

Car management app

Documentation

Ciprian Cuibus

2021

Contents

1	General presentation of the application	1
2	Used technologies	2
2.1	Technologies	2
3	Implementation & Features	3
3.1	Implementation	3
3.2	Features	4
3.2.1	Log in page	4
3.2.2	Register page	4
3.2.3	Main page	5
3.2.4	View my cars page	5
3.2.5	Add a new car page	6
4	Diagrams	7
4.1	Use case diagram	7
4.1.1	User account creation	7
4.2	Sequence diagram	8
4.2.1	Register a new user	8
4.2.2	Add a new car	8
4.2.3	Delete a car	9
4.3	Class diagram	9
5	Feature improvements	10
5.1	Improvements	10
6	Conclusion	11

Chapter 1

General presentation of the application

The application represents a small place where car enthusiasts can post specs about their car, such as a BMW 3 series which was tuned to have 1000HP and can be used only on the drag strip. The users can visualize the cars from other users to see their cars, but also visualize their own cars in a separate tab. Here they can add a car, edit an existing one or delete it.

Adding a car is comprised by adding some mandatory fields such as the brand, model, the year in which it was manufactured, the engine capacity, engine type e.g. V8 or V6, the weight of the car, horse power, torque and finally but not least the spec-ed type of the car i.e. for Circuit, Road, Drag and so on.

The UI is modern and straightforward with no unnecessary ads or useless content just the cars of some passionate people.

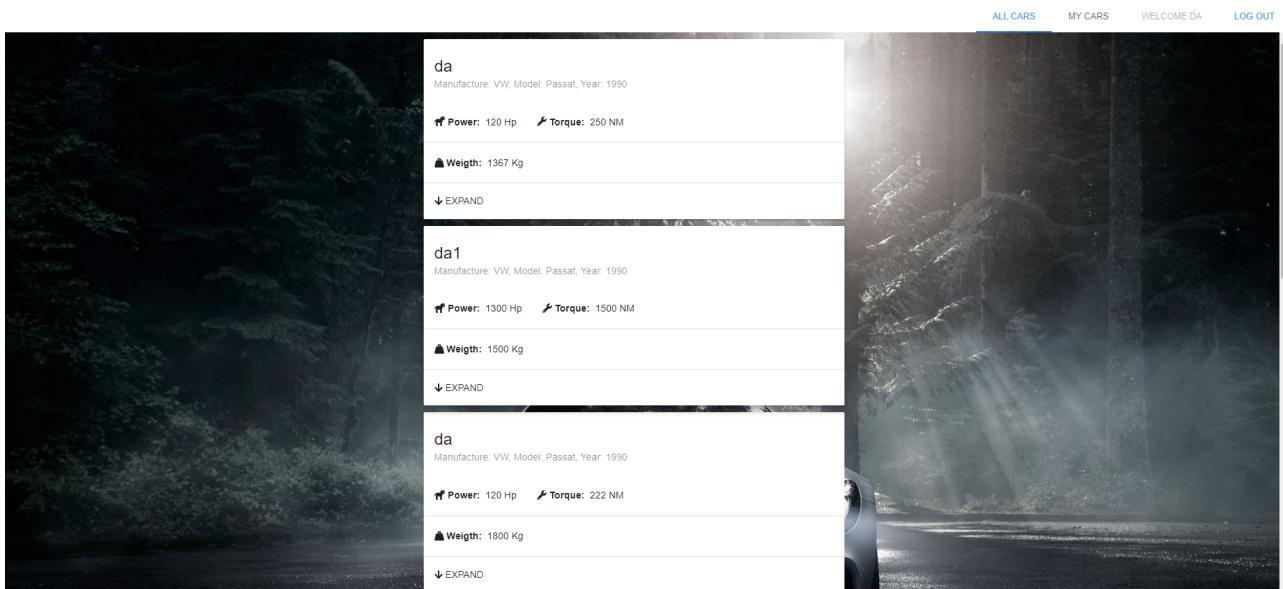


Figure 1.1: The main page

Chapter 2

Used technologies

2.1 Technologies

1. Back-end
 - 1.1. Node.js
 - 1.2. nginx
 - 1.3. Docker
 - 1.4. MongoDB
 - 1.5. jwt
 - 1.6. express
2. Front-end
 - 2.1. Vue.js
 - 2.2. Vue router
 - 2.3. Vue material
 - 2.4. axios

