VEHİCLE DEALER PROJECT

Oğuzhan Karakaya

Table of Contents

1. Introduction

1.1 What the Problem is

Our company which we made program for, was enterprise and they want to become more professional so beginning of that process was creating a software which managers can control employees and clients will able to reach lots of company's opportunity. Company is about car selling and fixing cars. They have got employees to sell cars and each car has specific price but sometimes some employees were trying to sell lesser price than price that company specified. Firstly company want to stop this disadvantage so that team created a desktop application which have a system includes the car prices. Dealer selects car from application then when one sell the car, car is delivered automatically. Secondly company was complaining about the number of employees.They want automation system to sell cars and according to this demand we created an option for clients on desktop application. Clients can buy car or motorbike. Nowadays company is trying to open new segment about renting car. Team is waiting them to create that option on desktop application. Thirdly, company want ordering for their service. All service employees were working according the car in queue , there was no scheduling so team created three app for clients and service employees. Clients can request an appointment from service by using webpage. They have to sign in before taking appointment

and then can request an appointment. According to available time, the appointment is gonna be given to client, one can control the appointment on mobile application. When the time comes, client goes the information desk and one tells the problem in car. Employee in information desk uses the desktop application to write this problem and when telling problem is ended, employee clicks send button then this problem is saved on database. Furthermore service employee can see that problem when car comes.Three apps are requested user register and signing.

1.2.Goals For The Project

The software present the company many opportunity ,such as looking up the available cars, read the service report which is written by user and earning more money.

A dealer can look up the available cars in gallery but it has to login the app in desktop application. Then application shows a list which includes cars. If dealer's customer selects one of the car in list, it can sell by clicking the button near the list. After a while, the customer who bought a car from the company , comes the gallery to hand over its car to service person to have fixed. Before servis person take the car, client writes a report about problem in car. The report can be written in desktop application. While it is waiting, it has a chance to look up new cars on webpage or mobile app. It has one more big opportunity which

it can arrange the service time by mobile app , before it comes so it doesn't has to wait long time.

The user is be able to search car by price, model, year, motor performance etc. If this is the first time of the user, it has to fill a form about its information. Whatever he bought or not , company can send information about campaigns or sales. When the service time comes, company can send a warning. Form is filled in desktop application.

Every company wants to explain their information to their client. So client can reach information in the webpage.

1.3 STAKEHOLDERS

Different types of stakeholders can be noted when they come to our software. The most obvious are those that required for this software: CEO and company board. In every stage of progress we kept in touch with CEO and necessary persons. At the beginning of the project, only desktop application was wanted from us. That was the beginning of the project, however, during the ongoing process mobile application was required to make necessary process better and more comfortable using. In this process, using of the database was not considered but database was added to the project by request of CEO. CEO

submitted that by whom the cars on sale are bought is important for tributing to the government and this vehicles would faciliate the customs procedures. During this process, for a while, we decided how we would use the database and which database would be. As a result, after this process we began to the design of the project and database was added as required. It was demanded by the manager that desktop application can be used by both customer and employee. According to this demand, the ability to data input about user by customer and employee was aadded. No special code was arranged for employees to log in. This arrangement is delayed for another time. Employee is going to get a commission for each car sold and this is going to be added to the application. It was asked by the manager to add a login input to the user interface to view service records. According to the result of this input , a system, which can control the stuff used by service workers, is going to be made. It was asked by CEO to make a website to get more information about company and especially to find the place of company and service.

1.4 Motivation for the Project

The want of help to a developing company was the focal motivation point of this project. While the company was putting into practice some process newly, we wanted to help them with our skills. We determined the inadequate points in the meeting that held with them and we concentrated on these points.

The meeting, which we held, contributed to our motivation. Throughout the project we fulfilled the basic needs and of course there are some missing points but our project runs as requested. In this way it is going to help them. As they develop, our project is also going to develop correspondingly. Throughout this project, we improved ourselves, too. During improving process we increased our working place in our team and especially we managed to run projects synchronously on different platforms. Honestly the requests which was changing throughout the project demoralized us a bit but it didn't lessened our ambition for the project.

1.5 Process Flow Preview

We gained some speed for proceeding during this process. Main needs of this project were not much but extra works wanted were so much. In consequence of the meeting, we arranged how it was wanted . And we finished much earlier. At this remained time, we started to add add-on to develop the project as if everyone, who had specialized on specific fields, had garnished a cake. Design patterns, themes and new outlook were added to the content of the project, while we added database, local memory and user process to the code in platforms. The project was finished before the determined date and we handed over it. The things required extra are in the project. However, for this additions, fundamental information is supposed to be given by the company and during this process, company is supposed to fulfill the need.

2. Analysis and Design

2.1 Plan for Requirements Engineering

Inception Task:

The aim for the beginning is to create application to buy or rent a car easily. There are some applications for pre-owned vehicles. But this application should be belong a gallery.Prefencially car gallery should have lots of vehicle.Stakeholders thought this software brings statistics about finance as well.Then customer can easily look at the vehicle on sale ,or calculate credit if he does not afford to pay with cash.We stakeholders thought that it should be more than a buy,sell application.

Elicitation Task:

Our aim at this part is to identify the problem, search solutions, and talk amongst each other on the many different approaches. Meetings are arranged with the stakeholders in order to get a more clear understanding.The plan is merge all the features in one application In this reason,requirements analysis were taken.Standarts were composed.

Elaboration Task:

Scenarios were created to describe and aid in understanding
how the customer will interact with the application and how the dealer will interact with the software.Features were defined to code simply.Any attributes are to be defined as

well as how each function interacts with one another.Database system was designed to give meaning to data.

Specification Task:

During the project, software requirements specification template were created.This is useful for who writes code.Software features,user interfaces,classes and database are noted in template how coders design these parts.

Validation Task:

This is a user interface application.In case a invalid input,there must be validation.This is provided by testing the software.And it also has information part for developers to see how this code was designed.

Requirements Management:

Any potential changes would be looked over, discussed and determined if the time allotted for the construction of the project can allow for such a change.There was no plan to make a webstite and android application.These were assinged to developers.Statistic screen was removed.This changes were agreed all of the participants on the projects.

2.2 Functional Requirements

Hardware Requirements:

The software should be ran on any sort of desktop or laptop environment and any operating system.As well it should be ran on android based mobile phones.

User Interface - Primary Tasks:

- Show all avaiable purchasable motorcycle and cars.
  - Connect vehicles with database,sort them according to desired features
  - List vehicle features like year,motor,make etc.

- Select desired vehicle to buy
  - Click button buy then if client's need take credit,calculate then remove selected car from purchasable vehicle list.

- Register to reach application features.
  - Without registeration , it is not allowed to see vehicles and not allowed for sell vehicle as a dealer.

User Interface - Secondary Tasks:

- Allow the user select bank to get credit.

  - Show list of bank with credit proportions and desired amount of money.

- Store customer information in the database.

  - When registering, the customer will provide their information in the form . This information will be sent securely and stored in the database

- For client's send the car to service without entering any other information.

Android side User Interface - Primary Tasks:

- Show avaiable vehicle with its features.
- List vehicle according to their made and some features.
- Get information from user then arrange service appointment.

Android side User Interface - Secondary Tasks:

- Information of the user and service date is printing at the end of the service part to remind the information to user.

- Filter vehicle according to user selection.

2.3 Non Functional Requirements

Performance Requirements:

- Connect database fast.
- Sort data fast up to filter selection.

