb8cpga7s) 36

[b. Class Specification](#_heading=h.9k1pzhuezb4k) 37

[c. Sequence Diagram(s)](#_heading=h.6q9nmledg66z) 38

[d. Database queries](#_heading=h.a1t3vz3fts4i) 38

[20. Verification email](#_heading=h.wvllfp9n5mr8) 38

[a. Class Diagram](#_heading=h.anfl6d3skocf) 38

[b. Class Specification](#_heading=h.lwlmsgkxfk97) 39

[c. Sequence Diagram(s)](#_heading=h.qtyvx8ns84pw) 39

[d. Database queries](#_heading=h.ivizuvp9ekks) 39

[21. Edit information](#_heading=h.nqxvkndgewdm) 39

[a. Class Diagram](#_heading=h.e73ug6em4l4y) 39

[b. Class Specification](#_heading=h.6l5zmnmgps19) 40

[c. Sequence Diagram(s)](#_heading=h.2ehfa2e8s782) 41

[d. Database queries](#_heading=h.gjxt0z7avfjo) 41

[22. Change password](#_heading=h.srx12ti4rzmd) 41

[a. Class Diagram](#_heading=h.pgy1iv9qtu1o) 41

[b. Class Specification](#_heading=h.ypn9k5qniani) 42

[c. Sequence Diagram(s)](#_heading=h.71tztup7xlgi) 42

[d. Database queries](#_heading=h.e2oschhemmb8) 42

[23. My Car](#_heading=h.expan4awzbno) 42

[a. Screen Design](#_heading=h.p2kxb8tkwcen) 42

[b. Class Diagram](#_heading=h.aeba16jyiwi7) 43

[c. Class Specifications](#_heading=h.8rm80v88jtan) 43

[d. Sequence Diagram(s)](#_heading=h.9fzdltmwk9jl) 44

[e. Database queries](#_heading=h.d5mmaipna5zb) 44

[24. Car Information](#_heading=h.v50k9dpxbw5i) 45

[a. Screen Design](#_heading=h.77t22bdylz18) 45

[b. Class Diagram](#_heading=h.t07ujoozkxg4) 45

[c. Class Specifications](#_heading=h.9rkmh2cdgjju) 46

[d. Sequence Diagram(s)](#_heading=h.bluq94a2tahu) 47

[e. Database queries](#_heading=h.2u2y22gjhs3g) 48

[25. Add New Car](#_heading=h.cv7ftmkaxbl6) 48

[a. Class Diagram](#_heading=h.icgqen2pvuc6) 48

[b. Class Specifications](#_heading=h.ev4yzd3fg301) 49

[c. Sequence Diagram(s)](#_heading=h.es2abncelogt) 51

[d. Database queries](#_heading=h.su0ajncpg8f2) 52

[26. Login](#_heading=h.q881n86tltl3) 52

[a. Class Diagram](#_heading=h.4zp5xn3i1oh) 52

[b. Class Specifications](#_heading=h.l22iq8urist5) 52

[c. Sequence Diagram(s)](#_heading=h.5nj8na5198ec) 53

[d. Database queries](#_heading=h.18udpcsbzgc) 53

[27. Register](#_heading=h.dm6dxfv12mg3) 54

[a. Class Diagram](#_heading=h.p2dfqcxbmhg) 54

[b. Class Specifications](#_heading=h.o5uprlk4j1kw) 54

[c. Sequence Diagram(s)](#_heading=h.xeicbl4hhabs) 54

[d. Database queries](#_heading=h.b93818e6xrla) 55

[28. Forget Password](#_heading=h.nja9n1ksdyv3) 56

[a. Class Diagram](#_heading=h.luy3posanjnr) 56

[b. Class Specifications](#_heading=h.km15pricf4gv) 56

[c. Sequence Diagram(s)](#_heading=h.mfnxzz7wmmy) 56

[d. Database queries](#_heading=h.97v28f9bykxw) 57

[29. Support](#_heading=h.9qod2wclkim) 57

[a. Class Diagram](#_heading=h.c58anyc05dub) 57

[b. Class Specifications](#_heading=h.n31aa2cwxfcs) 58

[c. Sequence Diagram(s)](#_heading=h.7prbrqswr0qu) 58

[d. Database queries](#_heading=h.o7spala4rh8) 59

[30. Home](#_heading=h.w51q6ef2tw2r) 59

[a. Class Diagram](#_heading=h.fdds7c5ooalh) 60

[b. Class Specifications](#_heading=h.hoieok66w644) 60

[c. Sequence Diagram(s)](#_heading=h.205b0l1unwvp) 61

[d. Database queries](#_heading=h.9vl7zdxd5ohh) 61

[31. Notification](#_heading=h.flrb0ucwqa1a) 62

[a. Class Diagram](#_heading=h.rccg05tc6uty) 62

[b. Class Specifications](#_heading=h.9celrsagwi2z) 62

[c. Sequence Diagram(s)](#_heading=h.w1mf9qb8gjrx) 63

[d. Database queries](#_heading=h.b5npjfyjs5r8) 63

[32. Notification Detail](#_heading=h.wprm7stbw1fs) 63

[a. Class Diagram](#_heading=h.ro39qqt9h5aw) 63

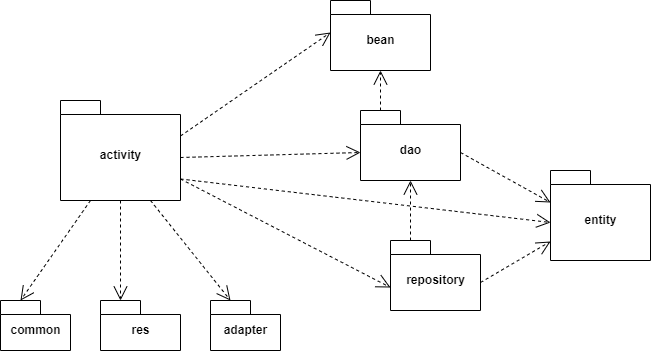
[b. Class Specifications](#_heading=h.57s6vxmrm2dp) 64

[c. Sequence Diagram(s)](#_heading=h.qlz2dwz6f9i3) 65

[d. Database queries](#_heading=h.l9183a6qsd56) 65

# I. Overview

## 1. Code Packages/Namespaces

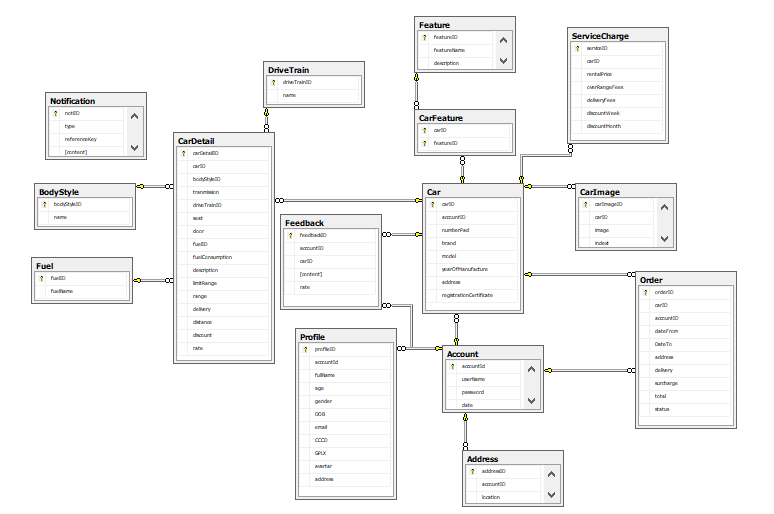


***Package descriptions & package class naming conventions***

| **No** | **Package** | **Description** |
| --- | --- | --- |
| *01* | *activity* | *Contain Activity files* |
| *02* | *common* | *Contain common files* |
| *03* | *res* | *Contain application resources* |
| *04* | *adapter* | *Contain adapter files* |
| *05* | *bean* | *Contain objects* |
| *06* | *dao* | *Contain DAO files* |
| *07* | *repository* | *Contain repository files* |
| *08* | *entity* | *Contain entities* |

## 2. Database Design

**2.1 Database Schema**

****

**2.2 Database table**

***Account***

| **#** | **Field name** | **Type** | **Size** | **Unique** | **Not Null** | **PK/FK** | **Notes** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | accountD | int |  |  |  | PK |  |
| 2 | userName | nvarchar | 200 |  |  |  |  |
| 3 | password | varchar | 10 |  |  |  |  |
| 4 | date | datetime |  |  |  |  |  |

***Profile***

| **#** | **Field name** | **Type** | **Size** | **Unique** | **Not Null** | **PK/FK** | **Notes** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | profileID | int |  |  |  | PK |  |
| 2 | accountID | int |  |  |  | FK |  |
| 3 | fullName | nvarchar | 200 |  |  |  |  |
| 4 | age | int |  |  |  |  |  |
| 5 | gende | bit |  |  |  |  |  |
| 6 | DOB | date |  |  |  |  |  |
| 7 | email | varchar | 50 |  |  |  |  |
| 8 | CCCD | text |  |  |  |  |  |
| 9 | GPLX | text |  |  |  |  |  |
| 10 | avartar | text |  |  |  |  |  |
| 11 | addressID | int |  |  |  |  |  |

