Garage app with microservices

Documentation

Cota Ionas-Calin

2021

Contents

Chapter 1

General presentation of the application

It is a small CRUD app that lets you store the cars you own or want to buy. The user can view, add, remove and update existing cars through a minimalistic pre 2000s web user interface. It's sole purpose was for learning about microservices and how to use them in an application + using external apis (login).

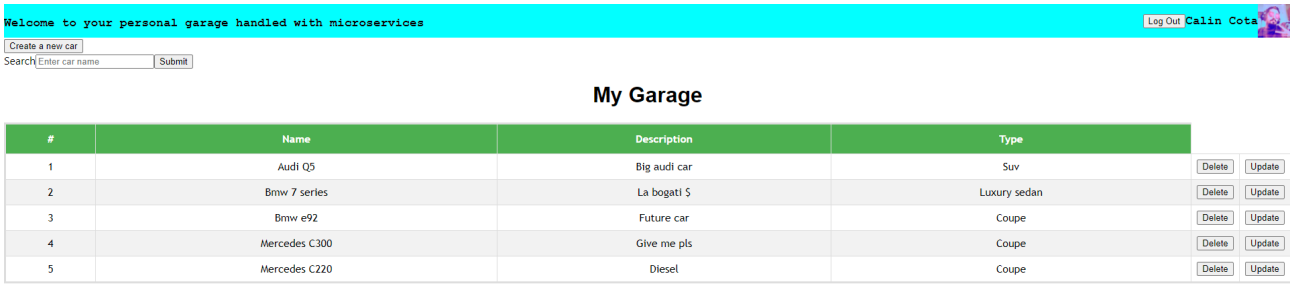


Figure 1.1: The main page

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. Mongoose
    1.6. express
2. Front-end
    2.1. React

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 8080 and the other micro-services, car micro-service and user micro-service which runs on ports 3000, 3001, 3002. 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 use cases are simple crud operations through apis exposed using express.

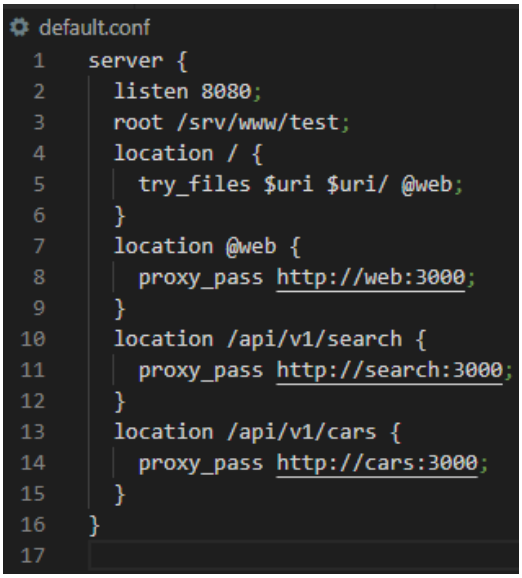


Figure 3.1: Config 1

The front-end was made in React which is a javascript library for building user interfaces. it makes it painless to create interactive Uis, too bad I don't know nor like making uis. It is component based, each state having its own state and are then composed to make complex Uis and for api calls to the back-end the Fetch API was used. The login was done using the Auth0 authentication and authorization service.

Figure 3.2: Docker compose

3.2 Features

The application let's the user view their own cars, update them, delete them and create a new car. This can be done via a very very simple ui.

3.2.1 Log in page

Pressing the login buton will take you to my auth0 domain. A successful login will take you back to the main page and your account data will be taken from google to show name+profile picture.