Chapter 3

Implementation & Features

3.1 Implementation

The back-end is made in node.js and used docker to create containers for the existing micro-services and nginx to create a reverse proxy to have only one access point to the micro-services, this runs on port 80 and the other micro-services, car micro-service and user micro-service which runs on port 3000 and respectively on port 3001. The role of this reverse proxy is to forward the request to the correct micro-service without knowing on which port it might run. The micro-services follow the clean architecture proposed by Bob Martin's Clean Architecture model proposed by Bob Martin. Where in the case of this implementation the entities are represented by the car and user. The use cases are simple crud operations and the controllers are gateways between the use cases and the api exposed using express.

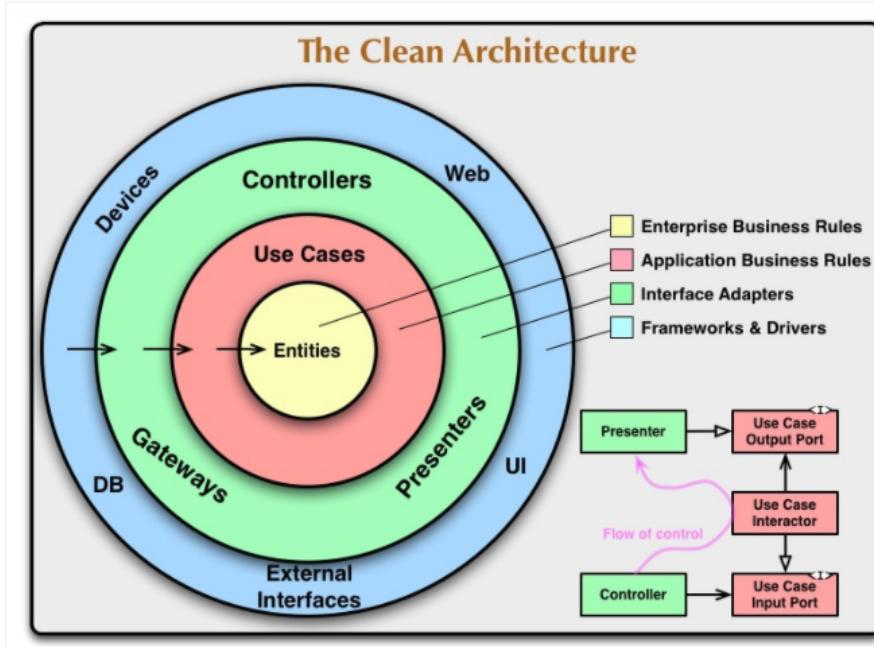


Figure 3.1: Clean architecture

The front-end was made in Vue.js a powerful javascript framework which let you create beautiful applications. For the look of the application Vue material was used and for api calls

to the back-end axios as utilised.

3.2 Features

The application let's the user visualize the cars of all the users and their own cars, update them, delete them and create a new car. This can be via a simple and clean UI.