Security Requirements:

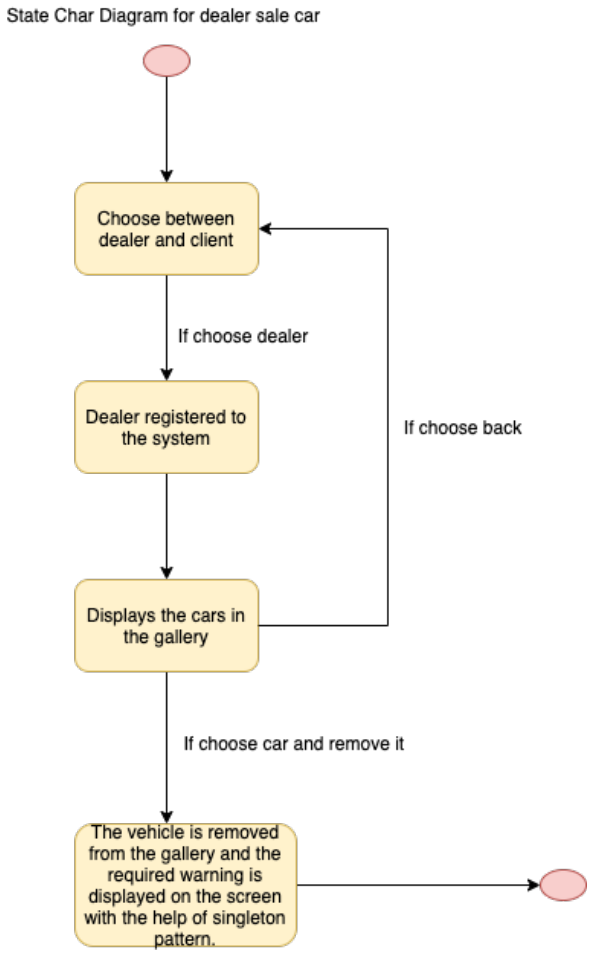
- Keep and store password and other information of the user and dealer safely.
- Prevent any potential threats such as SQL injections through the forms or search boxes.

Quality Attributes:

- Use object oriented principles to have a standart.
- Maintain readable content
- Design user friendly interfaces.

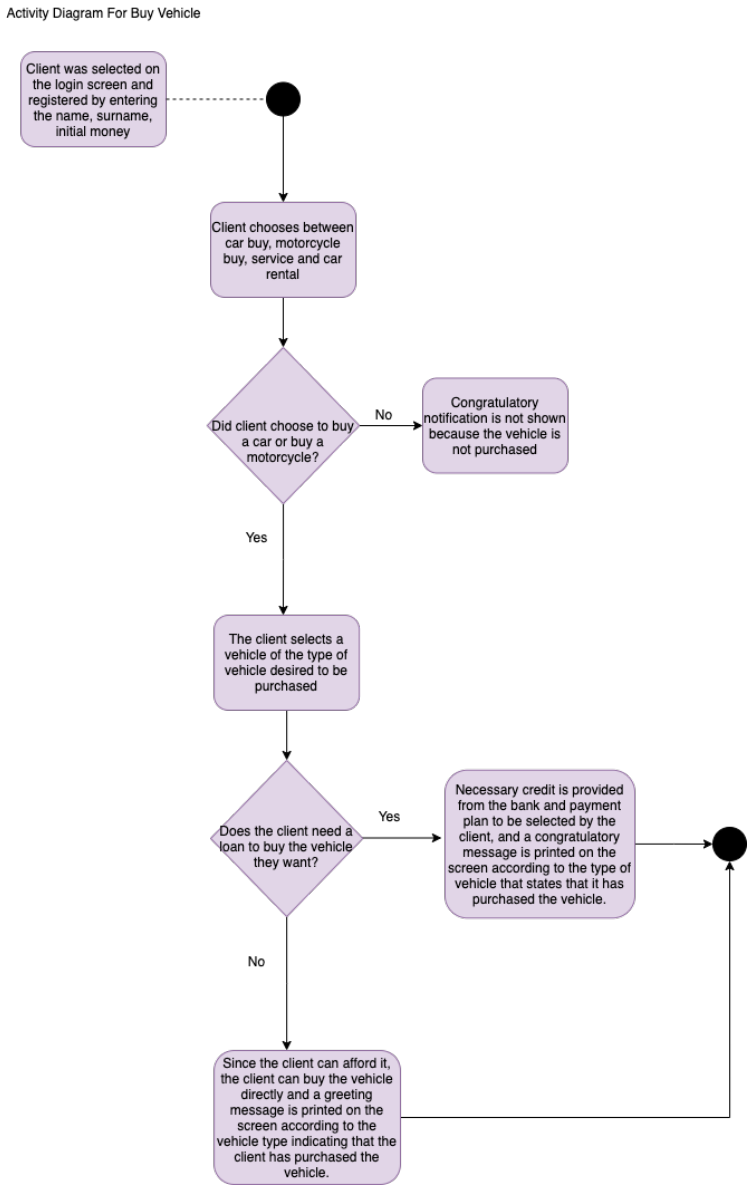
2.4 Use Cases and Diagrams

State Diagram



In this example of the state diagram shows that shows dealer's vehicle sales. At first, the dealer creates and logs in the program. If dealer wants to sell the vehicle dealer sees in the list, dealer then touches it and completes the sale of the vehicle after seeing the vehicle features. In this case, a warning notification appears on the screen (This warning notification was showed thanks to class created with Singleton Pattern.)

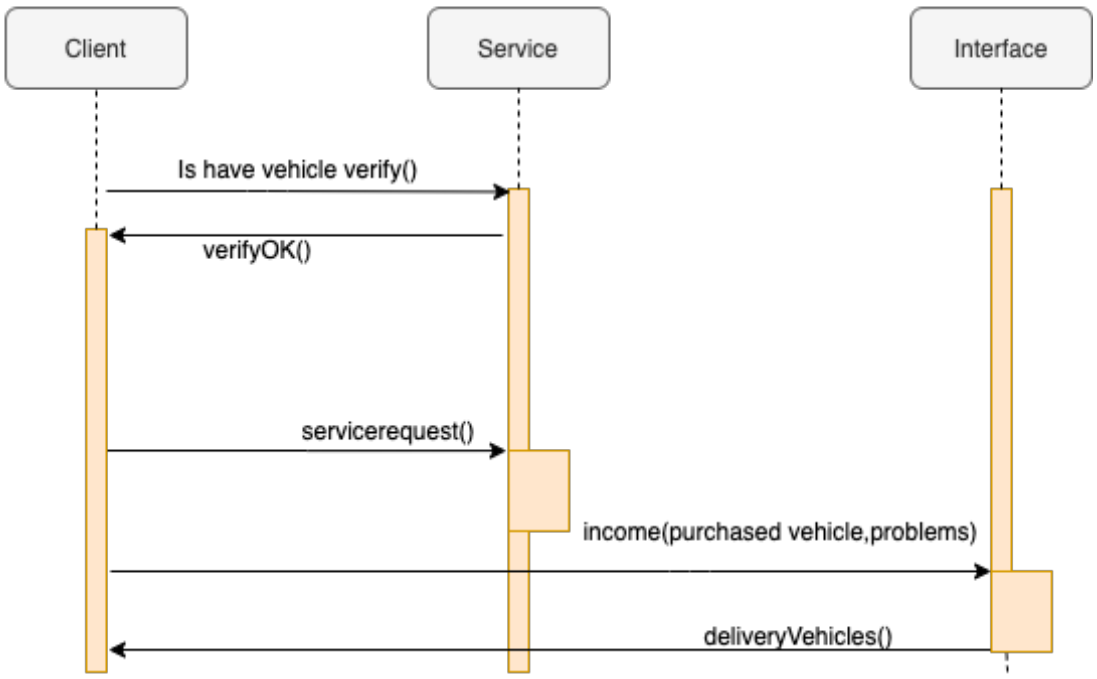
Activity Diagram



Client subscribes to the application and then logs on. The client clicks the option to buy a car or motorbike, and if he doesn't have money, he is directed to the credit department. If a car or motorbike is purchased as a result of this orientation, the greeting interface message is withdrawn from the main function, and the car result or motorcycle result in the function is returned with the type .Greeting message appears on the screen(See page 24-25)

Sequence Diagram

Sequence Diagram for vehicle service



This Sequence diagram shows the relations between client and vehicle service .If client have vehicle apply for the vehicle service. The client chooses the tool client wants to give to the service and the client reports the problem about the tool to the interface.Then the vehicle is delivered to the customer and the delivery is reported by a message.

3.Project Plan

3.1 Task Descriptions

Database Creation

A database is created using the models to provide storage for customer information, vehicle information, and dealer information. Testing of the database is ensured at this point**.**

Client Software Creation

The software part that is to be used by the client will be designed using Java using the guide of the mockups, requirements, and models.Application is working like vehicle browser.

Client informations like name,surname,born date,email are firstly taken after vehicle selection,cash money which client gives and up to this amount,credit has been calculated.

Testing

Testing will be implemented on both the MocOs and Windows operationg system.And as well developer ides Intelijj and Eclipse were tested if there is an error. Android part were written and testes in android studio.Any bugs or errors that may occur identified and resolved.

Finalization and Reports

All testing and function processes are finished at this stage. Reports will be created to ensure all information and functionality.