***Address***

| **#** | **Field name** | **Type** | **Size** | **Unique** | **Not Null** | **PK/FK** | **Notes** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | addressID | int |  |  |  | PK |  |
| 2 | accountID | int |  |  |  |  |  |
| 3 | location | nvarchar | 200 |  |  | FK |  |

***Notification***

| **#** | **Field name** | **Type** | **Size** | **Unique** | **Not Null** | **PK/FK** | **Notes** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | notiID | int |  |  |  | PK |  |
| 2 | type | varchar | 50 |  |  |  |  |
| 3 | referenceKey | int |  |  |  |  |  |
| 4 | content | text |  |  |  |  |  |

***Car***

| **#** | **Field name** | **Type** | **Size** | **Unique** | **Not Null** | **PK/FK** | **Notes** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | carID | int |  |  |  | PK |  |
| 2 | accountID | int |  |  |  | FK |  |
| 3 | numberPad | varchar | 50 |  |  |  |  |
| 4 | brand | varchar | 50 |  |  |  |  |
| 5 | model | varchar | 50 |  |  |  |  |
| 6 | yearOfManufacture | date |  |  |  |  |  |
| 7 | address | nvarchar | 200 |  |  |  |  |
| 8 | registrationCertificate | text |  |  |  |  |  |

***CarImage***

| **#** | **Field name** | **Type** | **Size** | **Unique** | **Not Null** | **PK/FK** | **Notes** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | carImageID | int |  |  |  | PK |  |
| 2 | image | text |  |  |  |  |  |
| 3 | indext | int |  |  |  |  |  |
| 4 | carID | int |  |  |  | FK |  |

***Feature***

| **#** | **Field name** | **Type** | **Size** | **Unique** | **Not Null** | **PK/FK** | **Notes** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | featureID | int |  |  |  | PK |  |
| 2 | deatureName | varchar | 200 |  |  |  |  |
| 3 | description | text |  |  |  |  |  |

***CarFeature***

| **#** | **Field name** | **Type** | **Size** | **Unique** | **Not Null** | **PK/FK** | **Notes** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | carID | int |  |  |  | PK,FK |  |
| 2 | featureID | int |  |  |  | PK,FK |  |

***DriverTrain***

| **#** | **Field name** | **Type** | **Size** | **Unique** | **Not Null** | **PK/FK** | **Notes** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | driveTrainID | int |  |  |  | PK |  |
| 2 | name | int |  |  |  |  |  |

***Fuel***

| **#** | **Field name** | **Type** | **Size** | **Unique** | **Not Null** | **PK/FK** | **Notes** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | fuelID | int |  |  |  | PK |  |
| 2 | fuelName | varchar | 100 |  |  |  |  |

***ServiceCharge***

| **#** | **Field name** | **Type** | **Size** | **Unique** | **Not Null** | **PK/FK** | **Notes** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | serviceID | int |  |  |  | PK |  |
| 2 | carID | int |  |  |  | FK |  |
| 3 | rentalPrice | float |  |  |  |  |  |
| 4 | overRangeFees | float |  |  |  |  |  |
| 5 | deliveryFees | float |  |  |  |  |  |
| 6 | discountWeek | float |  |  |  |  |  |
| 7 | discountMonth | float |  |  |  |  |  |

***BodyStyle***

| **#** | **Field name** | **Type** | **Size** | **Unique** | **Not Null** | **PK/FK** | **Notes** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | bodyStyleID | int |  |  |  | PK |  |
| 2 | name | varchar | 200 |  |  |  |  |

***CarDetail***

| **#** | **Field name** | **Type** | **Size** | **Unique** | **Not Null** | **PK/FK** | **Notes** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | carDetailID | int |  |  |  | PK |  |
| 2 | carID | int |  |  |  | FK |  |
| 3 | bodyStyleID | int |  |  |  | FK |  |
| 4 | transmission | bit |  |  |  |  |  |
| 5 | driveTrainID | int |  |  |  | FK |  |
| 6 | seat | int |  |  |  |  |  |
| 7 | door | int |  |  |  |  |  |
| 8 | fuelID | int |  |  |  | FK |  |
| 9 | fuelConsumption | float |  |  |  |  |  |
| 10 | description | text |  |  |  |  |  |
| 11 | limitRange | bit |  |  |  |  |  |
| 12 | range | int |  |  |  |  |  |
| 13 | delivery | bit |  |  |  |  |  |
| 14 | distance | int |  |  |  |  |  |
| 15 | discount | bit |  |  |  |  |  |
| 16 | rate | float |  |  |  |  |  |

***Order***

| **#** | **Field name** | **Type** | **Size** | **Unique** | **Not Null** | **PK/FK** | **Notes** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | orderID | int |  |  |  | PK |  |
| 2 | carID | int |  |  |  | FK |  |
| 3 | accountID | int |  |  |  | FK |  |
| 4 | dateFrom | datetime |  |  |  |  |  |
| 5 | dateTo | datetime |  |  |  |  |  |
| 6 | address | nvarchar | 200 |  |  |  |  |
| 7 | delivery | bit |  |  |  |  |  |
| 8 | surcharge | float |  |  |  |  |  |
| 9 | total | float |  |  |  |  |  |
| 10 | status | varchar | 50 |  |  |  |  |

***Feedback***

| **#** | **Field name** | **Type** | **Size** | **Unique** | **Not Null** | **PK/FK** | **Notes** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | feedbackID | int |  |  |  | PK |  |
| 2 | accountID | int |  |  |  | FK |  |
| 3 | carID | int |  |  |  | FK |  |
| 4 | content | text |  |  |  |  |  |
| 5 | rate | int |  |  |  |  |  |

# II. Code Designs

## 1. Manage order car/ View order list

### a. Class Diagram

Graphical user interface, application, Teams

Description automatically generated

### c. Class Specifications

***IndividualActivity Class***

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 1 | onClick(View) | Method that is called when the view(component) is clicked |
| 2 | onCreate(Bundle) | Method is used to start an activity |

***ViewOrderListActivity Class***

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 1 | onCreate(Bundle) | Method is used to start an activity |

***OrderDAO Class***

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 1 | getAllOrder(int) | Method is used to get all order of car by accountId |

### c. Sequence Diagram(s)

Graphical user interface

Description automatically generated with medium confidence

### d. Database queries

- SELECT \* FROM Order o JOIN Car c ON o.IdC = c.IdC WHERE c.IdA = ?

- SELECT \* FROM Car WHERE IdA = ?

## 2. Manage order car/ View order details

### a. Class Diagram

Graphical user interface, application, Teams

Description automatically generated

### b. Class Specifications

***ViewOrderListActivity Class***

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 1 | onClick(View) | Method that is called when the view(component) is clicked |
| 2 | onCreate(Bundle) | Method is used to start an activity |

***ViewOrderDetailActivity Class***

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 1 | onCreate(Bundle) | Method is used to start an activity |

***OrderDAO Class***

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 1 | getOrderById(int) | Method is used to get order of car by orderID |

### c. Sequence Diagram(s)

Diagram

Description automatically generated with medium confidence

### d. Database queries

- SELECT \* FROM Order WHERE IdO = ?

- SELECT \* FROM Car WHERE IdC = ?

## 3. Manage order car/ Extra surcharge

### a. Class Diagram

Graphical user interface, application, Teams

Description automatically generated

### b. Class Specifications

***ViewOrderDetailActivity Class***

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 1 | onClick(View) | Method that is called when the view(component) is clicked |
| 2 | onCreate(Bundle) | Method is used to start an activity |

***ExtraSurchargeActivity Class***

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 1 | onCreate(Bundle) | Method is used to start an activity |

***OrderDAO Class***

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 1 | setSurcharge(int, double) | Method is used to set surcharge for order and set status of order is finish |

***ChargesDAO Class***

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 1 | getAllCharges(int) | Method is used to get all charges of car |

### c. Sequence Diagram(s)

Graphical user interface, diagram

Description automatically generated with medium confidence

### d. Database queries

- SELECT \* FROM Charges WHERE IdC = ?

- UPDATE Order SET surcharge = ? WHERE IdO = ?

## 4. Manage order car/ Accept order, Cancel order, History/Cancel booking

### a. Class Diagram

Graphical user interface, application, Teams

Description automatically generated

### b. Class Specifications

***ViewOrderDetailActivity Class***

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 1 | onClick(View) | Method that is called when the view(component) is clicked |
| 2 | onCreate(Bundle) | Method is used to start an activity |

***SetStatusOrderActivity Class***

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 1 | onCreate(Bundle) | Method is used to start an activity |

***OrderDAO Class***

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 1 | setStatusOrder(int) | Method is used to set status for Order |

### c. Sequence Diagram(s)

Diagram

Description automatically generated with medium confidence

### d. Database queries

- UPDATE Order SET status = ‘…’ WHERE IdO = ?

## 5. History/ View list booked car

### a. Class Diagram

Graphical user interface, application, Teams

Description automatically generated

### b. Class Specifications

***HomeActivity Class***

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 1 | onClick(View) | Method that is called when the view(component) is clicked |
| 2 | onCreate(Bundle) | Method is used to start an activity |

***ViewListBookedActivity Class***

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 1 | onCreate(Bundle) | Method is used to start an activity |

***OrderDAO Class***

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 1 | getListBooked(int) | Method is used to get order of car by accountID |

### c. Sequence Diagram(s)

Graphical user interface

Description automatically generated with medium confidence

### d. Database queries

- SELECT \* FROM Order WHERE IdA = ?