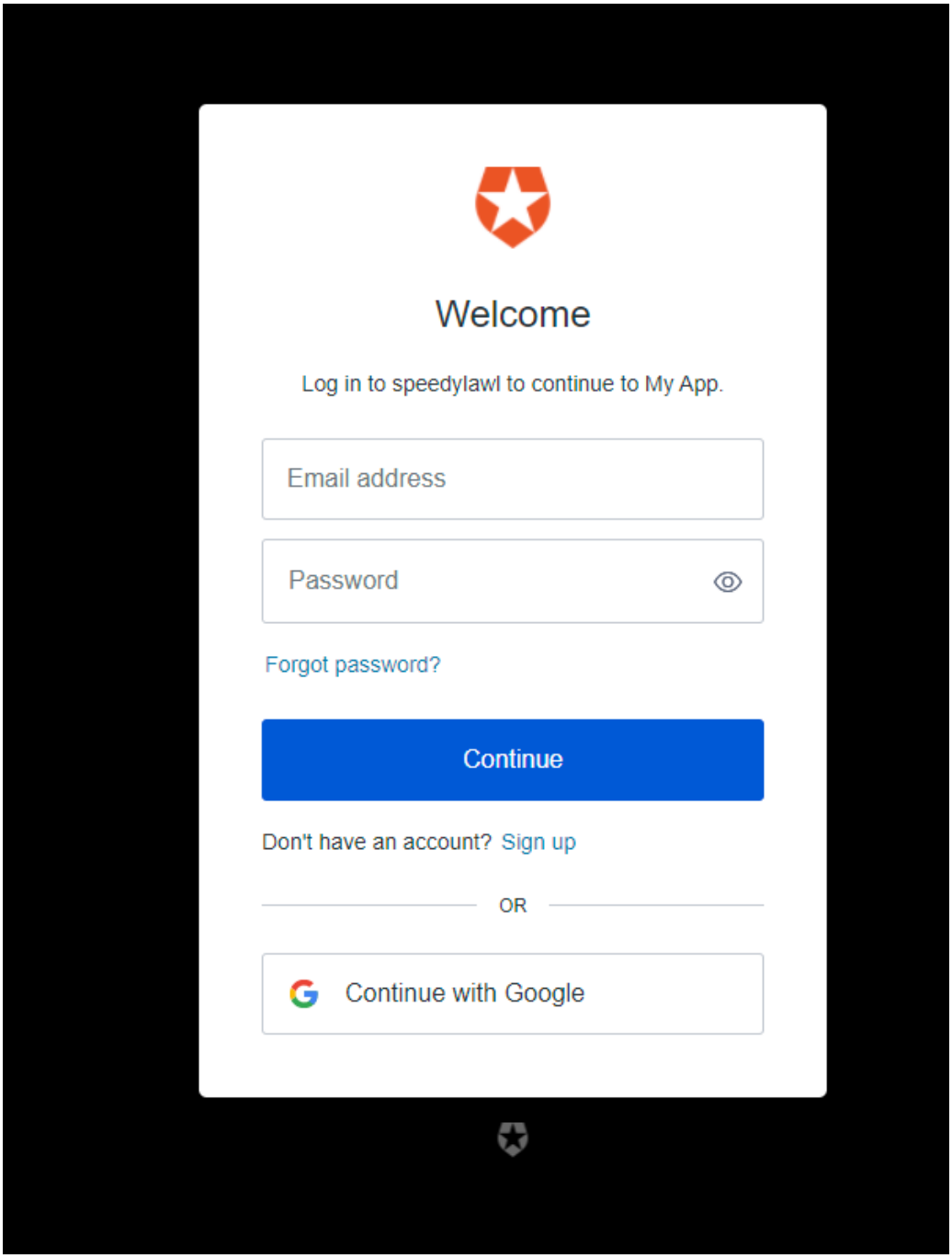