3.2 Task Assignment

TASK\TIME	February	March	April	May
Project finding				
Requirements Analysis				
Desktop Application				
Android Application				
Testing				
Use case diagram designing				
Database desinging				
Interface Designing				
Report Writing				

Tasks and its time period was colored.Every single color shows different task.White color means task has beeen finished.All team members work on every single task.In this reason assignments are not splitted.

3.3 Deliverables and Milestones

We had four major Milestones in this project:

1. Completion of Mockups and design diagrams.
2. Completion of Desktrop application.
3. Completon of android application.
4. Testing all software.

These milestones were all completed on schedule.And android part is not planned at first.This is additional requirement for this project.

Our four corresponding deliverables (respectively) in this project were as following:

- A finished and easily navigational GUI (Graphical user interface).
- Home work team studying were managed in this conditions.

3.4 Project Schedule

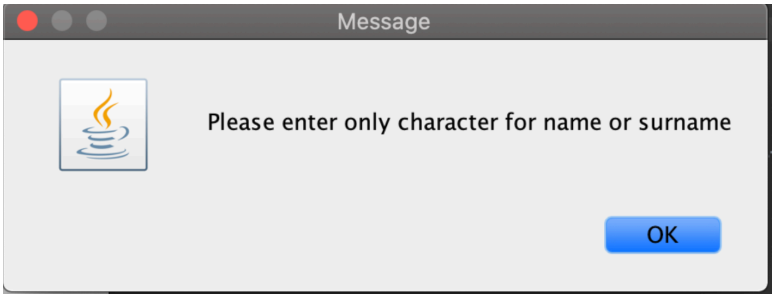
The first month of the project start date (end of the February) was decided on what kind of application should we write.Special interest was an important point.On March requirements were designed and up to this design desktop software was started to be written.Then software mainly has been finished before schedule.Thats why we add additional softwares.On April android design and coding were finished.At the end we have still time to do.We decided to add database to store vehicle and user information.Apart from testing and reporting everything almost is done.On may testing,some interface fixing,and reporting has been finished.We finished all the projects late May.

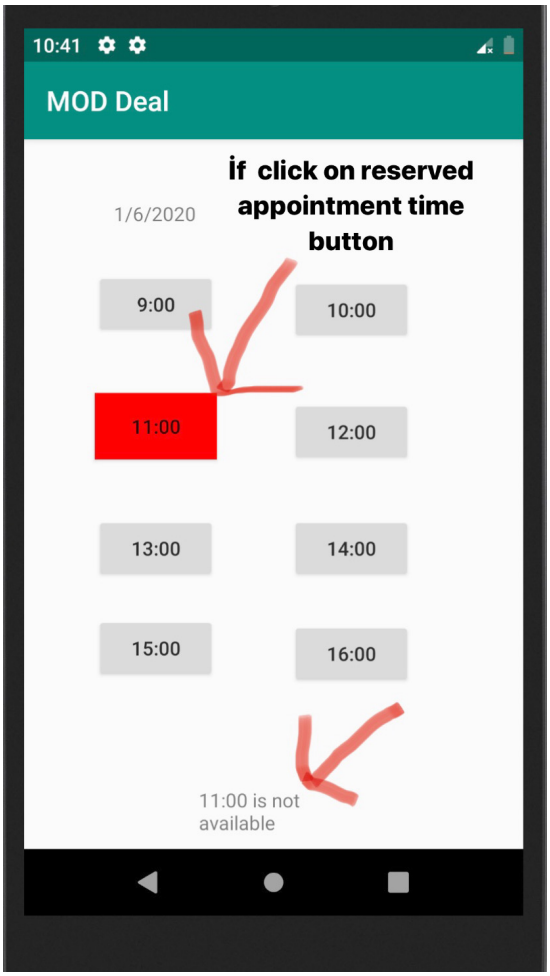
4.Testing

4.1 Test Cases

The following are instance of test cases we implemented:

- All fields on page (computer app and mobile app) should be aligned properly
- Input boxes was controlled by to their properties .(number box can not accept string)
- Checks full dates in service appointment section in mobile application and does not allow to make an appointment on full dates
- The system does not sell vehicles if you do not have enough money when controlled from a computer application and you do not want to take credit
- Checked all pages for broken links
- Checked all pages for broken images
- If the inputs are not entered according to the rules or if the input boxes that should not be selected are selected, the required error messages are displayed.





4.2 Testing Schedule

The test started with the start of the project,

By starting the test early, the errors were caught early and corrected immediately.

5.Conclusion

5.1 Main Purpose

The MOD car dealer is a car gallery app. In this gallery, the user can reach some car brands and the types of cars they want to achieve. Each type of car has its own characteristics: engine volume, type of fuel used by the car, year of manufacture, gear, price, etc. The same applies to motor bikes. Users and employees can easily handle their work using this application from various devices.

In addition, this application can be easily adapted to many galleries.Mod gallery App with an user-friendly application that would allow for customers to access and use on a wide range of devices: desktops, laptops, mobile devices.

User can look the vehicle and motorcycle models available from the mobile app. Before coming to the gallery, he can get ideas about the vehicle stock in the gallery. You can also get a service appointment from this app with the information required for your individual vehicle. In laptop and desktop applications, necessary information can be obtained for both the client and the gallery owner, and it is a computer application compatible with operating systems such as Mac os too .

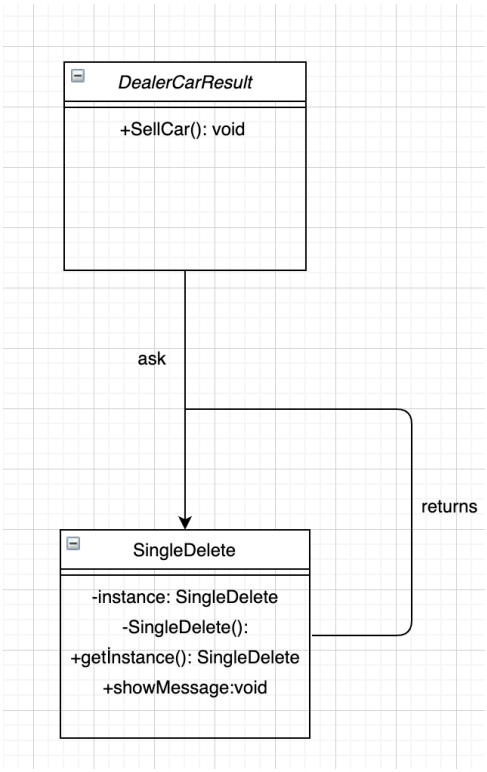
In this application, unlike other versions, vehicles can be sold. Bank loan application required for the customer can also be made. We think that our mobile, desktop computer, laptop and web application is complete, only pictures and image packages will be added where necessary.

5.2 The Team and the SE Process

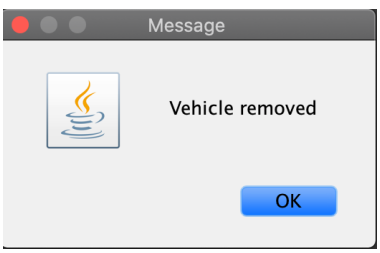
The software engineering process we used was on grouping. It was accomplished by grouping the required fields and combining them gradually during the project period. There have been separate studies on desktop and mobile app. In this way, the project construction time was reduced. The necessary improvements were added to at the our meetings.