3.2.1 Log in page

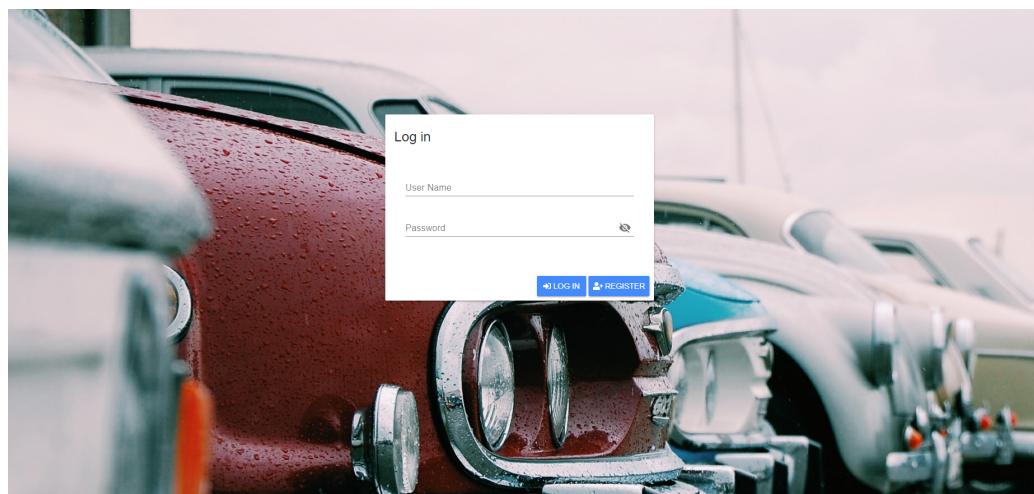


Figure 3.2: Log In page

3.2.2 Register page

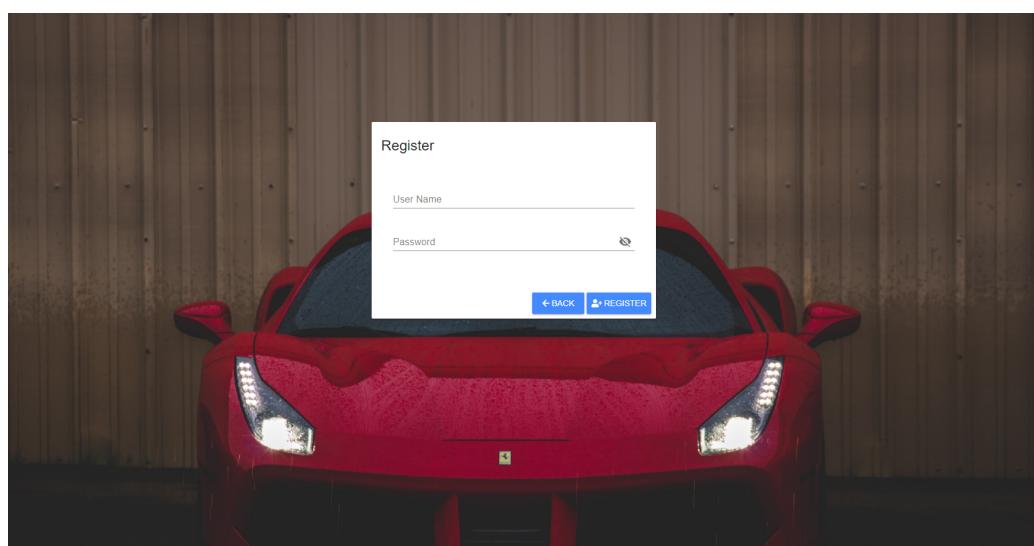