- SELECT \* FROM Car WHERE IdC = ?

## 6. History/ View booking details

### a. Class Diagram

Graphical user interface, application, Teams

Description automatically generated

### b. Class Specifications

***ViewOrderListActivity Class***

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 1 | onClick(View) | Method that is called when the view(component) is clicked |
| 2 | onCreate(Bundle) | Method is used to start an activity |

***ViewOrderDetailActivity Class***

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 1 | onCreate(Bundle) | Method is used to start an activity |

***OrderDAO Class***

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 1 | getOrderById(int) | Method is used to get order of car by orderID |

### c. Sequence Diagram(s)

Diagram

Description automatically generated with medium confidence

### d. Database queries

- SELECT \* FROM Order WHERE IdO = ?

- SELECT \* FROM Car WHERE IdC = ?

## 7. History/ Edit booking

### a. Class Diagram

Graphical user interface, application, Teams

Description automatically generated

### b. Class Specifications

***ViewBookingDetailActivity Class***

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 1 | onClick(View) | Method that is called when the view(component) is clicked |
| 2 | onCreate(Bundle) | Method is used to start an activity |

***EditBookingActivity Class***

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 1 | onCreate(Bundle) | Method is used to start an activity |

***OrderDAO Class***

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 1 | getOrderById(int) | Method is used to get order of car by orderID |
| 2 | updateOrder(Order) | Method is used to update order by orderID |

### c. Sequence Diagram(s)

Diagram

Description automatically generated

### d. Database queries

- UPDATE Order SET …… WHERE IdO = ?

## 8. History/ Feedback

### a. Class Diagram

Graphical user interface, application, Teams

Description automatically generated

### b. Class Specifications

***ViewBookingDetailActivity Class***

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 1 | onClick(View) | Method that is called when the view(component) is clicked |
| 2 | onCreate(Bundle) | Method is used to start an activity |

***FeedbackActivity Class***

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 1 | onCreate(Bundle) | Method is used to start an activity |

***FeedbackDAO Class***

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 1 | createFeedback(int) | Method is used to create feedback for car |

### c. Sequence Diagram(s)

Graphical user interface, text

Description automatically generated

### d. Database queries

- INSERT INTO Feedback (IdA, IdC, content, rate) VALUES(?,?,?,?)

## 9. Car rental/ Find car

### a. Class Diagram

Graphical user interface, application, Teams

Description automatically generated

### b. Class Specifications

***FindCarActivity Class***

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 1 | onClick(View) | Method that is called when the view(component) is clicked |
| 2 | onCreate(Bundle) | Method is used to start an activity |

### c. Sequence Diagram(s)

A picture containing diagram

Description automatically generated

### d. Database queries

## 10. Car rental/View list car

### a. Class Diagram

Graphical user interface, application, Teams

Description automatically generated

### b. Class Specifications

***ViewListCarActivity Class***

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 1 | onClick(View) | Method that is called when the view(component) is clicked |
| 2 | onCreate(Bundle) | Method is used to start an activity |

### c. Sequence Diagram(s)

Text

Description automatically generated

### d. Database queries

## 11. Car rental/ View detail car

### a. Class Diagram

Graphical user interface, application, Teams

Description automatically generated

### b. Class Specifications

***ViewCarDetailActivity Class***

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 1 | onClick(View) | Method that is called when the view(component) is clicked |
| 2 | onCreate(Bundle) | Method is used to start an activity |

### c. Sequence Diagram(s)

Text

Description automatically generated

### d. Database queries

## 12. Car rental/ Car rental

### a. Class Diagram

Graphical user interface, application, Teams

Description automatically generated

### b. Class Specifications

***RentalCarActivity Class***

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 1 | onClick(View) | Method that is called when the view(component) is clicked |
| 2 | onCreate(Bundle) | Method is used to start an activity |

### c. Sequence Diagram(s)

Diagram

Description automatically generated

### d. Database queries

## 13. Car rental/ View feedback

### a. Class Diagram

Graphical user interface, application, Teams

Description automatically generated

### b. Class Specifications

***ViewFeedbackActivity Class***

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 1 | onClick(View) | Method that is called when the view(component) is clicked |
| 2 | onCreate(Bundle) | Method is used to start an activity |

### c. Sequence Diagram(s)

Diagram, text

Description automatically generated

### d. Database queries

## 14. Car rental/ Add driving license

### a. Class Diagram

Graphical user interface, application, Teams

Description automatically generated

### b. Class Specifications

***CarRentalDocumentActivity Class***

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 1 | onClick(View) | Method that is called when the view(component) is clicked |
| 2 | onCreate(Bundle) | Method is used to start an activity |

### c. Sequence Diagram(s)

Diagram

Description automatically generated

### d. Database queries

## 15. Individual

### a. Class Diagram

Graphical user interface, application, Teams

Description automatically generated

### b. Class Specifications

***IndividualActivity Class***

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 1 | onClick(View) | Method that is called when the view(component) is clicked |
| 2 | onCreate(Bundle) | Method is used to start an activity |

### c. Sequence Diagram(s)

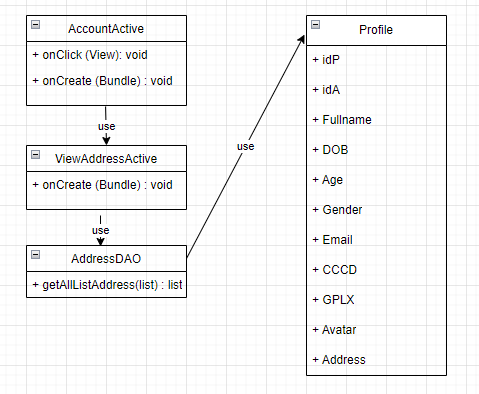
Text

Description automatically generated with low confidence

### d. Database queries

## 16. View address

### a. Class Diagram



### b. Class Specification

**AccountActive**

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 01 | onClick(View) | Method that is called when the view is clicked |
| 02 | onCreate(Bundle) | Method is used to start activity |

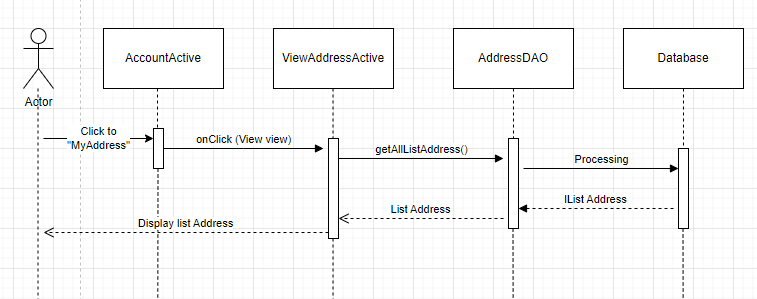
**ViewAddressActive**

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 01 | onCreate(Bundle) | Method is used to start an activity |

**AddressDAO**

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 01 | getAllListAddress(list) | Get all list address of account |

### c. Sequence Diagram(s)

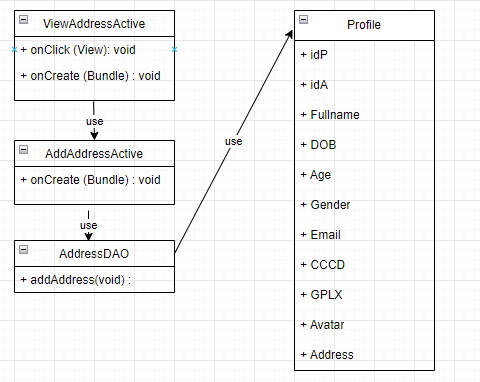


### d. Database queries

* Select \* from Profile where idA = ?

## 17. Add address

### a. Class Diagram



### b. Class Specification

**ViewAddressActive**

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 01 | onClick(View) | Method that is called when the view is clicked |
| 02 | onCreate(Bundle) | Method is used to start activity |

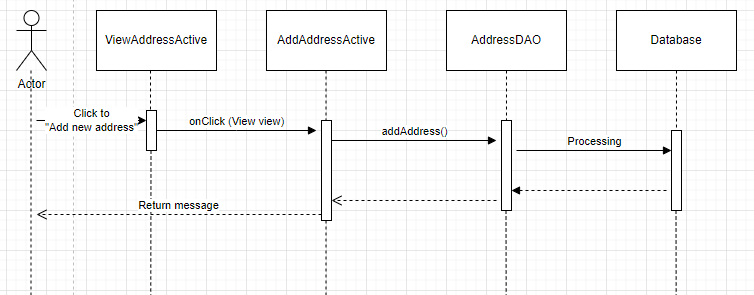
**AddAddressActive**

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 01 | onCreate(Bundle) | Method is used to start an activity |

**AddressDAO**

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 01 | addAddress(String):list | Add a new address into the table information . |

### c. Sequence Diagram(s)



### d. Database queries

* Insert into Information Address = ? where idA = ?