6.Descriptions of design pattern usage

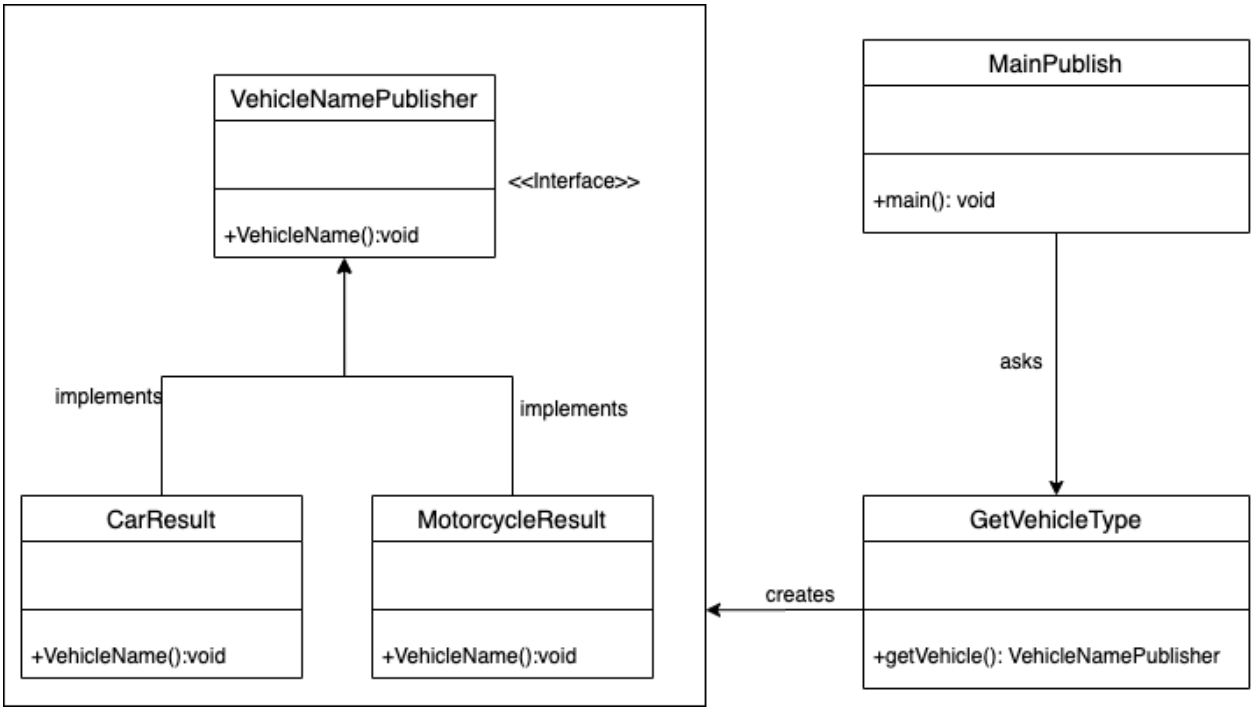
6.1 Singleton Pattern



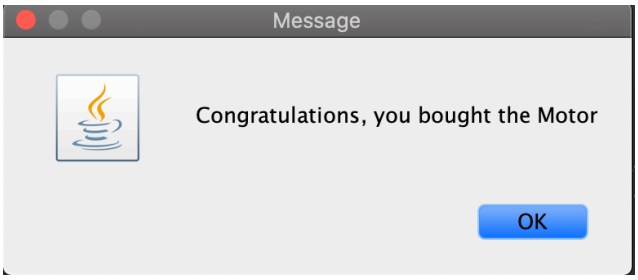
This pattern involves a single class which is responsible to create an object while making sure that only single object gets created. This class provides a way to access its only object which can be accessed directly without need to instantiate the object of the class.The SingleDelete class was created with the Singleton Pattern.When the dealer sold the vehicle thanks to the SingleDelete class, the notification screen showed that the vehicle was removed from the gallery stock.

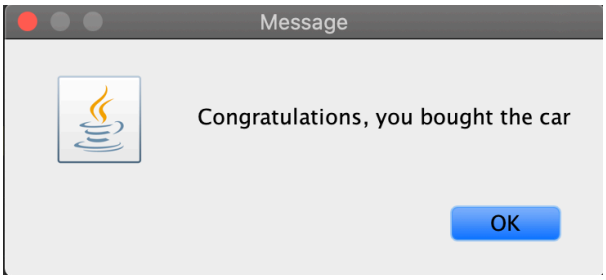


6.2 Factory Pattern

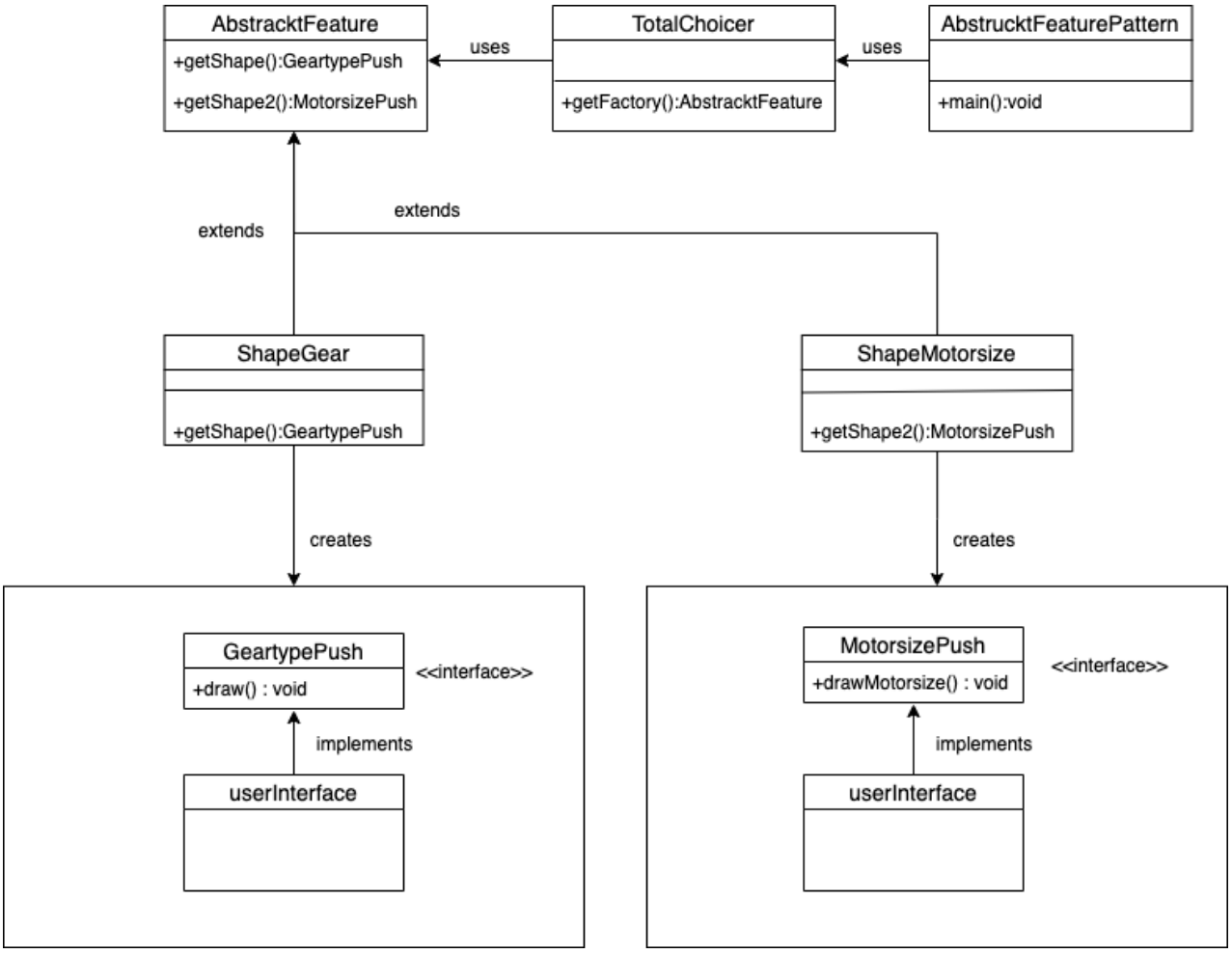


In Factory pattern, we create object without exposing the creation logic to the client and refer to newly created object using a common interface. VehicleNamePublisher interface and concrete classes that implementing the VehicleNamePublisher interface was created.Objects related to the vehicle type are created in the Getvehicletype class to be used in the MainPublish class according to the vehicle type.When the vehicle is purchased, the congratulatory messages are called with the function created from the interface according to the vehicle type.(The main function which belong MainPublis class is called in Credit class when congrat message is reflected .)

