**Grupa: Mateusz Czerski, Valeria Ząbkowska, Wiktor Wolak-Książek (Laboratorium 1-2)**

**Wybór tematu: Platforma motoryzacyjna, do kupowania i sprzedawania samochodów (konkurencja otomoto)**

**Wybór technologii:** Frontend - angular 4, bootstrap, typeScript, html, css

Backend – java 8, spring 4, hibernate 4, jwt

Bazy Danych – mysql

System wersji: GIT

**Analiza przedmiotowa:**

Otomoto

-dostęp do ofert motoryzacyjnych

-filtrowanie ogłoszeń

-kontaktowanie się z osobami, które zamieszczają ogłoszenia

-dodawanie ogłoszeń

-usuwanie ogłoszeń

-rejestracja i logowanie

Aplikację pobrano ponad 1 000 000 razy od premiery 19.02.2013.

Aplikacja została przeznaczona do szerokiej grupy osób.

Samochody używane polska

-wyszukiwanie i ewentualne przekierowanie do innych portali motoryzacyjnych jak: otomoto,olx,allegro,gratka

Aplikację pobrano ponad 1 000 000 razy od premiery 19.10.2016.

Aplikacja została przeznaczona do szerokiej grupy osób.

Nasza aplikacja webowa jest konkurencyjnym portalem internetowym dla portalu otomoto. Aplikacja służy do przeglądania i dodawania ogłoszeń motoryzacyjnych. Posiada ona funkcjonalności dla administratora: dostęp do panelu administratora, usuwanie kont użytkowników, usuwanie i modyfikacja ogłoszeń, przeglądanie i przeszukiwanie ogłoszeń. Funkcjonalności dla użytkownika/gościa: tworzenie konta użytkownika, logowanie, przeglądanie i przeszukiwanie ogłoszeń. Funkcjonalności dla użytkownika zalogowanego: przeglądanie i przeszukiwanie ogłoszeń, dostęp do panelu użytkownika, dodawanie, modyfikacja i usuwanie własnych ogłoszeń. Dedykowana jest dla firmy, która chce zrobić konkurencję otomoto, dzięki mniejszym cenom za udostępnienie ogłoszenia.

**Analiza SWOT.**

|  |  |
| --- | --- |
| Mocne strony | Słabe strony |
| -dobry pomysł na konwencję - może przyciągnąć ludzi  - konkurencyjne ceny - tani dostawca  -atrakcyjne pakiety dla stałych użytkowników  -konta premium  -promocje  -przyjazny interfejs użytkownika (prosty w obsłudze)  -efekt świeżości  -niskie koszty utrzymania aplikacji  -system wykonany w nowoczesnej technologii | -niskie rezerwy finansowe  -nieznana marka  -brak doświadczenia ( pierwsza aplikacja w tym sektorze) |
| Szanse | Zagrożenia |
| -duży popyt motoryzacyjny  -mało aplikacji konkurencyjnych dla otomoto (nie każdemu mu się podobać)  - badania, z których wynika, że ludzie szukają czegoś nowego  -rosnąca popularność sprzedaży internetowej | -konkurencja z ustabilizowaną pozycją  -możliwość powstania nowych, konkurencyjnych platform z efektem świeżości  -brak sponsora |

**User history:**

**Użytkownik**

Jako niezalogowany użytkownik mogę przeglądać listę ogłoszeń kupna samochodu z bazy wraz ze szczegółami. Mogę wyszukiwać ogłoszenia po danych parametrach. Jestem w stanie założyć konto po weryfikacji e-mailowej. Później loguję się do systemu, dzięki czemu mam dostęp do panelu użytkownika. W panelu mogę dowolnie zmieniać swoje dane. Jeżeli zechcę to mogę usunąć swoje konto w każdej chwili. Posiadając konto mogę dodawać własne ogłoszenia sprzedaży auta, modyfikować je oraz usuwać. Po skończeniu pracy w serwisie mogę się wylogować.

