



**Kolegium Nauk Przyrodniczych
Uniwersytet Rzeszowski**

**Przedmiot:
Proseminarium**

**Dokumentacja projektu:
*Wypożyczalnia samochodów***

Wykonał:

**Zespół projektowy *Jarosław Urbaniak, Krzysztof Bohaczyk,
Karol Suchta***

Prowadzący: dr inż. Marcin Ochab

Rzeszów 2020

Cel aplikacji

Umożliwienie sprawnego oraz łatwego wypożyczenia samochodów dla klienta oraz łatwy system zarządzania salonami dla pracowników.

Działanie aplikacji

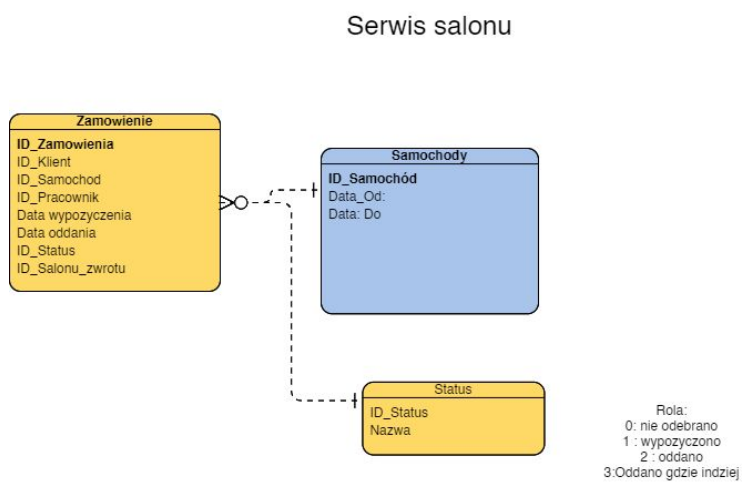
Aplikacja umożliwia wypożyczenie jednego, spośród wielu aut, znajdujących się w salonach firmy. Rezerwacja następuje poprzez określenie przez klienta okresu wypożyczenia (od-do), jeśli w tym terminie dostępne będą auta, petentowi zostanie wyświetlona lista dostępnych pojazdów. Dodatkowo klient wybiera, dogodną dla siebie lokalizację jednego spośród salonów firmy, w której zwróci samochód. Po kliknięciu przycisku "Reserve" przy danym samochodzie, zamówienie zostanie dodane do klienta.

Funkcjonalności

- Wyszukiwarka dostępnych samochodów w wybranym przez klienta terminie
- Możliwość sprawdzenia historii rezerwacji oraz podglądu obecnie wypożyczonych pojazdów
- Możliwość wyboru punktu zwrotu spośród listy salonów.
- Autentyfikacja - rejestracja, logowanie klientów