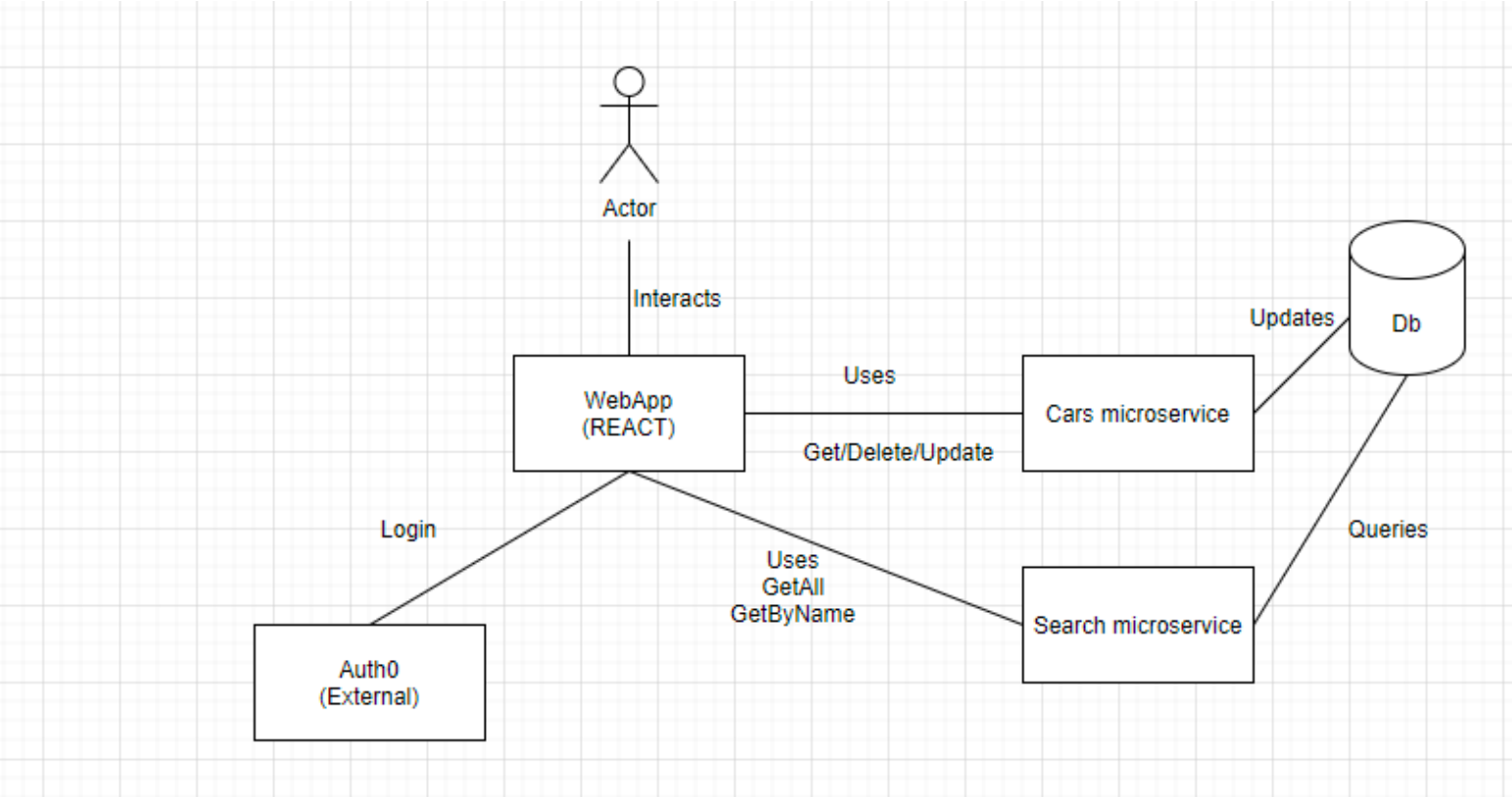