## 18. Account information

### a. Class Diagram

### 

### b. Class Specification

**AccountActive**

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 01 | onClick(View) | Method that is called when the view is clicked |
| 02 | onCreate(Bundle) | Method is used to start activity |

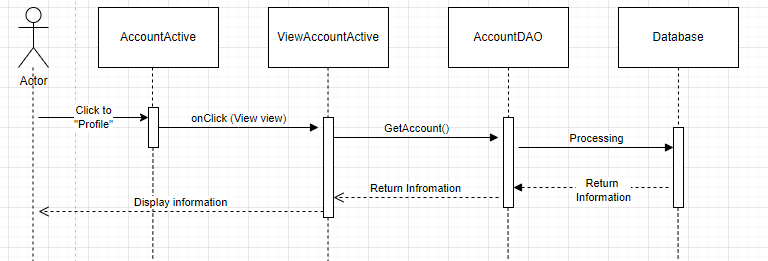
**ViewAccountActive**

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 01 | onCreate(Bundle) | Method is used to start an activity |

**AddressDAO**

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 01 | getAccount(int):Account | Get an Account object . |

### c. Sequence Diagram(s)



### d. Database queries

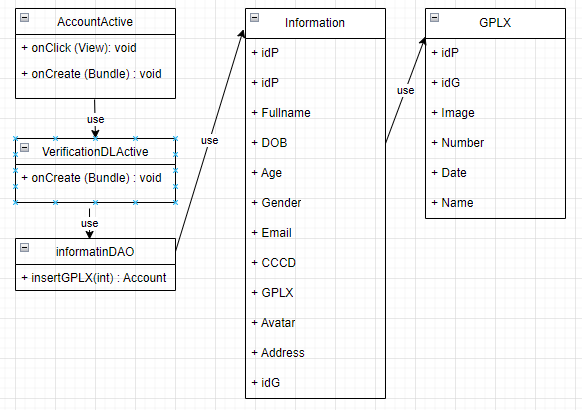
* Select \* from Account a

inner join Profile p

On a.idA = p.idA = ?

## 19. Verification driver’s license

### a. Class Diagram

****

### b. Class Specification

**AccountActive**

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 01 | onClick(View) | Method that is called when the view is clicked |
| 02 | onCreate(Bundle) | Method is used to start activity |

**VerificationDLActive**

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 01 | onCreate(Bundle) | Method is used to start an activity |

**informationDAO**

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 01 | insertGPLXt(int): void | insert driver’s license . |

### 

### c. Sequence Diagram(s)

### 

### d. Database queries

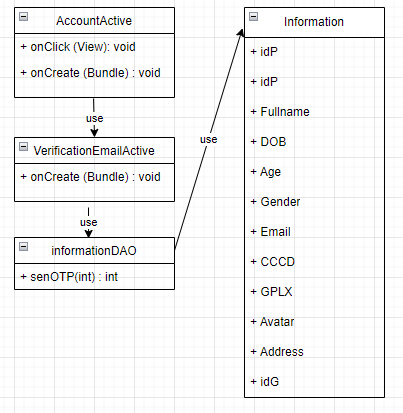
* Insert into GPLX g

inner join Information i

on i.idp = g.idP = ?

## 20. Verification email

### a. Class Diagram

****

### b. Class Specification

**AccountActive**

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 01 | onClick(View) | Method that is called when the view is clicked |
| 02 | onCreate(Bundle) | Method is used to start activity |

**VerificationEmailtActive**

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 01 | onCreate(Bundle) | Method is used to start an activity |

**InformationDAO**

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 01 | sendOTPt(int):int | Send code OTP to email . |

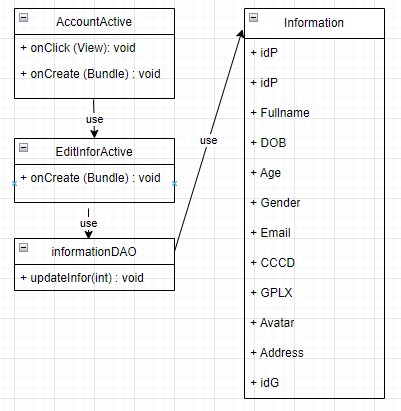
### 

### c. Sequence Diagram(s)

### d. Database queries

## 21. Edit information

### a. Class Diagram

****

### b. Class Specification

**AccountActive**

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 01 | onClick(View) | Method that is called when the view is clicked |
| 02 | onCreate(Bundle) | Method is used to start activity |

**EditInforActive**

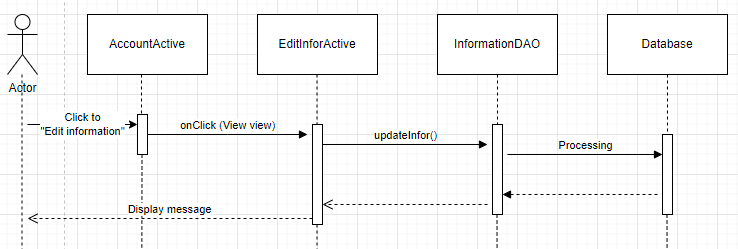
| **No** | **Method** | **Description** |
| --- | --- | --- |
| 01 | onCreate(Bundle) | Method is used to start an activity |

**InformationDAO**

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 01 | updateInfor(int): void | Update information of account . |

### 

### c. Sequence Diagram(s)



### d. Database queries

* Update Information

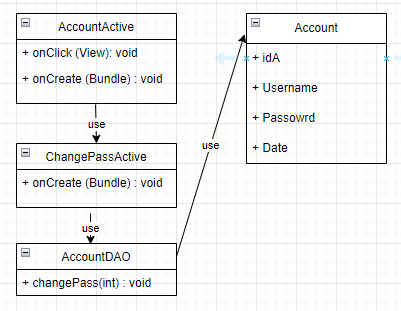
set Full name = ?, Age =?,

Gender = ?, Avarta = ?

where idP = ?

## 22. Change password

### a. Class Diagram



### b. Class Specification

**AccountActive**

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 01 | onClick(View) | Method that is called when the view is clicked |
| 02 | onCreate(Bundle) | Method is used to start activity |

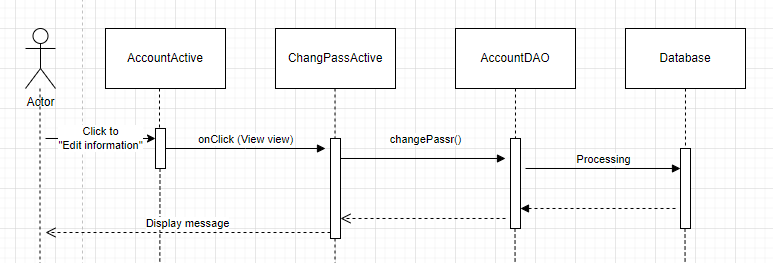
**ChangePassActive**

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 01 | onCreate(Bundle) | Method is used to start an activity |

**AccountDAO**

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 01 | changePass(int): void | Change password. |

### c. Sequence Diagram(s)



### d. Database queries

* update Account set password = ? where username = ?

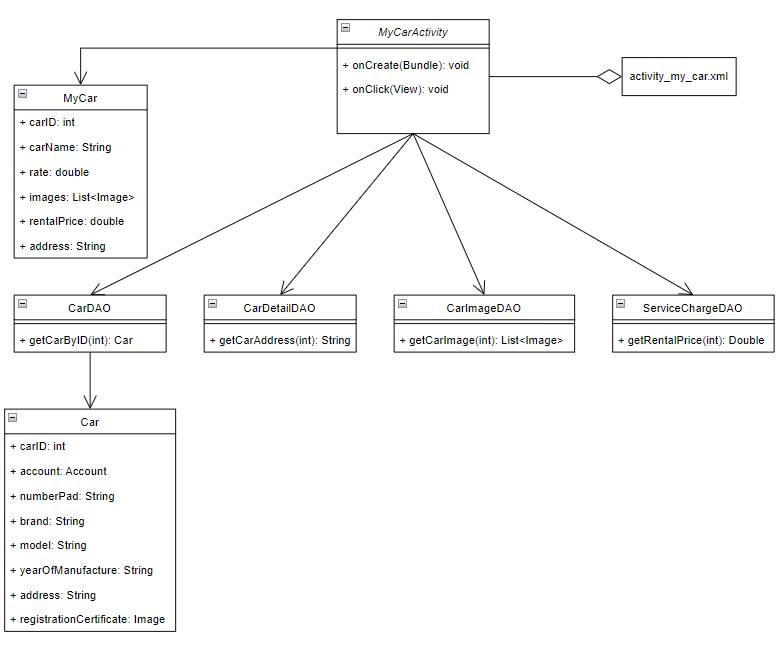
## 23. My Car

### a. Screen Design

**Table 1:** **My Car Definition**

| **No** | **Object/Control Name** | **Type** | **Required** | **Length** | **Description** |
| --- | --- | --- | --- | --- | --- |
| 1 | Back | Button | Yes |  | Display by icon left arrow |
| 2 | Screen name | Text | Yes | 20 |  |
| 3 | Add new car | Button | Yes |  | Display by icon plus |
| 4 | Car item | Button |  |  | Display car’s basic information |

### b. Class Diagram



### c. Class Specifications

***MyCarActivity***

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *onCreate()* |  |
| *02* | *onClick()* |  |

***CarDAO***

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *getCarByID()* | *Get the car from database by its id* |

***CarDetailDAO***

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *getCarAddress()* | *Get address information of the car by car’s id* |