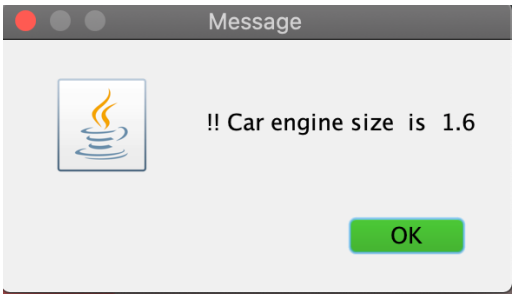




6.3 Abstract Factory Pattern



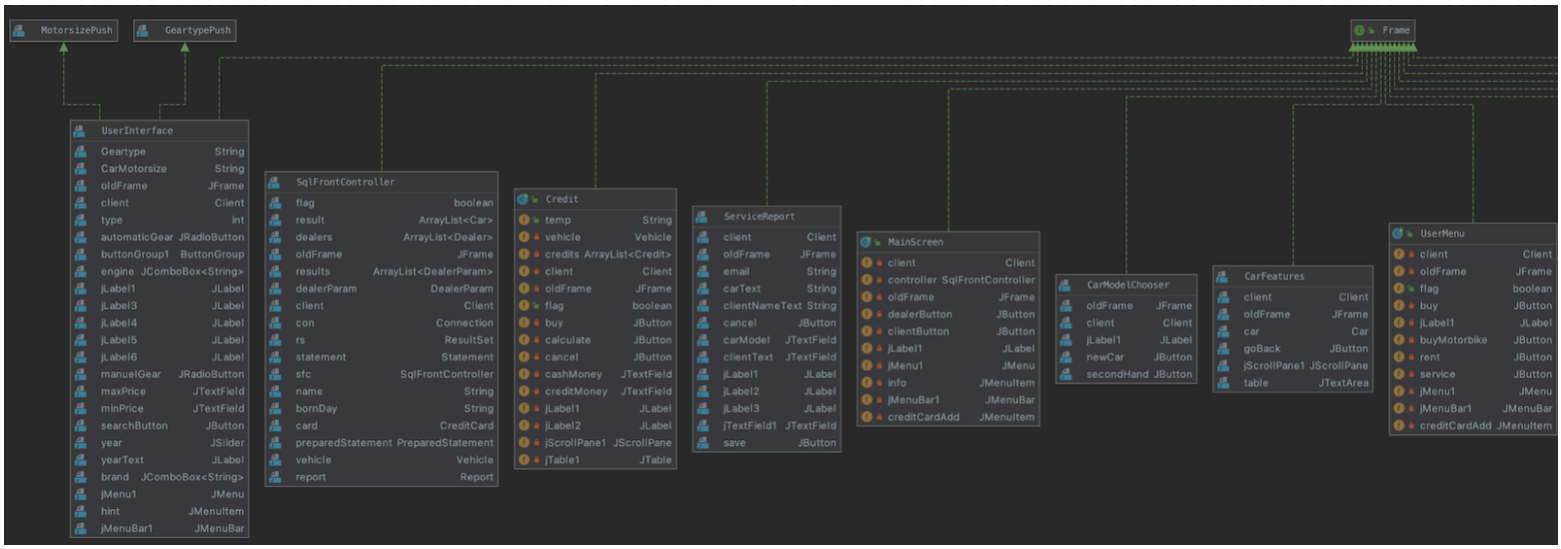
In Abstract Factory pattern an interface is responsible for creating a factory of related objects without explicitly specifying their classes. Each generated factory can give the objects as per the Factory pattern.Gear and Motorsize interfaces and concrete classes implementing these interfaces are creaetd. An abstract factory class AbstractFeature as next step are created. Factory classes ShapeGear and ShapeMotorSize are defined where each factory extends AbstractFeature. A factory creator class TotalChoicer is created. AbstrucktFeaturePattern, demo class uses TotalChoicer to get a AbstractFeature object. It will pass information (Gear for ShapeGear) to AbstractFeature to get the type of object it needs. It also passes information (Motorsize for ShapeMotorsize) to AbstractFeature to get the type of object it needs.