Diagram ERD

Visual Paradigm Online Express Edition



Serwis przenoszenia
auta

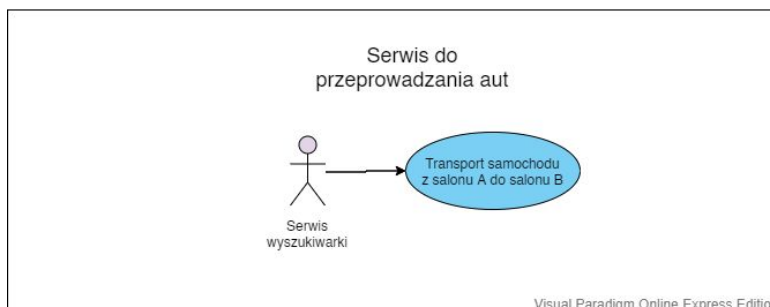
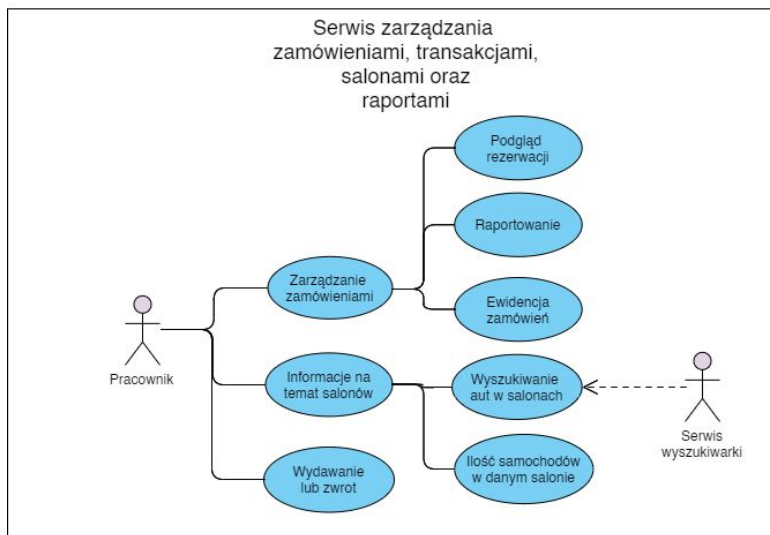
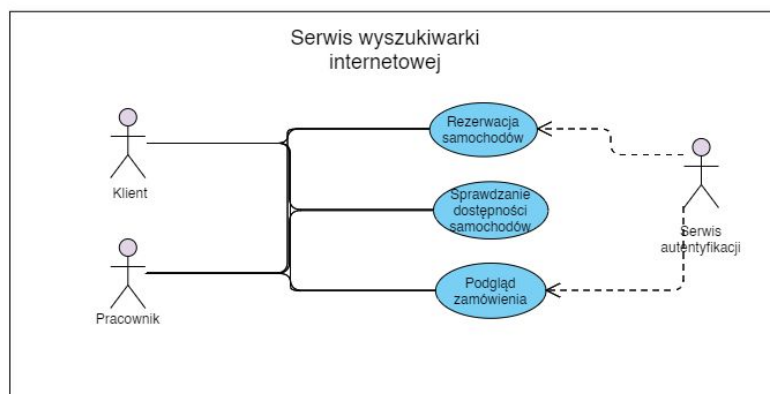
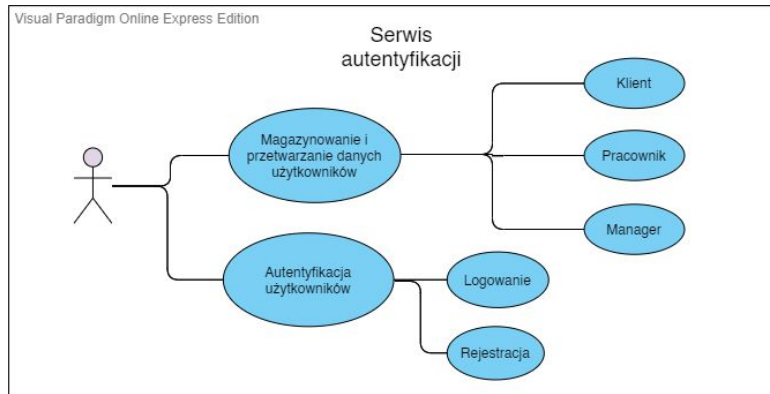
Autentyfikacja



Serwis
wyszukiwarki

Visual Paradigm Online Express Edition

Diagram UML



Wykorzystane technologie

Vue.js - frontendowy framework javascriptowy, na nim zbudowana jest cała warstwa widoczna dla użytkownika

Node.js z Express.js - backend, obsługa bazy danych oraz poleceń wbijanych przez przeglądarkę bądź Postmana (json z danymi do wprowadzenia), obecnie zastąpionych przez mechanizm frontendowy