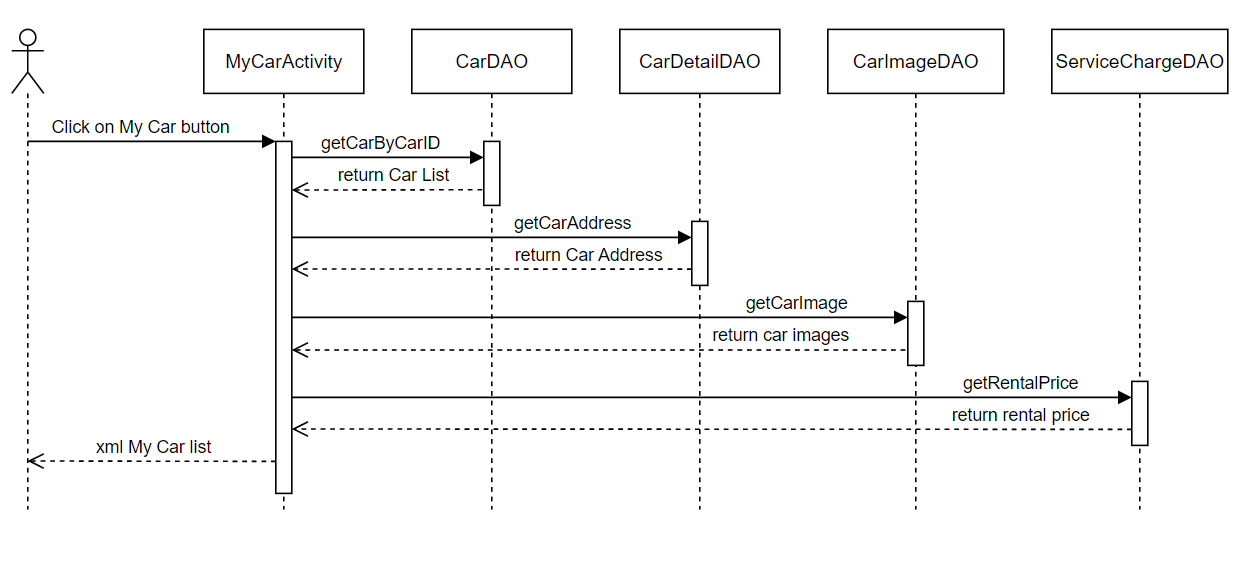
***CarImageDAO***

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *getCarImage()* | *Get all images of the car from database by car’s id* |

***ServiceChargeDAO***

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *getRentalPrice()* | *Get rental price of the car by car’s id* |

### d. Sequence Diagram(s)



### e. Database queries

*- select \* from Car where carID = ?*

*- select address from CarDetail where carID = ?*

*- select \* from CarImage where carID = ?*

*- select retal\_price from ServiceCharge where carID = ?*

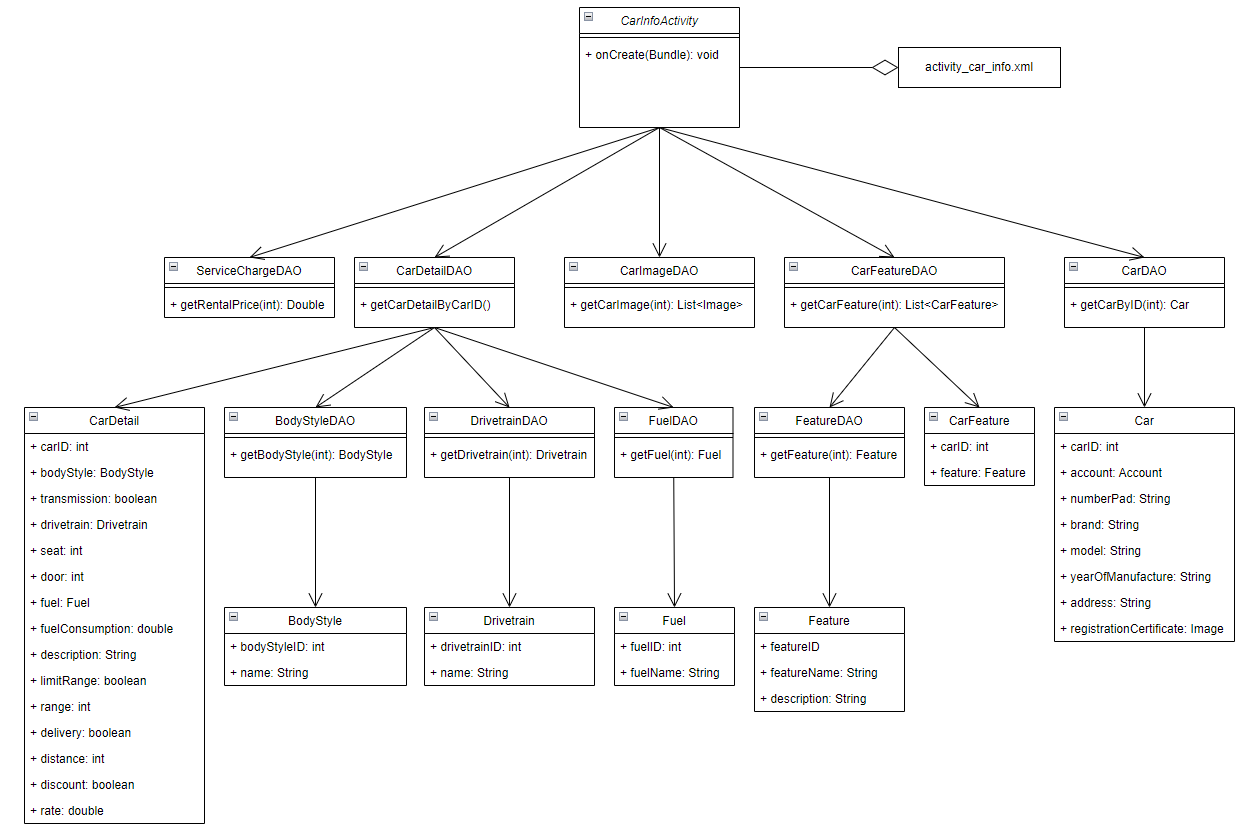
## 24. Car Information

### a. Screen Design

**Table 2:** **Car Information Definition**

| **No** | **Object/Control Name** | **Type** | **Required** | **Length** | **Description** |
| --- | --- | --- | --- | --- | --- |
| 1 | Back | Button | Yes |  | Display by icon left arrow |
| 2 | Screen name | Text | Yes | 20 |  |
| 3 | Image | Button | Yes |  |  |
| 4 | Previous | Button |  |  |  |
| 5 | Next | Button |  |  |  |
| 6 | Name | Button |  |  |  |
| 7 | Rental price | Text |  | 10 |  |
| 8 | Address | Text |  | 50 |  |
| 9 | Seat | Text |  | 2 |  |
| 10 | Door | Text |  | 2 |  |
| 11 | Fuel | Text |  | 20 |  |
| 12 | Transmission | Text |  | 20 |  |
| 13 | Body style | Text |  | 20 |  |
| 14 | Drivetrain | Text |  | 20 |  |
| 15 | Description | Text |  | 200 |  |
| 16 | Feature | Icon |  |  |  |

### b. Class Diagram



### c. Class Specifications

***CarInformationActivity***

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *onCreate()* |  |
|  |  |  |

***CarDetailDAO***

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *getCarDetailByCarID()* | *Get all details information of the car by its id* |
|  |  |  |

***CarImageDAO***

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *getCarImage()* | *Get all images of the car from database* |
|  |  |  |

***CarDAO***

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *getCarByCarID()* | *Get the car from database by its id* |
|  |  |  |

***CarFeatureDAO***

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *getCarFeature()* | *Get the list of features of the car* |
|  |  |  |

***ServiceChargeDAO***

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *getRentalPrice()* | *Get rental price of the car* |
|  |  |  |

***BodyStyleDAO***

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *getBodyStyle()* | *Get body style of the car* |
|  |  |  |

***DrivetrainDAO***

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *getDrivetrain()* | *Get drivetrain of the car* |
|  |  |  |

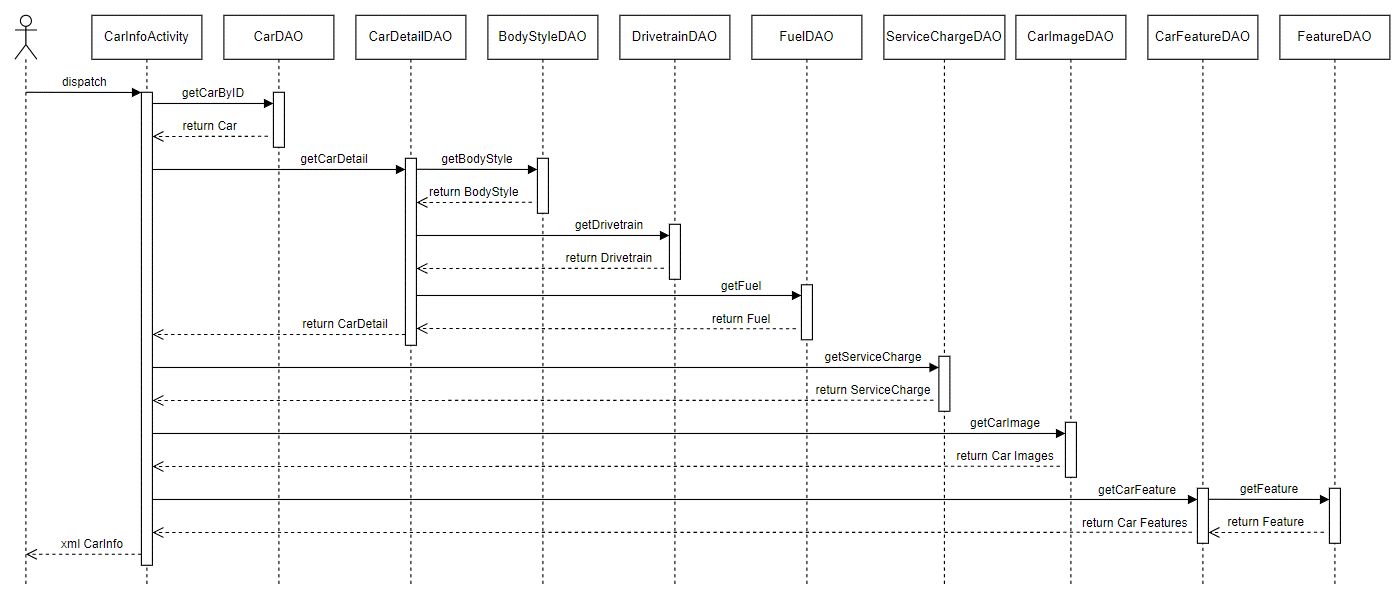
***FuelDAO***

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *getFuel()* | *Get fuel type of the car* |
|  |  |  |