Figure 3.3: Register page

3.2.3 Main page

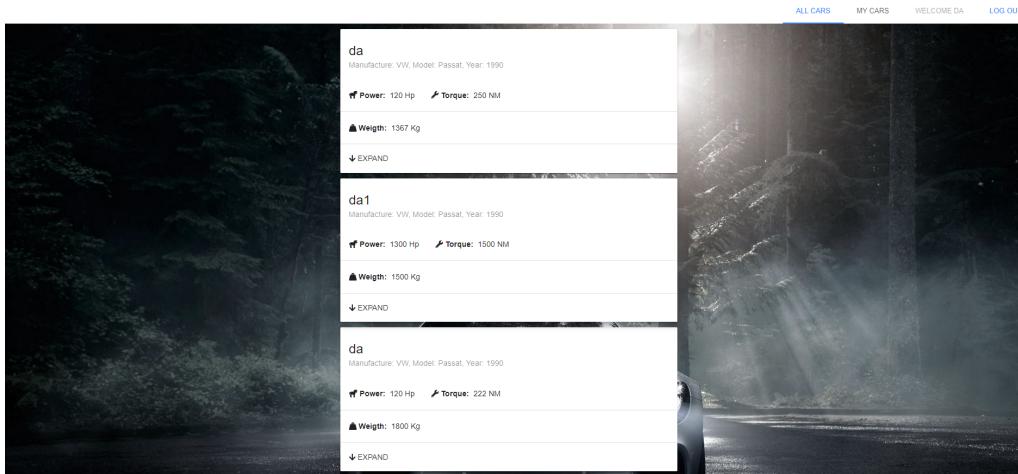


Figure 3.4: The main page

3.2.4 View my cars page

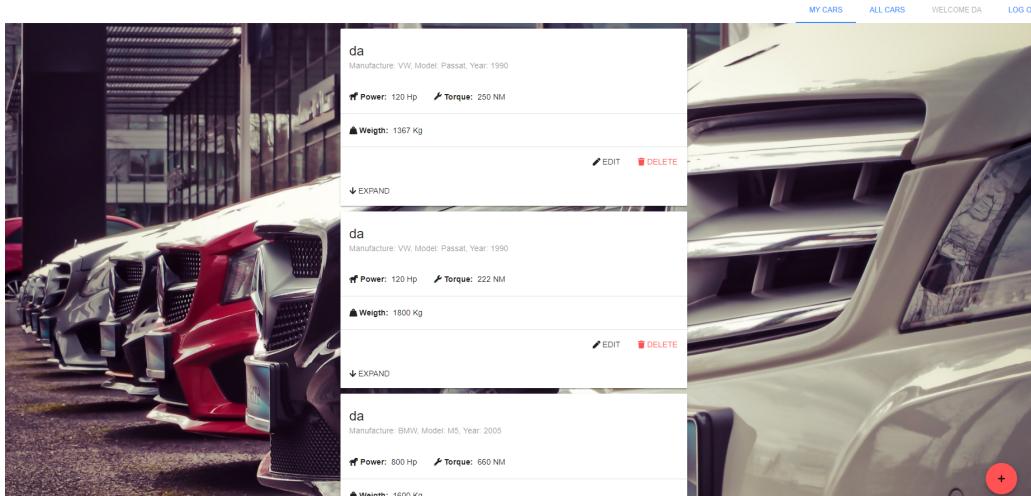