Figure 3.3: Log In page

Chapter 4

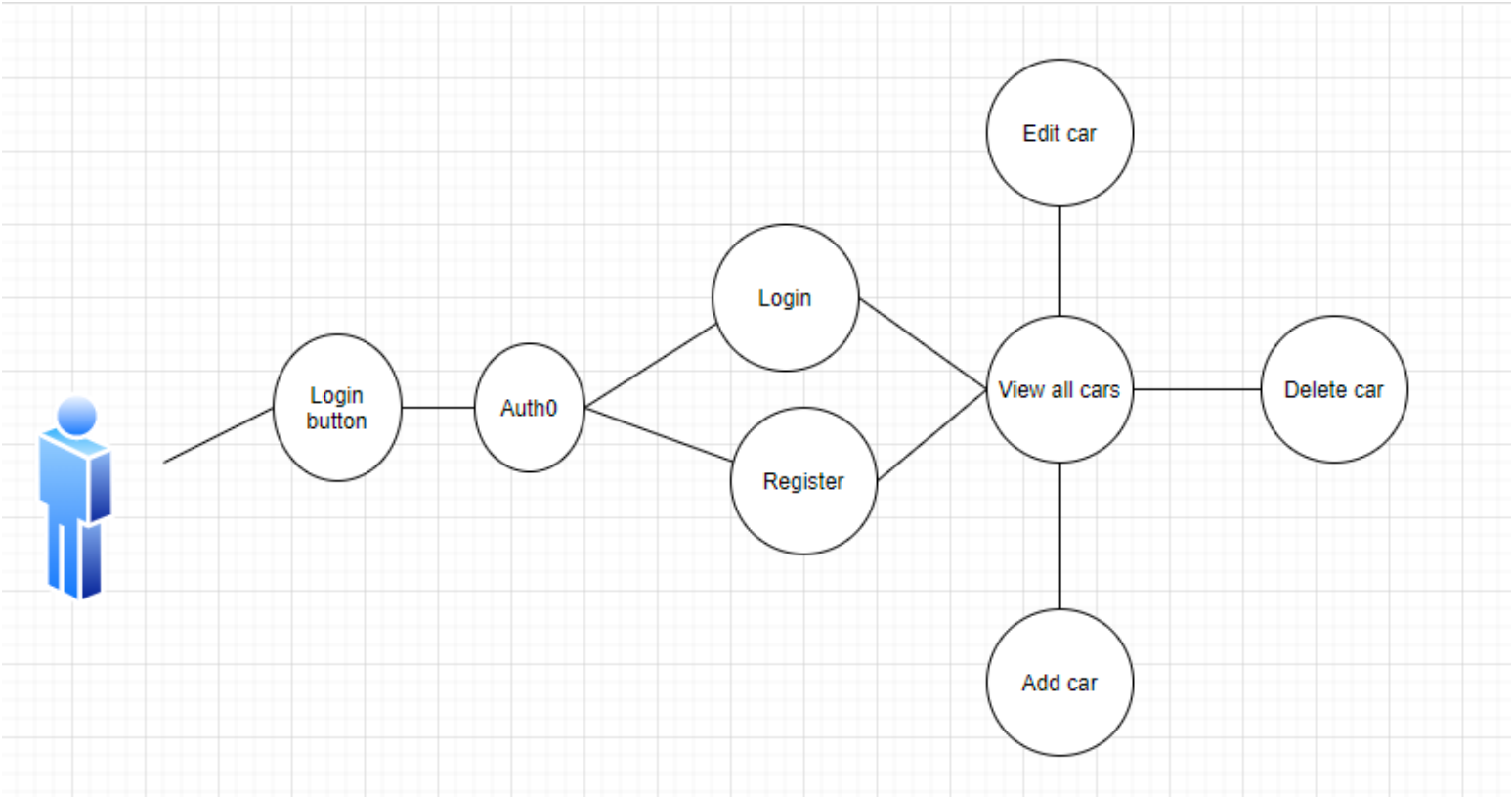
Diagrams



4.1 Use case diagram

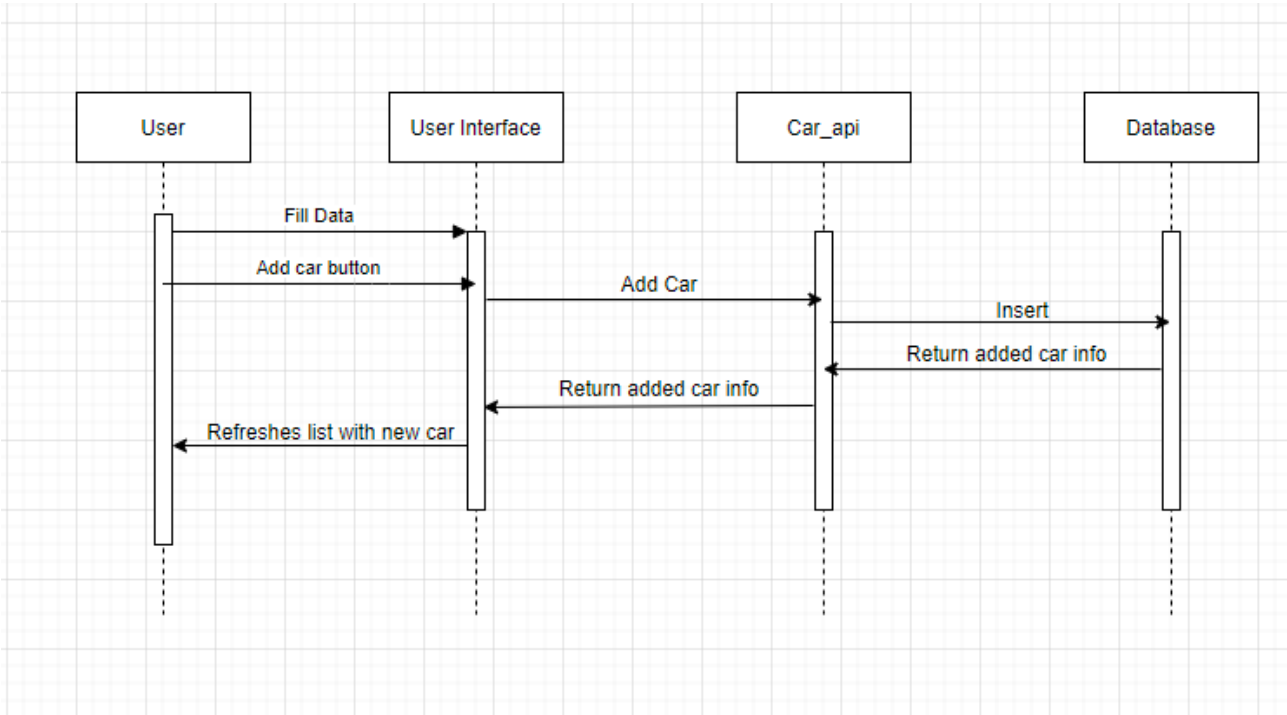
4.1.1 User account creation

1. User opens the main site
2. User presses login button
3. User is taken to Auth0 domain for login/register
4. User finishes login/register and is redirected to main page

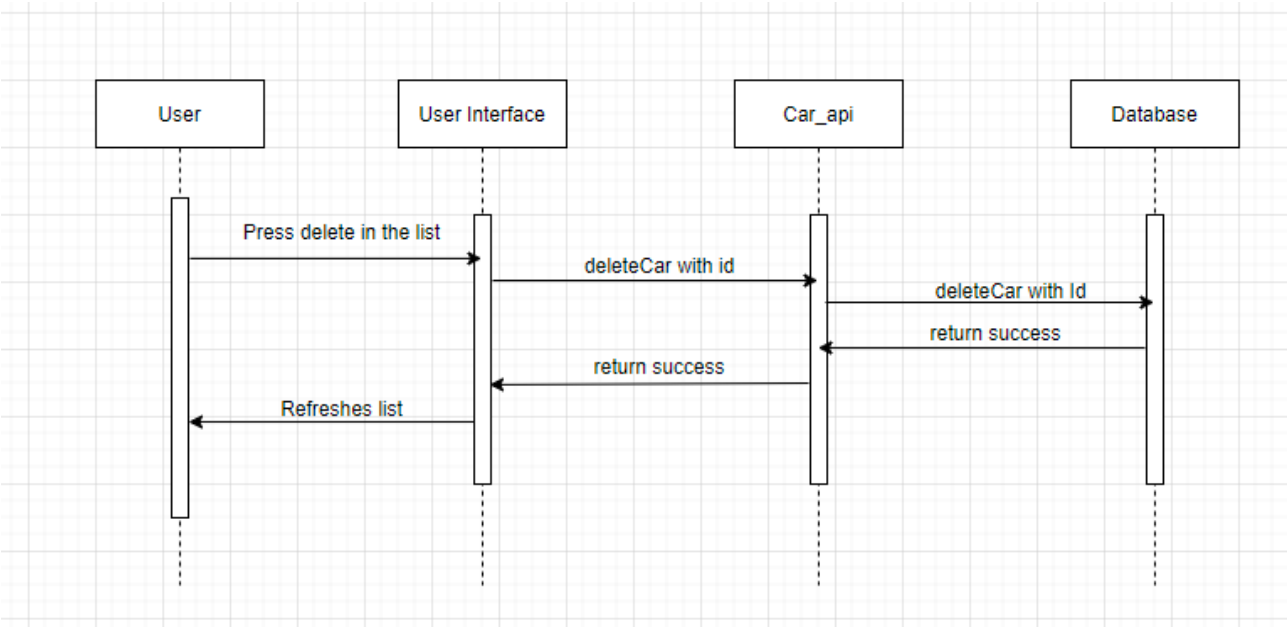


4.2 Sequence diagram

4.2.1 Add a new car



4.2.2 Delete a car



Chapter 5

Feature improvements

5.1 Improvements

In the feature this application could receive a proper user interface from someone who enjoys that, garages could be made public so people can share their wishlists, ownings etc. Social media integration could be added to link real life images of the cars from instagram/facebook.

Chapter 6

Conclusion

In conclusion this application taught me the basics of microservices while also providing a simple CRUD for cars