***FeatureDAO***

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *getFeature()* | *Get the feature’s information* |
|  |  |  |

### d. Sequence Diagram(s)



### e. Database queries

*- select \* from Car where carID = ?*

*- select \* from CarDetail where carID = ?*

*- select \* from BodyStyle where bodystyleID = ?*

*- select \* from Drivetrain where drivetrainID = ?*

*- select \* form Fuel where fuelID = ?*

*- select \* from ServiceCharge where carID = ?*

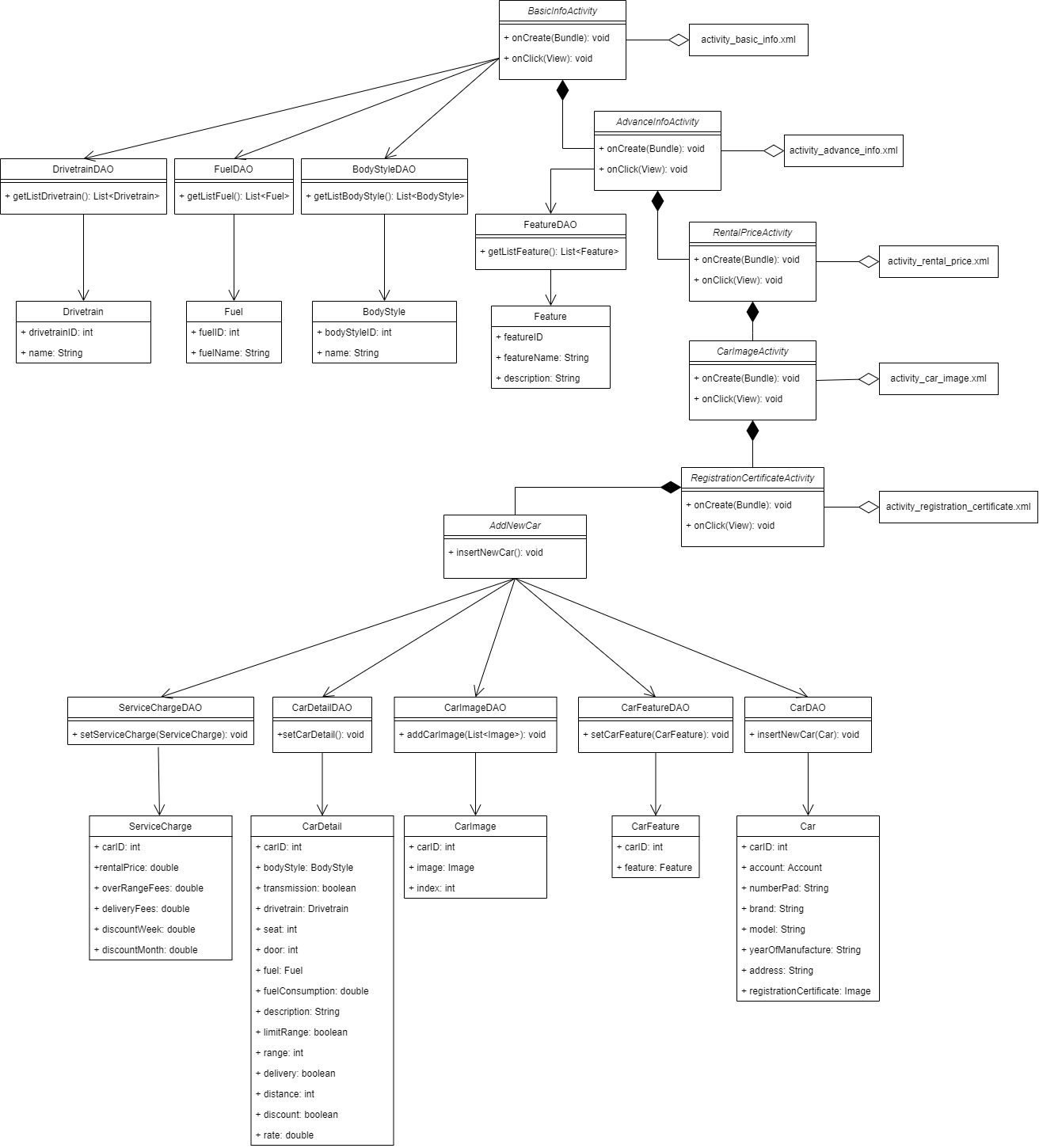
*- select \* from CarImage where carID = ?*

*- select \* from CarFeature where carID = ?*

*- select \* from Feature where featured = ?*

## 25. Add New Car

### a. Class Diagram



### b. Class Specifications

***AddNewCar***

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *insertNewCar()* | *Insert new car’s information into the database* |
|  |  |  |

***BasicInfoActivity***

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *onCreate()* | *Create graphic user interface* |
| *02* | *onClick()* | *Direct user to next screen* |

***AdvanceInfoActivity***

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *onCreate()* | *Create graphic user interface* |
| *02* | *onClick()* | *Direct user to next screen* |

***RentalPriceActivity***

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *onCreate()* | *Create graphic user interface* |
| *02* | *onClick()* | *Direct user to next screen* |

***CarImageActivity***

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *onCreate()* | *Create graphic user interface* |
| *02* | *onClick()* | *Direct user to next screen* |

***RegistrationCertifiateActivity***

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *onCreate()* | *Create graphic user interface* |
| *02* | *onClick()* | *Direct user to next screen* |

***DrivetrainDAO***

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *getListDrivetrain()* | *Get list of all drivetrains* |
|  |  |  |

***FuelDAO***

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *getListFuel()* | *Get list of all fuel type* |
|  |  |  |

***BodyStyleDAO***

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *getListBodyStyle()* | *Get list of all bodystyle* |
|  |  |  |

***FeatureDAO***

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *getListFeature()* | *Get list of all feature* |
|  |  |  |

***CarDAO***

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *insertNewCar()* | *Insert new car in to the database* |
|  |  |  |

***ServiceChargeDAO***

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *setServiceCharge()* | *Set service charge of the new car in database* |
|  |  |  |

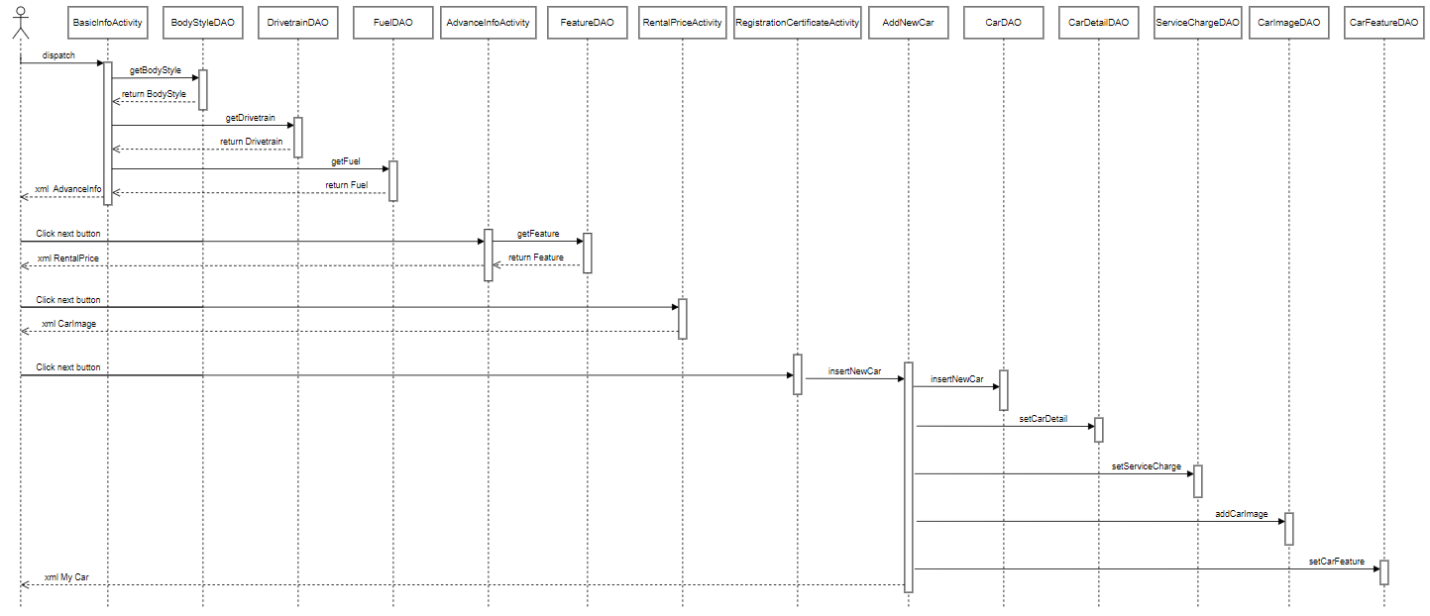
***CarDetailDAO***

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *setCarDetail()* | *Set details information of the new car in database* |
|  |  |  |