**Admin**

Po zalogowaniu posiadam dostęp do panelu admina. Mogę przeglądać wszystkie ogłoszenia sprzedaży aut oraz dodawać i usuwać własne. Jestem w stanie modyfikować i usuwać ogłoszenia dodane przez innych użytkowników. Mam dostęp do bazy użytkowników gdzie mogę usunąć każde konto w systemie. Mogę dowolnie modyfikować swoje własne dane na koncie admina.

**Metodyka zarządzania projektem – model klasyczny**

Do budowy naszej aplikacji wybraliśmy model klasyczny z powodów:

-cel projektu jest dobrze zdefiniowany

-będzie to duży projekt, z perspektywą utrzymania i dalszego rozwoju funkcjonalności biznesowych

-stabilne wymagania projektowe, brak zmian

-dostępność klienta (klient może być dostępny dopiero podczas fazy odbioru oraz testów akceptacyjnych)

-ściśle określony zakres oraz koszt wykonanych prac

-czas dostarczenia (możliwość dostarczenia na sam koniec projektu)

-dzięki korzystaniu z metodyki klasycznej zespół może zawierać osoby początkujące, pracujące pod nadzorem doświadczonych. Nie jest to możliwe w metodyce zwinnej przez co każda osoba powinna być „samowystarczalna i doświadczona”

-preferencje klienta. Znajomość metodyk przez klienta jest kluczem do sukcesu projektu. Realizacja projektu w nieznanej klientowi metodyce spowoduje chaos oraz irytację.

**Założenia funkcjonalne i niefunkcjonalne**

Celem projektu jest stworzenie serwisu online z ogłoszeniami motoryzacyjnymi. Poniżej zostały zestawione ogólne założenia funkcjonalne i niefunkcjonalne aplikacji oraz funkcje programu dostępne dla poszczególnych grup użytkowników – niezalogowanych (gości), zalogowanych oraz administratora.

**Założenia funkcjonalne:**

W systemie przewidziane są 2 role: użytkownik i administrator.

Funkcje dostępne dla administratora:

- dostęp do panelu administracyjnego,

- usuwanie kont użytkowników.

- usuwanie i modyfikacja ogłoszeń dodanych przez użytkowników.

- przeglądanie listy ogłoszeń motoryzacyjnych i ich szczegółów,

**- przeszukiwanie ogłoszeń (parametryzowane).**

Funkcje dostępne dla użytkownika niezalogowanego(gościa):

- tworzenie konta użytkownika wraz z jego weryfikacją via email,

- logowanie do systemu (podanie loginu/email i hasła),

- przeglądanie listy ogłoszeń obecnych w bazie i ich szczegółów,

**- przeszukiwanie ogłoszeń (parametryzowane).**

Funkcje dostępne dla użytkownika zalogowanego:

- przeglądanie listy ogłoszeń obecnych w bazie i ich szczegółów,

**- przeszukiwanie ogłoszeń (parametryzowane),**

- dostęp do panelu użytkownika,

- dodawanie własnych ogłoszeń,

- modyfikacja własnych ogłoszeń,

- usuwanie własnych ogłoszeń,

- usuwanie własnego konta,

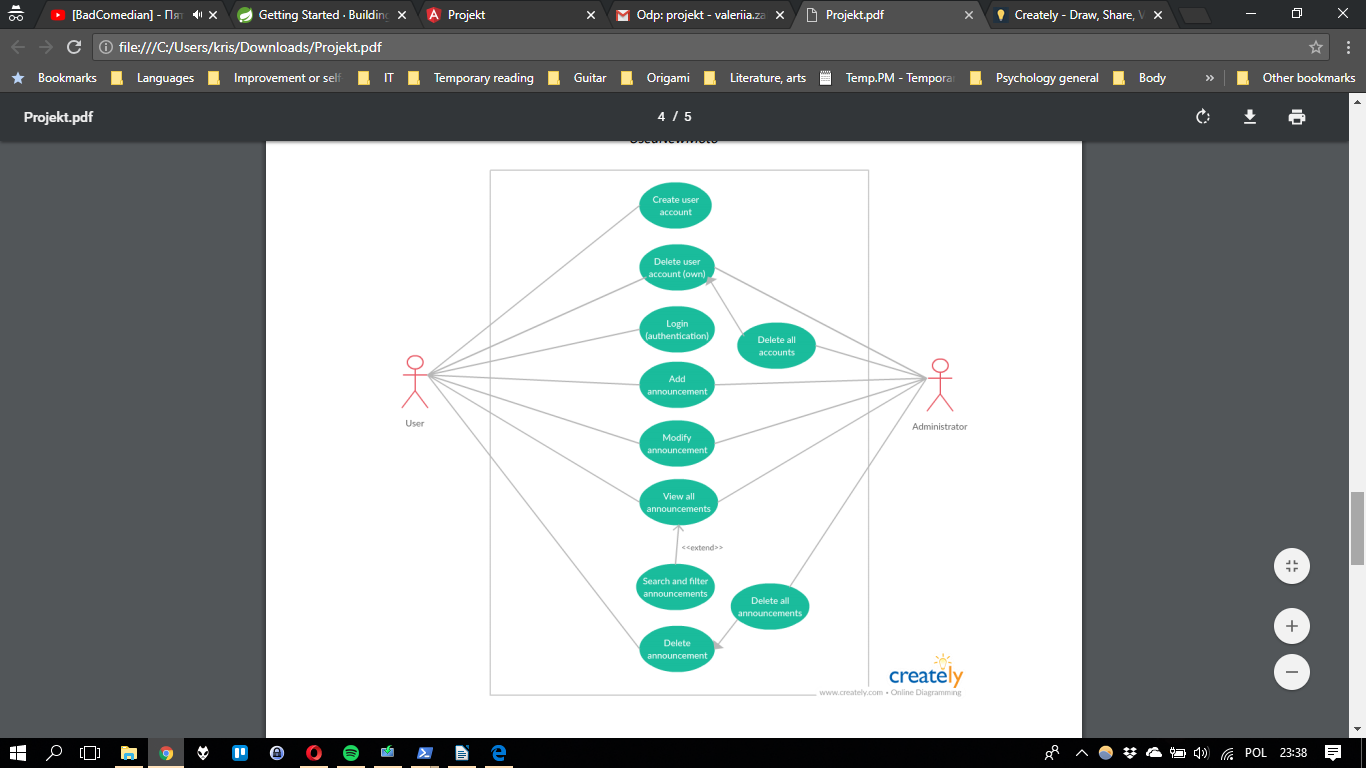
- wylogowanie z aplikacji.

**Założenia niefunkcjonalne:**

- oprogramowanie jest responsywne i automatycznie dostosowuje się do różnych rozdzielczości ekranu;

- aplikacja ma działać oraz wyglądać podobnie w najpopularniejszych przeglądarkach.

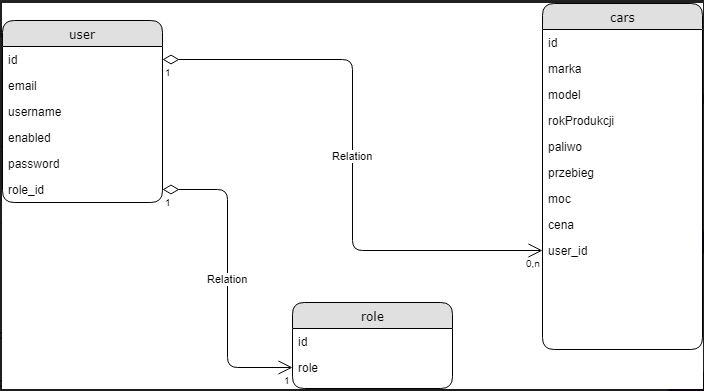
Diagram przypadków użycia:



Na powyższym diagramie przypadków użycia dotyczącego możliwych do wykonania akcji w portalu ogłoszeń motoryzacyjnych, widoczni są dwaj aktorzy, którymi są Użytkownik oraz Administrator serwisu. Użytkownik niezalogowany (gość) ma do wyboru akcje dotyczące przeglądania, wyszukiwania, filtrowania wszystkich ogłoszeń oraz rejestracji konta użytkownika. Użytkownik po poprawnej rejestracji i zalogowaniu może dodatkowo dokonywać akcje związane z dodawaniem, modyfikacją i usuwaniem ogłoszenia, a także usuwaniem konta. Administrator może dokonać akcji związanych z usuwaniem konta użytkownika oraz ogłoszeń, a także może podejmować akcje związane z dodawaniem, modyfikowaniem i wyświetlaniem ogłoszeń.