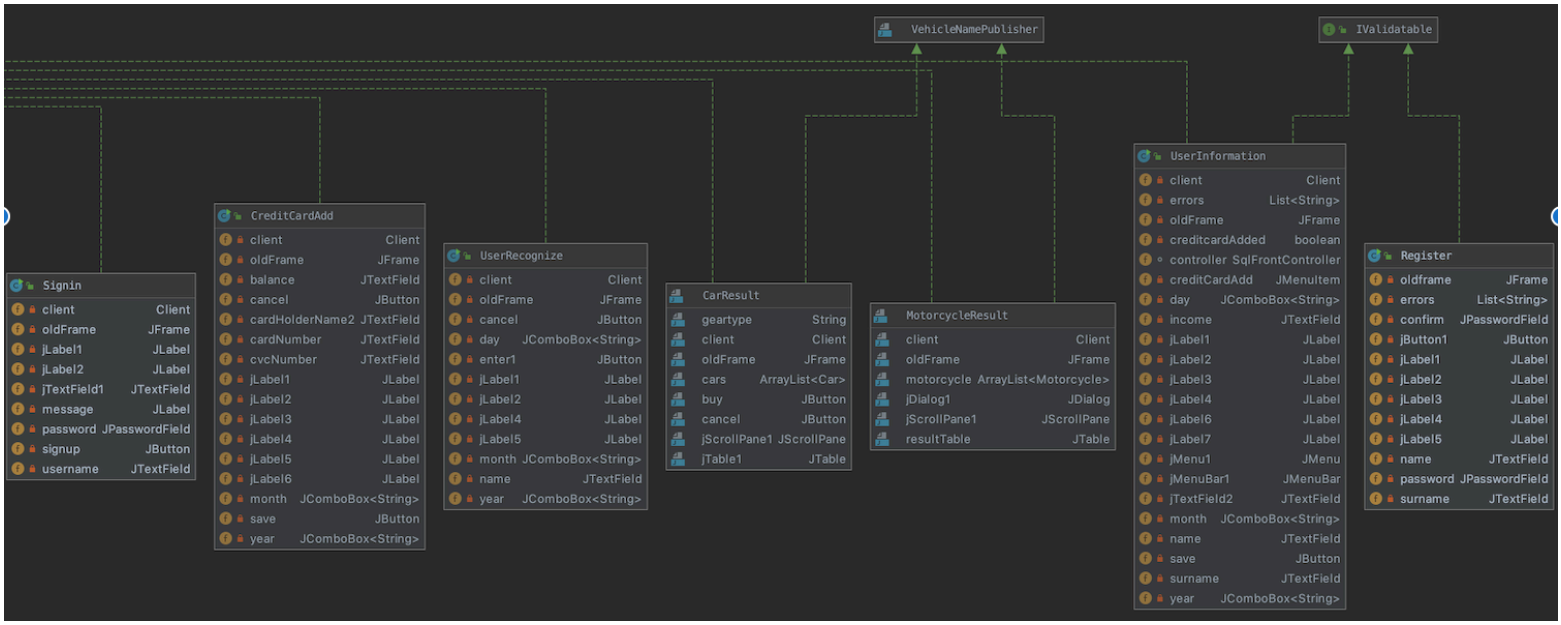


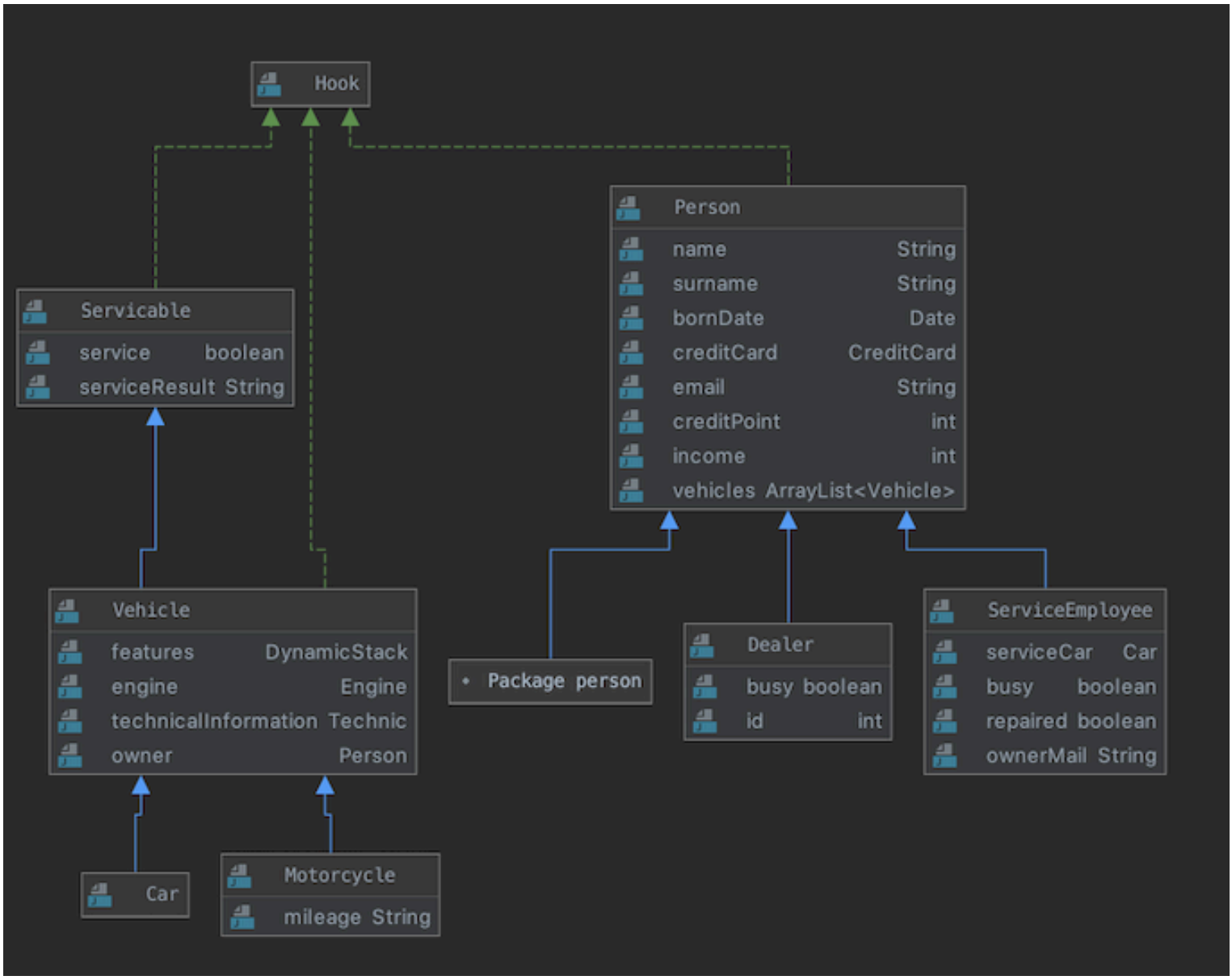
7.Class Diagram

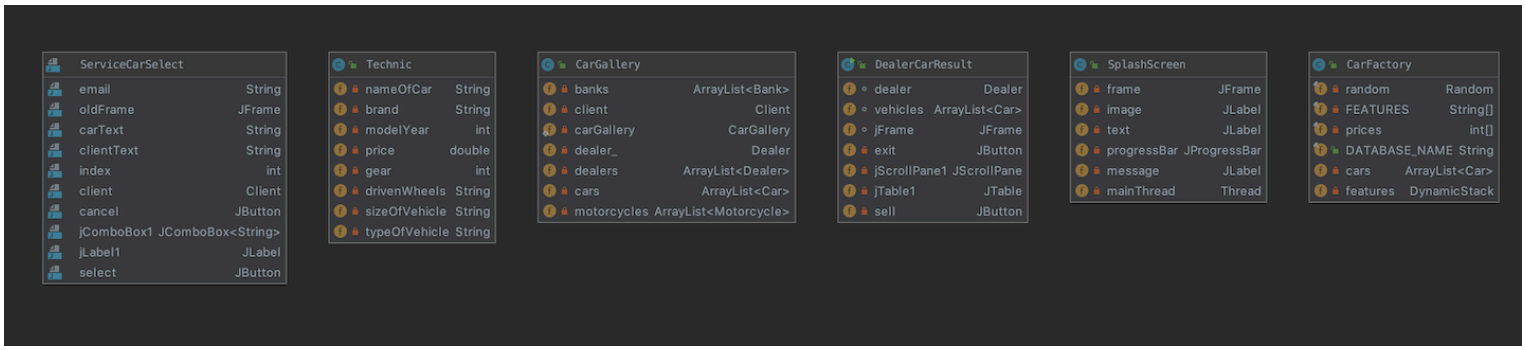
7.1 Computer Version

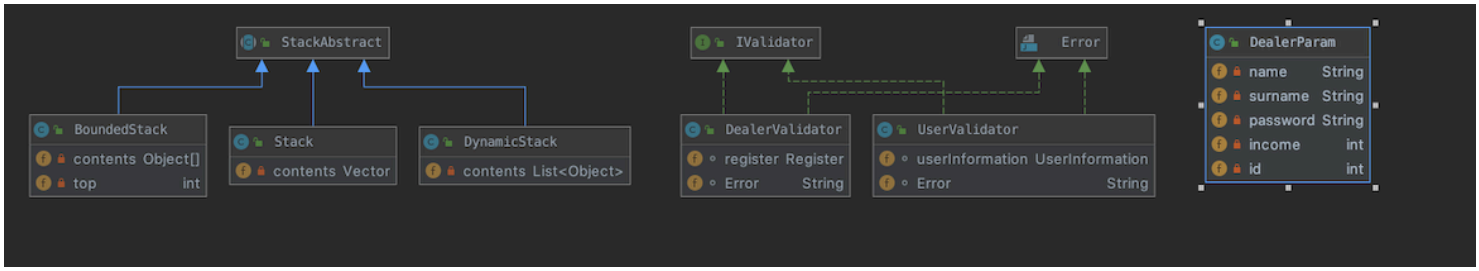
(The whole version of the diagram is loaded into github in one piece)

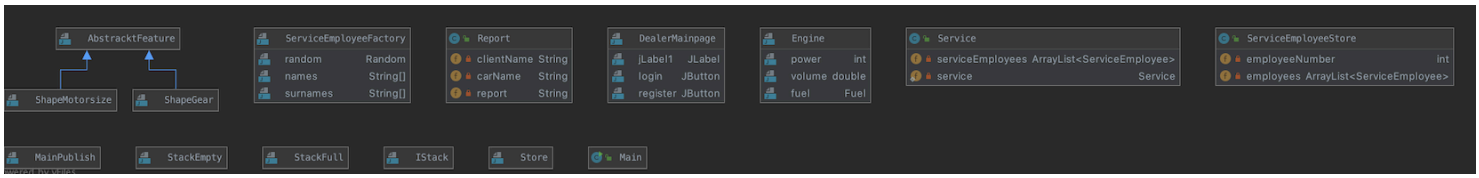





Hook
Person
name String
surname String
bornDate Date
creditCard CreditCard
email String
creditPoint int
income int
vehicles ArrayList<Vehicle>
Servicable
service boolean
serviceResult String
Vehicle
features DynamicStack
engine Engine
technicalInformation Technic
owner Person
Package person
Dealer
busy boolean
id int
ServiceEmployee
serviceCar Car
busy boolean
repaired boolean
ownerMail String
Car
Motorcycle
mileage String