Figure 3.5: View my cars page

3.2.5 Add a new car page

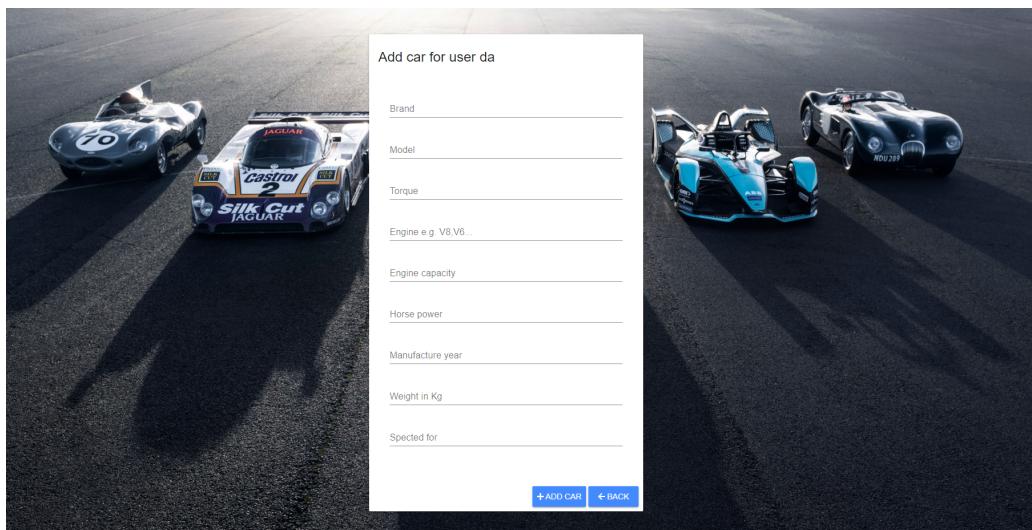
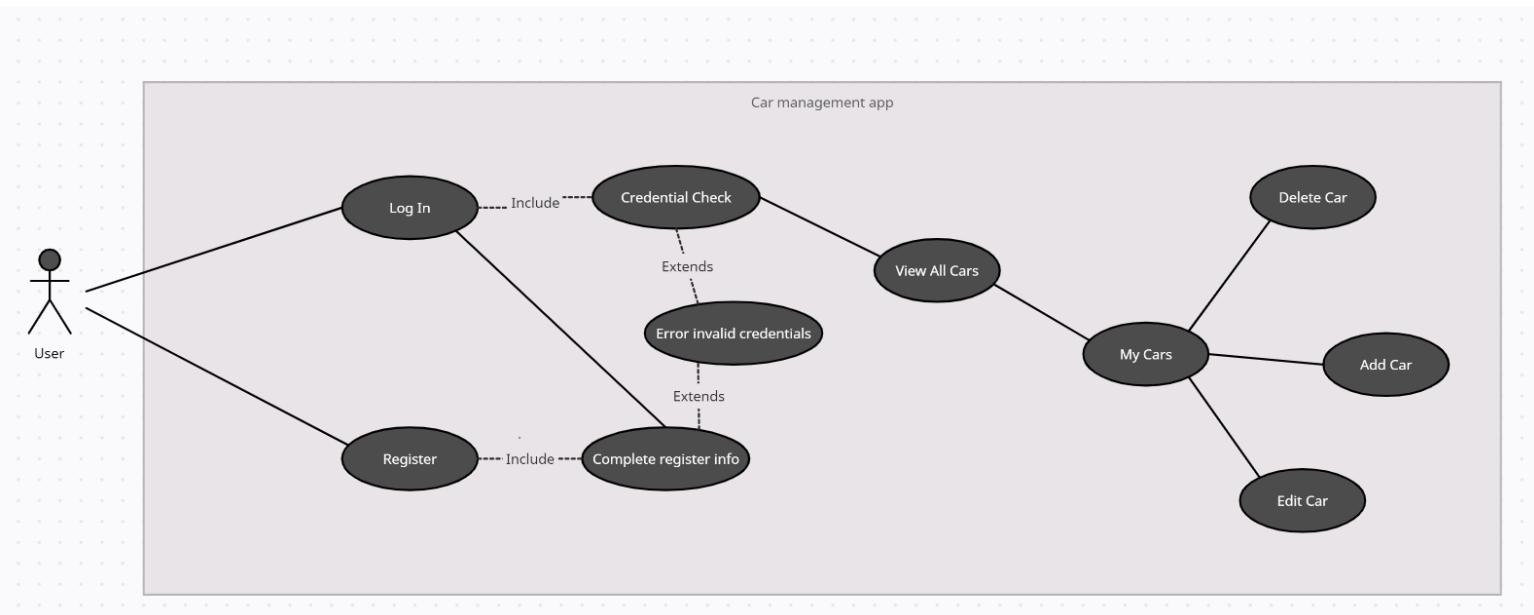