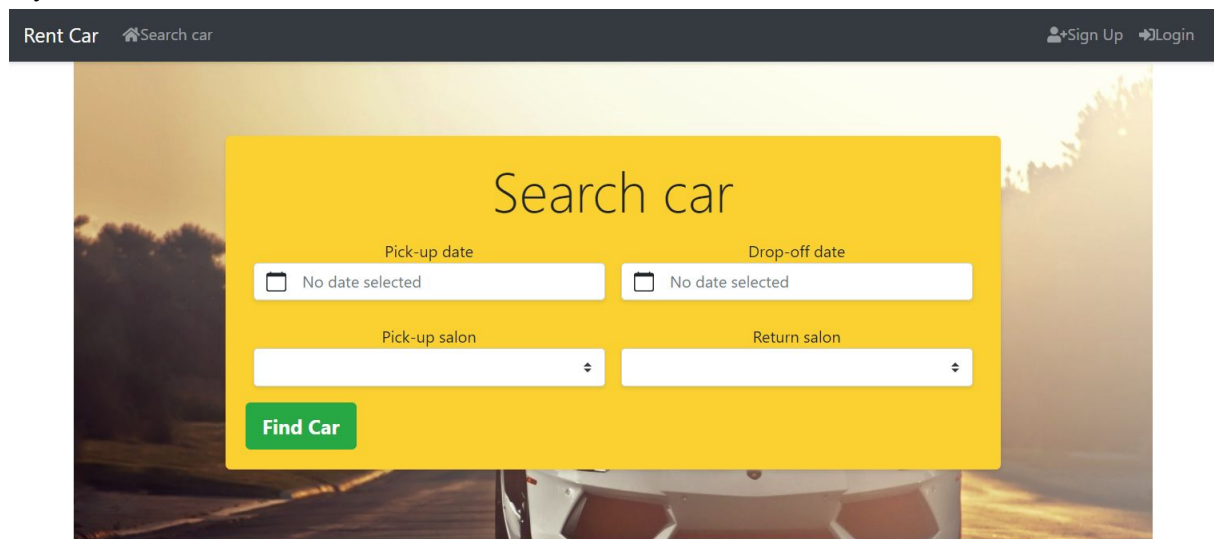
BootstrapVue - biblioteka CSS do zaawansowanego tworzenia interfejsu graficznego

MySQL - wolnodostępny, otwartoźródłowy system zarządzania relacyjnymi bazami danych

MongoDB - otwarty, nierelacyjny system zarządzania bazą danych napisany w języku C++.

Prezentacja aplikacji - grafika

Wyszukiwarka:



Rent Car [Search car](#) [Sign Up](#) [Login](#)


Search car

Pick-up date Drop-off date

Pick-up salon Return salon

[Find Car](#)

Logowanie:




Email

Password

Login

Rejestracja:



First name

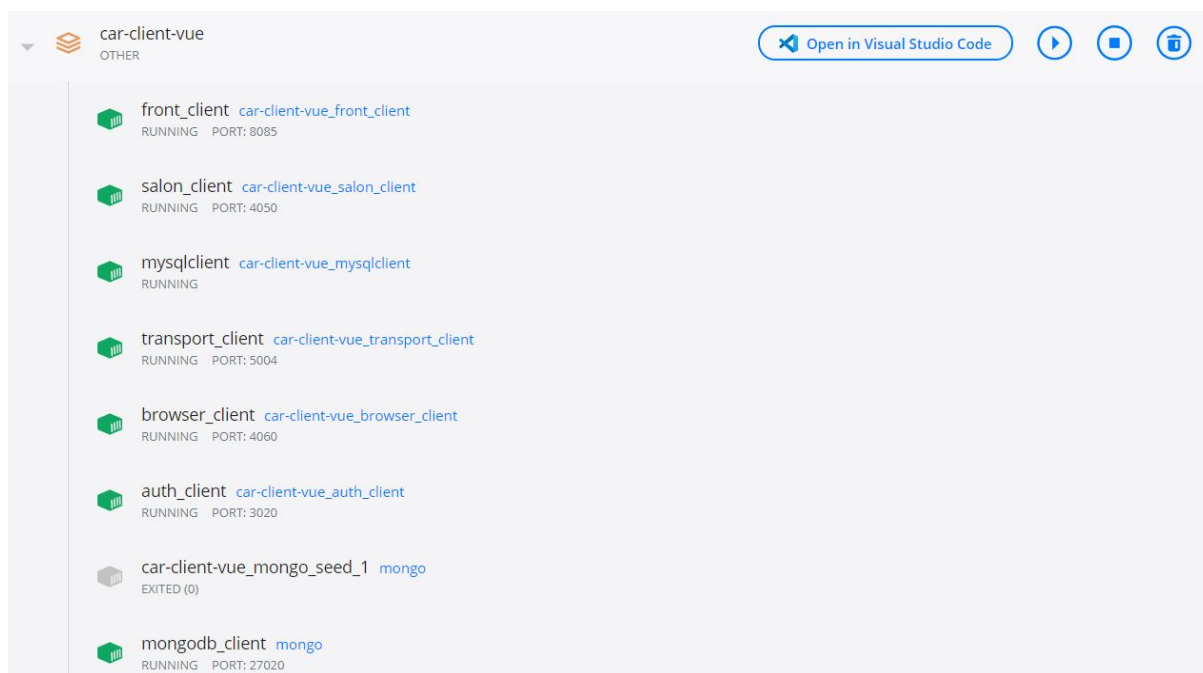
Last name

Email

Password

Address









Testowanie aplikacji










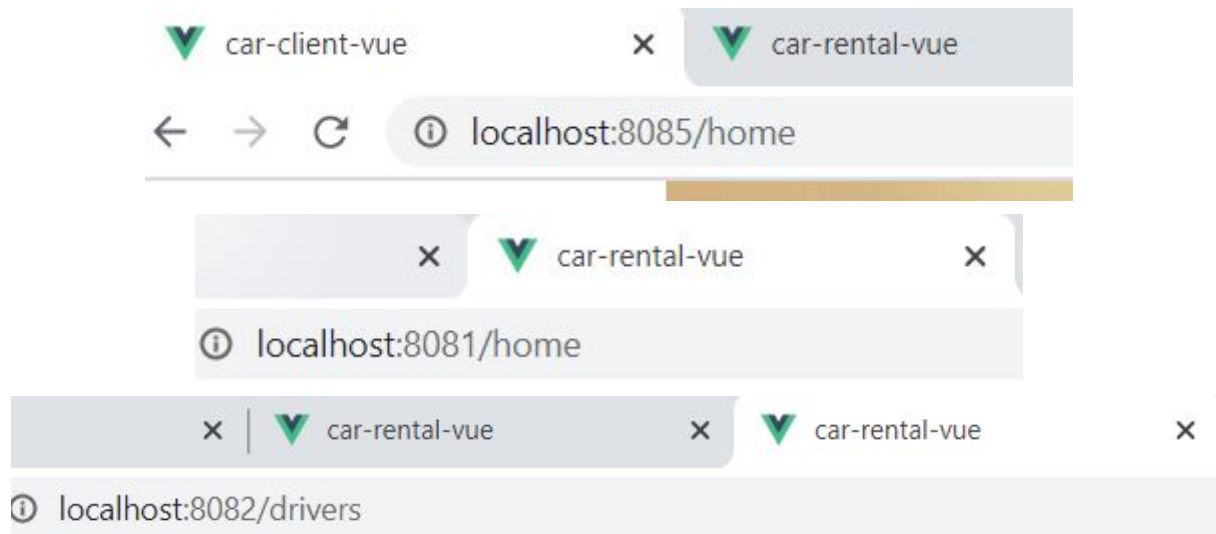
salon-container

OTHER

Open in Visual Studio Code

	salon-container_front_1	salon-container_front	RUNNING	PORT: 8081
	salon-container_browser_1	salon-container_browser	RUNNING	PORT: 4030
	salon-container_browser2_1	salon-container_browser2	RUNNING	PORT: 4040
	salon-container_front2_1	salon-container_front2	RUNNING	PORT: 8082
	salon-container_auth_1	salon-container_auth	RUNNING	PORT: 3000
	salon-container_auth2_1	salon-container_auth2	RUNNING	PORT: 3010
	salon-container_mongo_seed2_1	mongo	EXITED (0)	
	salon-container_mongodb2_1	mongo	RUNNING	PORT: 27018

	salon-container_mongo_seed2_1	mongo	EXITED (0)	
	salon-container_mongodb2_1	mongo	RUNNING	PORT: 27018
	salon2	salon-container_salon2	RUNNING	PORT: 4020
	salon-container_transport_1	salon-container_transport	RUNNING	PORT: 5002
	c5903d12ed98_mysql2	salon-container_mysql2	RUNNING	PORT: 3307
	salon-container_salon_1	salon-container_salon	RUNNING	PORT: 4010
	mysql	salon-container_mysql	RUNNING	PORT: 3306
	salon-container_mongo_seed_1	mongo	EXITED (1)	
	858bc4158dbf_salon-container_mongodb_1	mongo	RUNNING	



Opis instalacji aplikacji

Aby uruchomić aplikację należy wykonać poniższe komendy:

```
cd salon-container  
docker-compose build  
docker-compose up
```

```
cd car-client-vue  
docker-compose build  
docker-compose up
```