***CarImageDAO***

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *addCarImage()* | *Add new car’s images into the database* |
|  |  |  |

### c. Sequence Diagram(s)



### d. Database queries

*- insert into Car[accountID, number\_pad, brand, model, year\_of\_manufacture, address, registration\_certificate] values (?,?,?,?,?,?,?)*

*- insert into CarDetail[carID, body\_style, transmission, drivetrain, seat, door, fuel, fuel\_consumption, description, limit\_range, range, delivery, distance, discount, rate] values (?,?,?,?,?,?,?,?,?,?,?,?,?,?,?)*

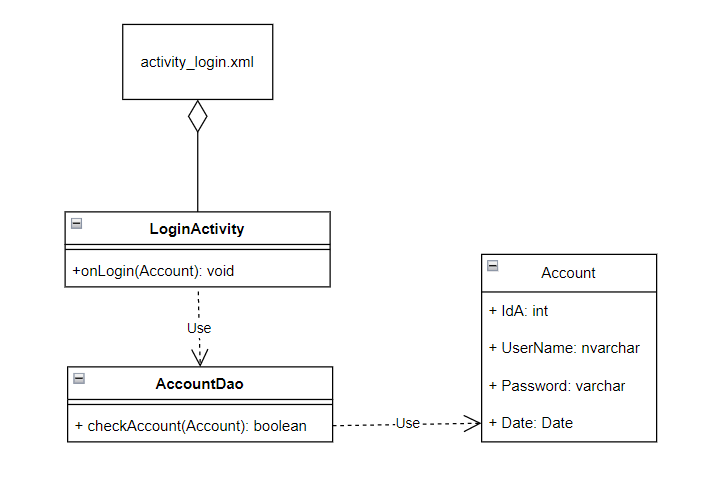
*- insert into ServiceCharge[carID, rental\_price, over\_range\_fees, delivery\_fees, discount\_week, discount\_month] values (?,?,?,?,?,?)*

*- insert into CarImage[carID, image, index] values (?,?,?)*

*- insert into CarFeature[carID, featured] values (?,?)*

## 26. Login

### a. Class Diagram



### b. Class Specifications

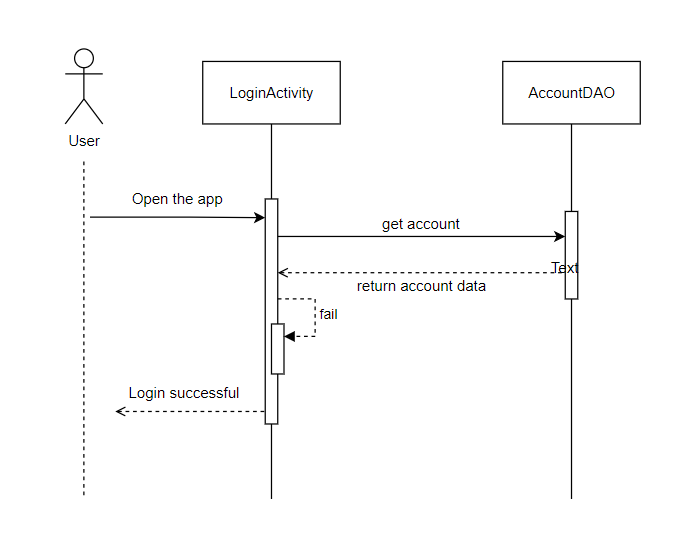
***LoginActivity class***

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 01 | onLogin(View view) | Method for users to login to the app |

***AccountDAO Class***

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 01 | checkAccount(Account) | get username and password data |

### c. Sequence Diagram(s)

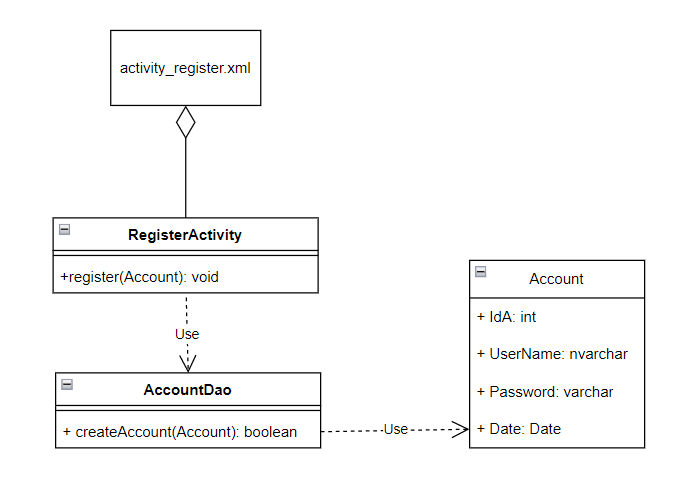


### d. Database queries

* select \* from Account where username = ?

## 27. Register

### a. Class Diagram



### b. Class Specifications

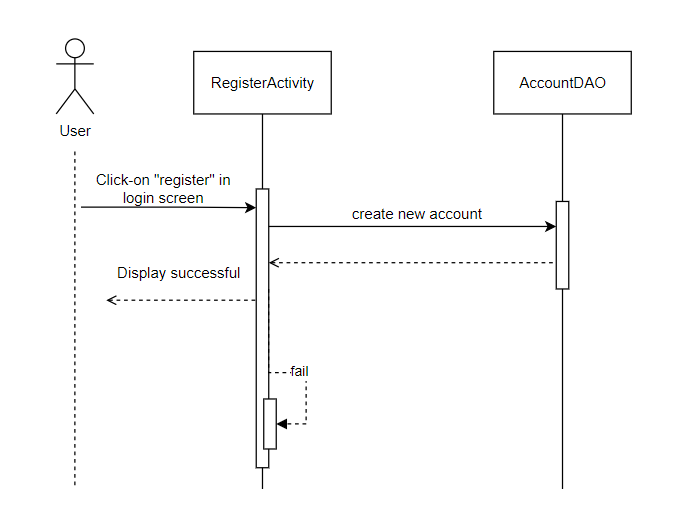
***RegisterActivity class***

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 01 | onRegister(View view) | Method for users to register account |

***AccountDAO Class***

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 01 | createAccount(Account) | Create a new data for the account table |

### c. Sequence Diagram(s)

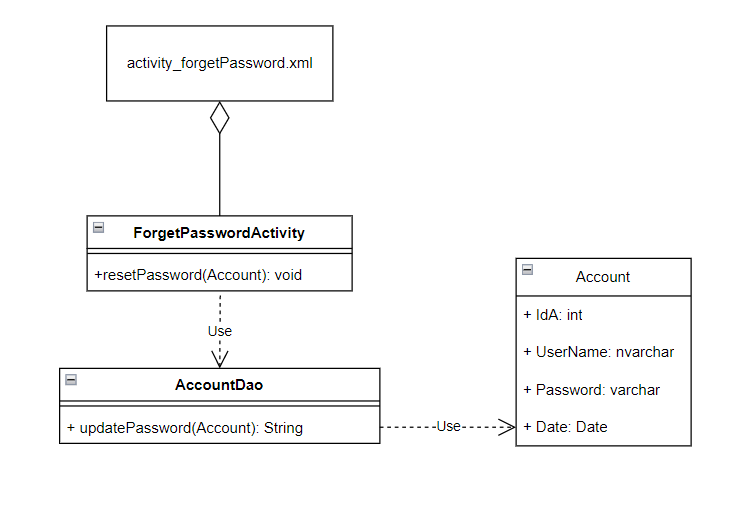


### d. Database queries

* INSERT INTO [dbo].[Account] username, password Values(?,?)