Figure 3.6: Add new car page

Chapter 4

Diagrams

4.1 Use case diagram

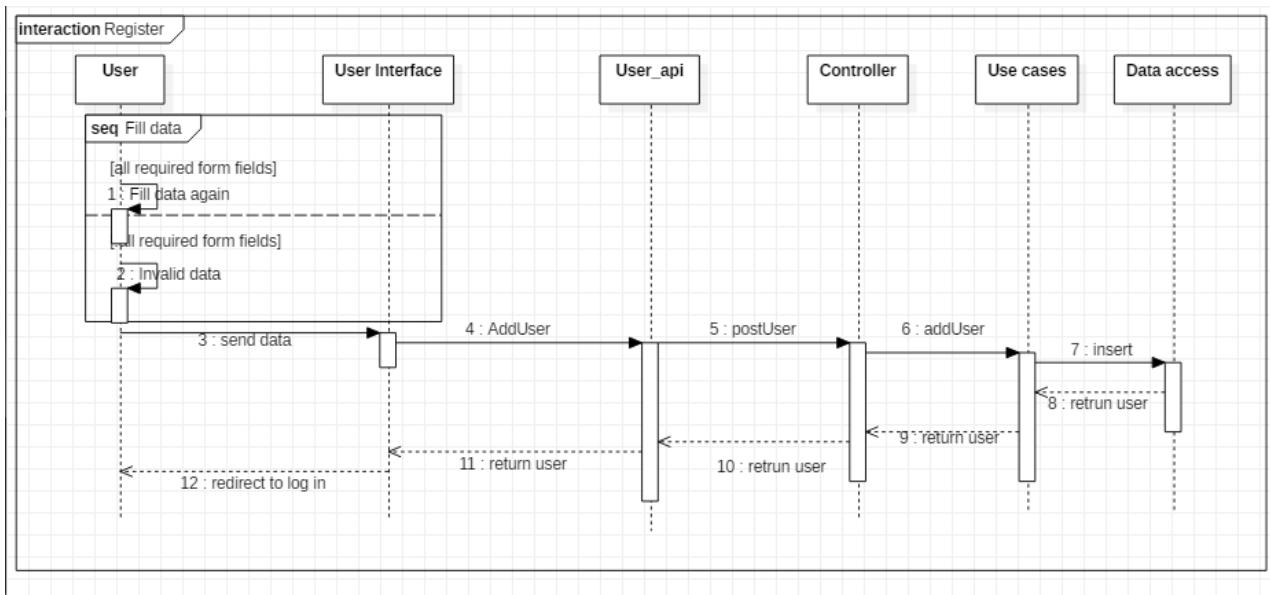


4.1.1 User account creation

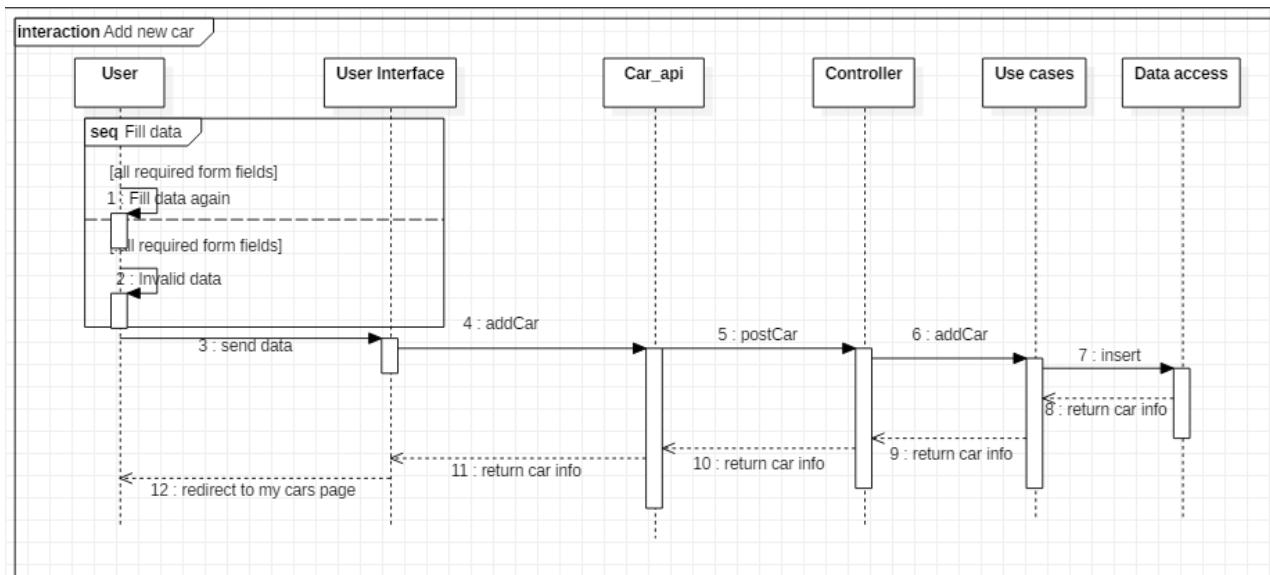
1. Gogu, a non-user, opens the website
2. Log in screen is displayed
3. Gogu, clicks on register
4. Gogu, inserts his data and clicks on register
5. Gogu is redirected to login
6. Gogu enters his account credentials
7. Gogu is redirected to main page