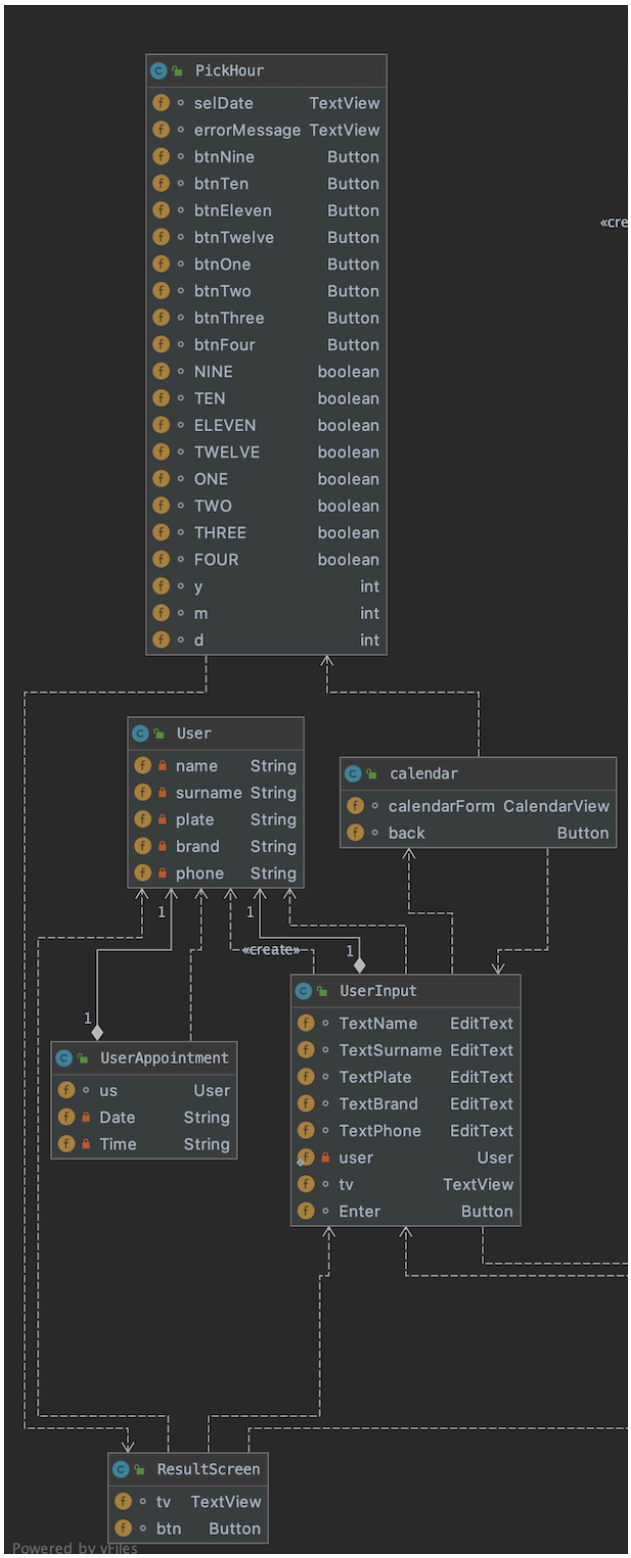

ServiceCarSelect
email String
oldFrame JFrame
carText String
clientText String
index int
client Client
cancel JButton
jComboBox1 JComboBox<String>
jLabel1 JLabel
select JButton
Technic
nameOfCar String
brand String
modelYear int
price double
gear int
drivenWheels String
sizeOfVehicle String
typeOfVehicle String
CarGallery
banks ArrayList<Bank>
client Client
carGallery CarGallery
dealer_ Dealer
dealers ArrayList<Dealer>
cars ArrayList<Car>
motorcycles ArrayList<Motorcycle>
DealerCarResult
dealer Dealer
vehicles ArrayList<Car>
jFrame JFrame
exit JButton
jScrollPane1 JScrollPane
jTable1 JTable
sell JButton
SplashScreen
frame JFrame
image JLabel
text JLabel
progressBar JProgressBar
message JLabel
mainThread Thread
CarFactory
random Random
FEATURES String[]
prices int[]
DATABASE_NAME String
cars ArrayList<Car>
features DynamicStack

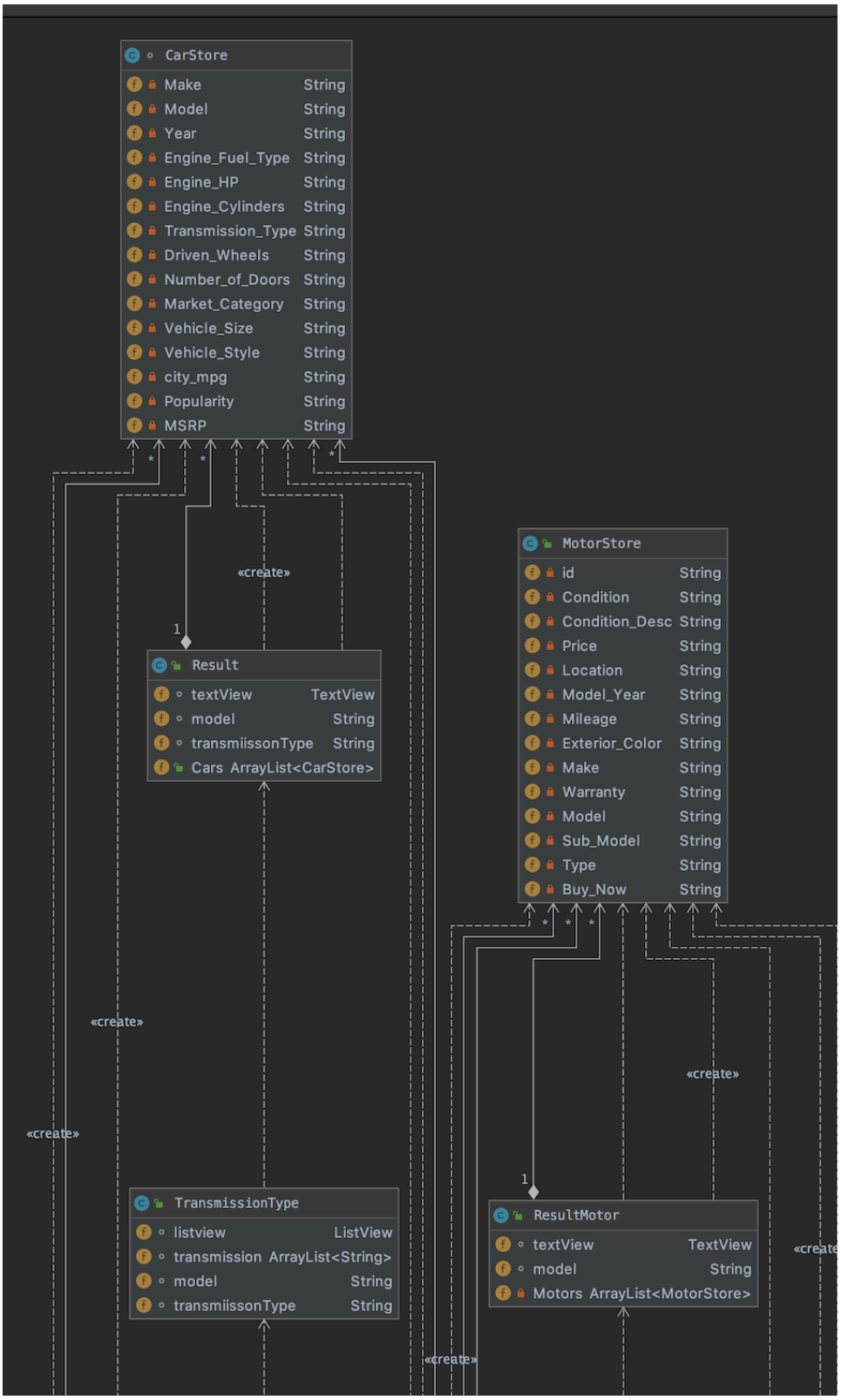

StackAbstract
BoundedStack
contents Object[]
top int
Stack
contents Vector
DynamicStack
contents List<Object>
IValidator
Error
DealerValidator
register Register
Error String
UserValidator
userInformation UserInformation
Error String
DealerParam
name String
surname String
password String
income int
id int