## 28. Forget Password

### a. Class Diagram



### b. Class Specifications

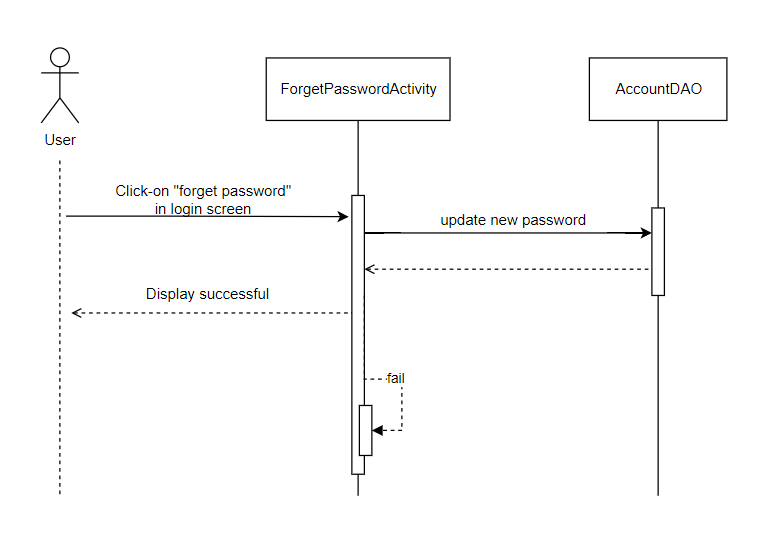
***ForgetPasswordActivity Class***

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *resetPassword(Account)* | *Reset password* |

***AccountDAO***

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *updatePassword* | *Allow user to change new password* |

### c. Sequence Diagram(s)



### d. Database queries

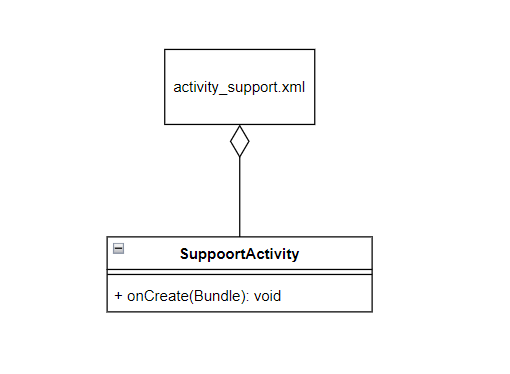
*UPDATE* [dbo].[Account]

*SET password = ?*

*WHERE username = ?;*

## 29. Support

### a. Class Diagram

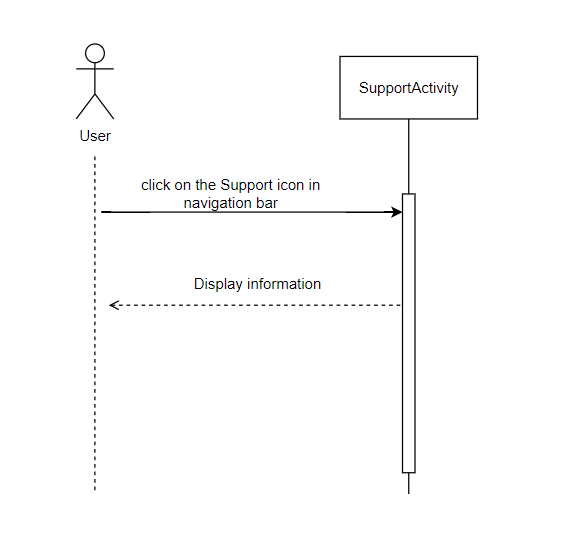


### b. Class Specifications

***SupportActivity class***

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *onCreate* | *Display some information related to the app to the user, for example phone number, email, ....* |

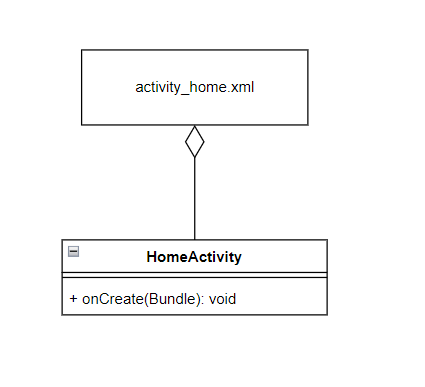
### c. Sequence Diagram(s)



### d. Database queries

## 30. Home

### a. Class Diagram



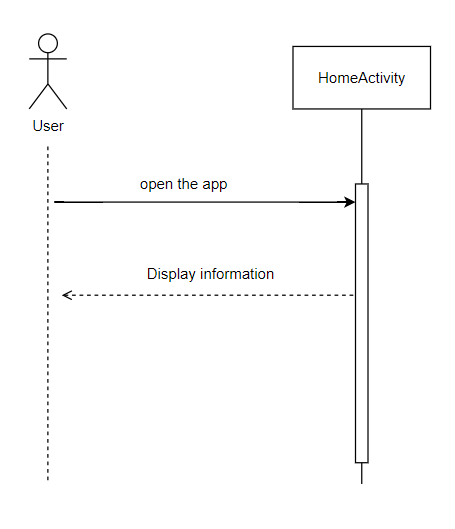
### b. Class Specifications

***HomeActivity Class***

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *onCreate* | *Show outstanding features* |

#### 

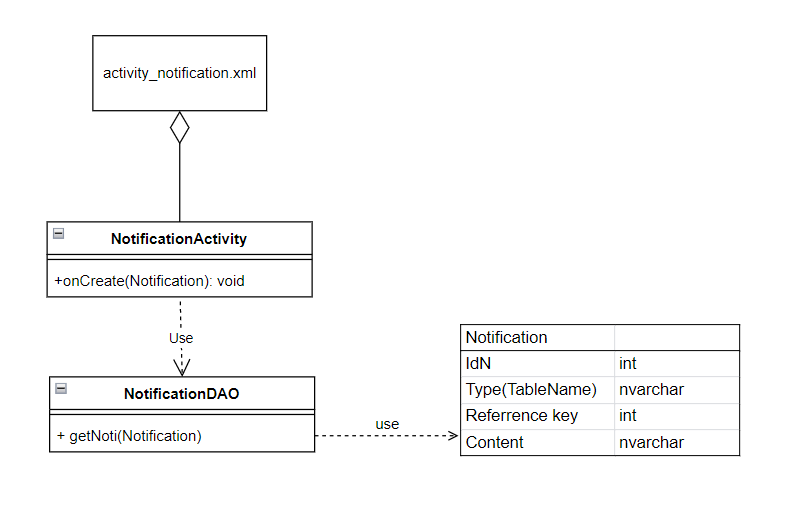
### c. Sequence Diagram(s)



### d. Database queries

## 31. Notification

### a. Class Diagram



### b. Class Specifications

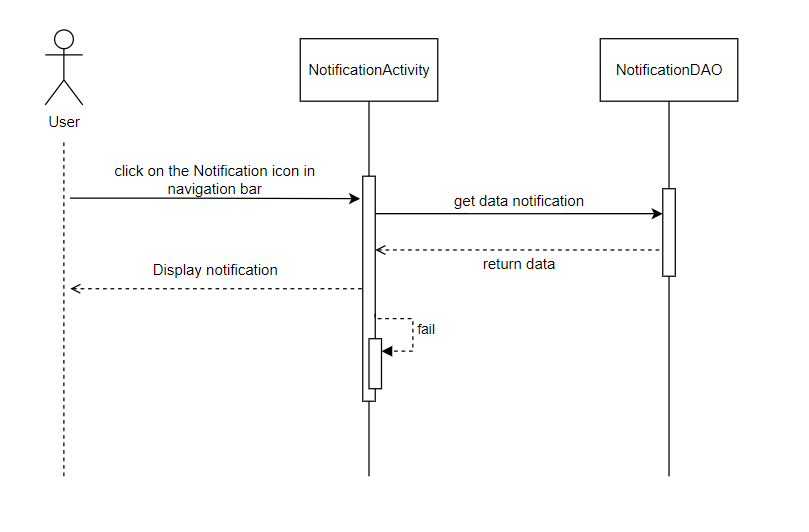
***NotificationActivity Class***

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *onCreate(Notification)* | *display the notification title and a small part of the content* |

***NotificationDAO***

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *getNoti(Notification)* | *get data of notification* |

### c. Sequence Diagram(s)

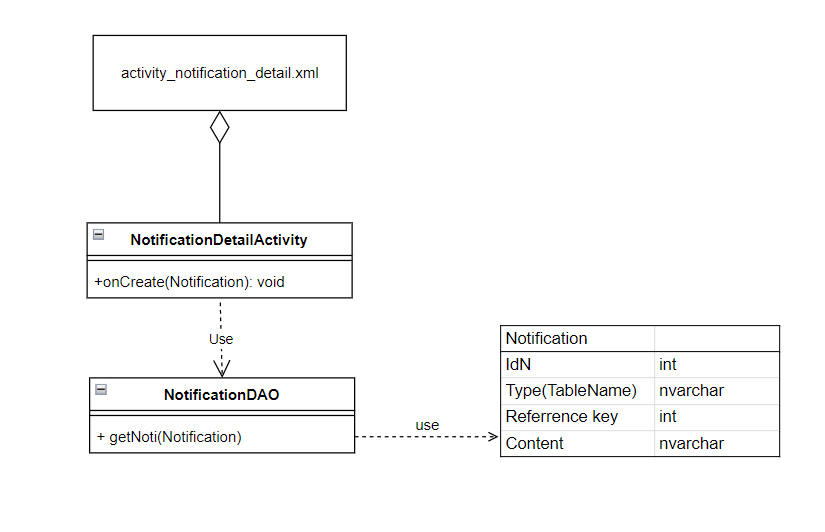


### d. Database queries

select \* from Notification

## 32. Notification Detail

### a. Class Diagram



### b. Class Specifications

***NotificationDetailActivity Class***

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *onCreate(Notification)* | *display the notification title and all of the content* |

***NotificationDAO***

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *getNoti(Notification)* | *get data of notification* |

### c. Sequence Diagram(s)

### d. Database queries

select \* from Notification