4.2 Sequence diagram

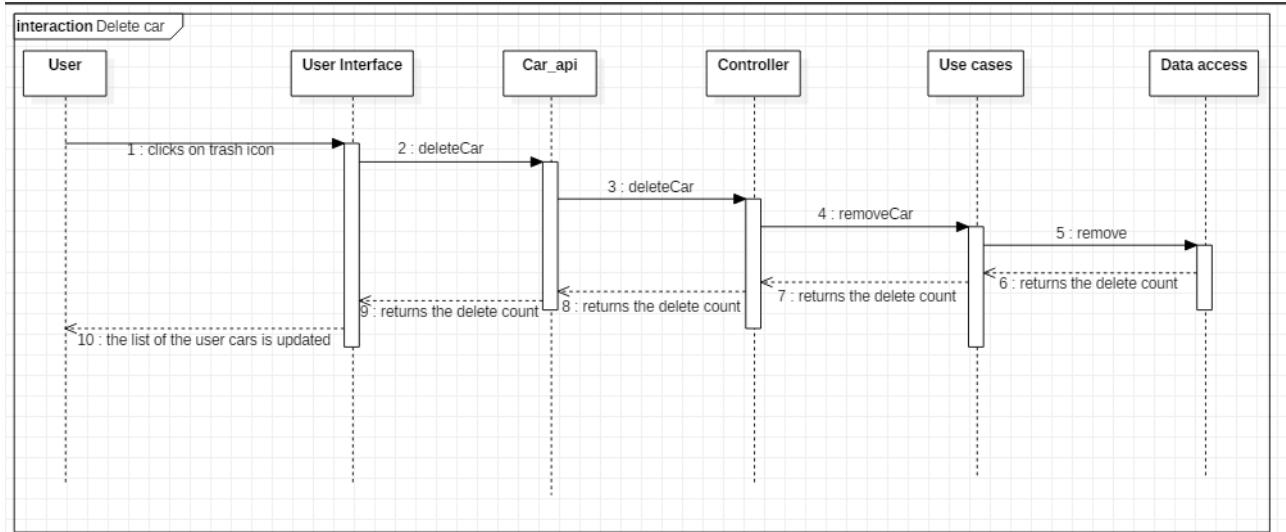
4.2.1 Register a new user



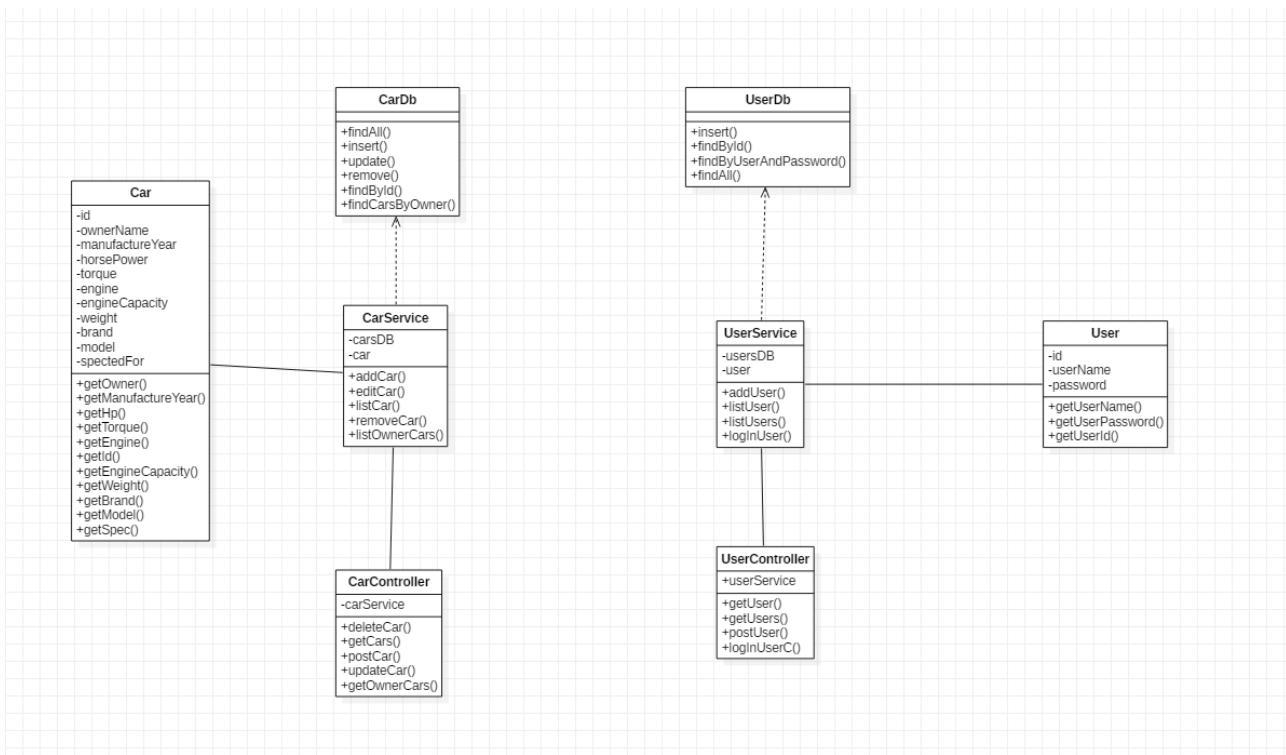
4.2.2 Add a new car



4.2.3 Delete a car



4.3 Class diagram



Chapter 5

Feature improvements

5.1 Improvements

In the feature this application could receive a search bar to search after a specific query, a filter framework to filter the data as the user wishes. Also a comments system could a welcome feature. Also the ability to like and save others post could make this application a truly wonderful place for car enthusiasts.

Chapter 6

Conclusion

In conclusion this application targets the car enthusiasts more specifically the so called petrol heads which love the cars which have an internal combustion engine. The app offers a smooth UI with simple CRUD operations.