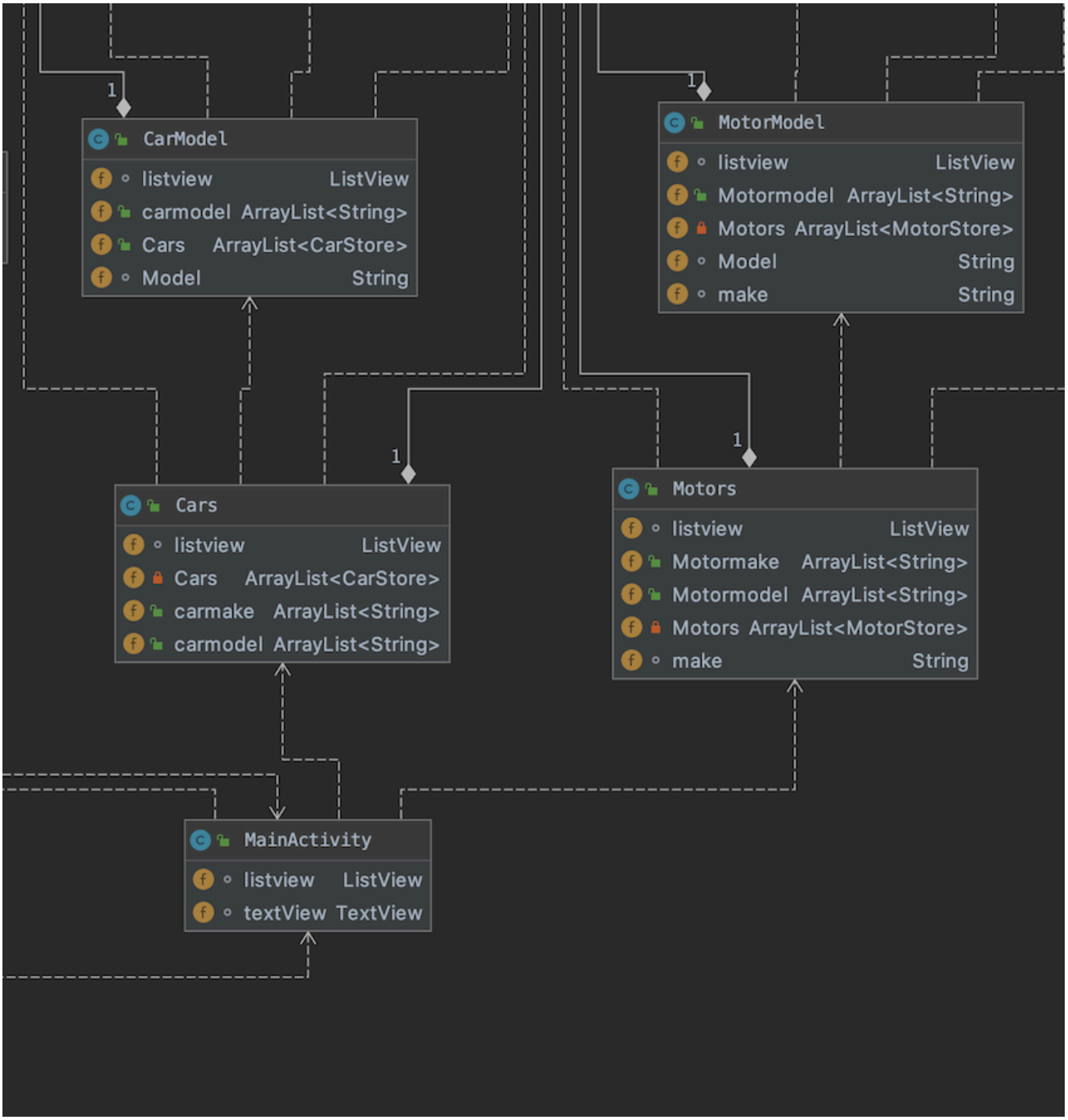

AbstracktFeature
ShapeMotorsize
ShapeGear
ServiceEmployeeFactory
random Random
names String[]
surnames String[]
Report
clientName String
carName String
report String
DealerMainpage
jLabel1 JLabel
login JButton
register JButton
Engine
power int
volume double
fuel Fuel
Service
serviceEmployees ArrayList<ServiceEmployee>
service Service
ServiceEmployeeStore
employeeNumber int
employees ArrayList<ServiceEmployee>
MainPublish
StackEmpty
StackFull
IStack
Store
Main

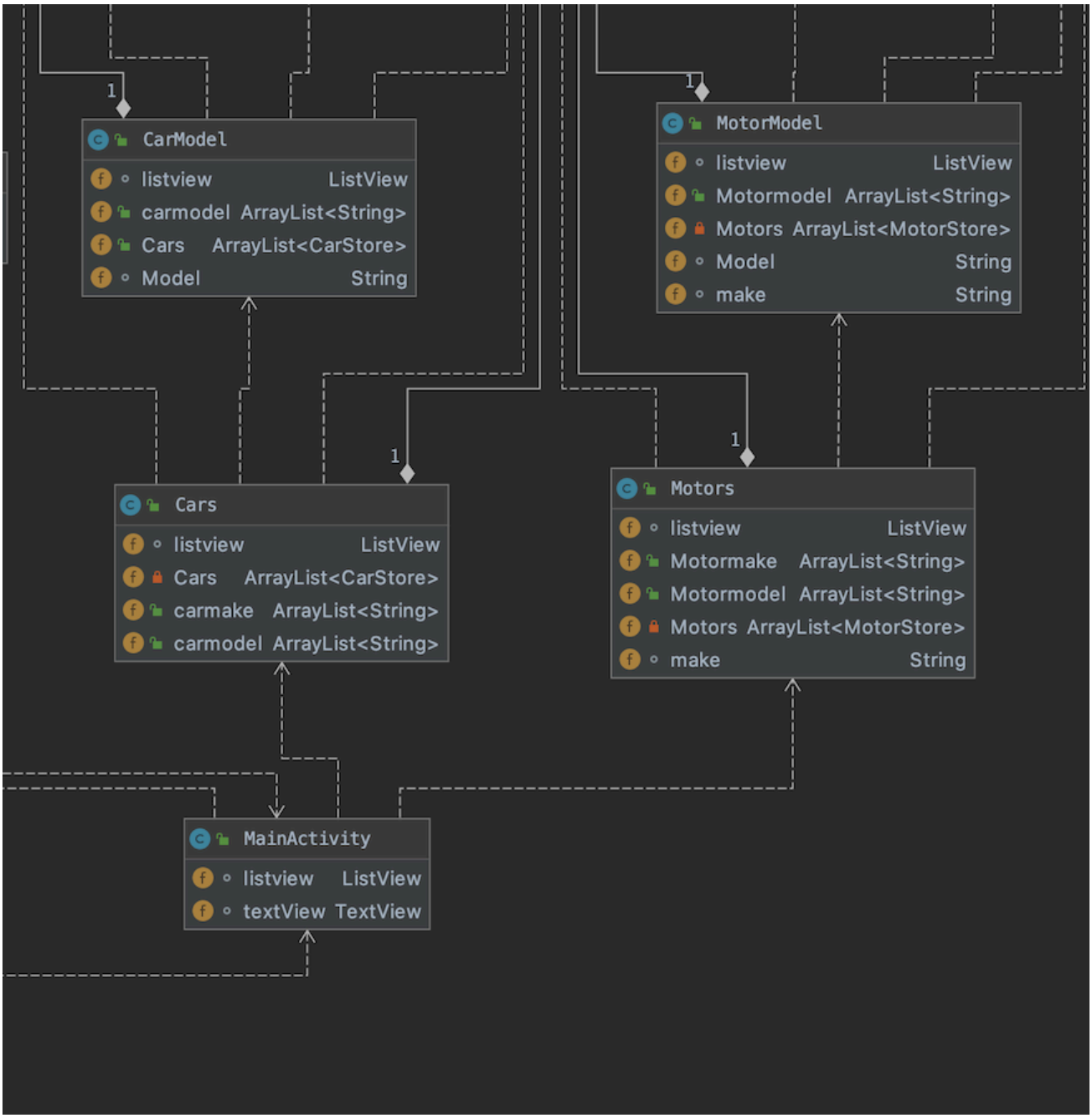

ServiceEmployeeStore
employeeNumber int
employees ArrayList<ServiceEmployee>
MotorcycleFactory
DATABASE_NAME String
motorcycles ArrayList<Motorcycle>
Fuel
highwayMPG double
cityMPG double
Feature
name String
price int
SingleDelete
deleteMessage SingleDelete
BankFactory
bankNames String[]
AbstrucktFeaturePattern
getVehicleType
TotalChoicer

7.2 Android Version




CarStore
Make String
Model String
Year String
Engine_Fuel_Type String
Engine_HP String
Engine_Cylinders String
Transmission_Type String
Driven_Wheels String
Number_of_Doors String
Market_Category String
Vehicle_Size String
Vehicle_Style String
city_mpg String
Popularity String
MSRP String
«create»
Result
textView TextView
model String
transmiissonType String
Cars ArrayList<CarStore>
MotorStore
id String
Condition String
Condition_Desc String
Price String
Location String
Model_Year String
Mileage String
Exterior_Color String
Make String
Warranty String
Model String
Sub_Model String
Type String
Buy_Now String
«create»
«create»
«create»
TransmissionType
listview ListView
transmission ArrayList<String>
model String
transmiissonType String
ResultMotor
textView TextView
model String
Motors ArrayList<MotorStore>
«create»
«create»


1
CarModel
listview ListView
carmodel ArrayList<String>
Cars ArrayList<CarStore>
Model String
1
MotorModel
listview ListView
Motormodel ArrayList<String>
Motors ArrayList<MotorStore>
Model String
make String
1
1
Cars
listview ListView
Cars ArrayList<CarStore>
carmake ArrayList<String>
carmodel ArrayList<String>
Motors
listview ListView
Motormake ArrayList<String>
Motormodel ArrayList<String>
Motors ArrayList<MotorStore>
make String
MainActivity
listview ListView
textView TextView


CarModel
listview ListView
carmodel ArrayList<String>
Cars ArrayList<CarStore>
Model String
MotorModel
listview ListView
Motormodel ArrayList<String>
Motors ArrayList<MotorStore>
Model String
make String
Cars
listview ListView
Cars ArrayList<CarStore>
carmake ArrayList<String>
carmodel ArrayList<String>
Motors
listview ListView
Motormake ArrayList<String>
Motormodel ArrayList<String>
Motors ArrayList<MotorStore>
make String
MainActivity
listview ListView
textView TextView
1
1
1
1