**Model danych**

Diagram encji ERD na podstawie modelu bazy danych



Opracowanie i wdrożenie zabezpieczeń.

Cross-origin resource sharing (w skrócie CORS) – mechanizm umożliwiający współdzielenie zasobów pomiędzy serwerami znajdującymi się w różnych domenach. Ściślej rzecz biorąc chodzi o możliwość wykonywania żądań AJAX między takimi serwerami przy zachowaniu pewnych ograniczeń co do dopuszczalnego źródła żądania.

Wykorzystano w aplikacji przy filtrze uwierzytelnienia JWT. Łączy on aplikacje front-end z back-end.

JWT -(JSON Web Tokens) to otwarty standard (RFC 7519), który definiuje sposób wymiany danych między stronami w bezpieczny sposób poprzez obiekt JSON. Przesyłane informacje mogą być weryfikowane dzięki cyfrowemu podpisowi, który jest elementem tokenu.

Token JWT jest podpisany za pomocą sygnatury – algorytmem HMAC lub za pomocą klucza publicznego/prywatnego RSA lub ECDSA.

JWT może być wykorzystany przy:

1. Autoryzacji – JWT znajduje szerokie zastosowanie w autoryzacji kiedy jedna ze stron chce przyznać dostęp drugiej do zasobów i serwisów, a później bez przechowywania stanu po swojej stronie weryfikować czy dostęp powinien być możliwy.

2. Transmisji danych – Kiedy chcemy przesłać pomiędzy stronami informacje i potrzebujemy mieć pewność, że nadawca jest tym za kogo się podaje i dane które wysyła nie zostały zmienione. Możemy to zweryfikować właśnie dzięki cyfrowemu podpisowi, które jest częścią JWT.

JWT w swojej wynikowej postaci (jako token) składa się z trzech części oddzielonych od siebie kropkami.

Przykład:

Aaa.bb.ccc

Te części to kolejno:

• Nagłówek (Header)

• Zawartość (Payload)

• Sygnatura (Signature)

Każda usługa w aplikacji jest zabezpieczona przez JWT, gdy żądanie jest wysyłane przez front-end następuje przekierowanie go na filtry uwierzytelniające jego podpis (JWT). Po dekodowaniu wiadomości o konkretny algorytm hashujący z wykorzystaniem klucza, następuje sprawdzenie poprawności credentiali z zapisanymi na bazie danych. Dodatkowo hasła na bazie danych również są zahashowane algorytmem tym samym co generowanie podpisu JWT. Po poprawnym przejściu procesu weryfikacji, żądanie dostaje uprawnienia do korzystania z usług. Dodatkowo podczas procesu uwierzytelnienia, następuję określenie roli i uprawnień danego żądania.

**Proponowany model**

Projekt opiera się na tradycyjnym modelu MVC (Model-View-Controller).

Model – warstwa klas, które będą wyświetlały odpowiednie informacje. Opisuje logikę biznesową oraz model danych. Encje oraz wyniki zapytań bazodanowych z MySql’a.

View – warstwa projektu (w tym przypadku frontend), która ma za zadanie wyświetlać poprzez UI odpowiednie komponenty. Wyświetlanie odpowiednich danych otrzymanych od kontrolera. Ma za zadanie wyświetlać też zmiany dokonane w warstwie modelu.

Controller – jest odpowiedzielany za przetwarzanie zapytań od użytkownika. Przetwarza odpowiednio dane poprzez Model i zwraca je z powrotem do warstwy View. Jest mediatorem pomiędzy modelem a widokiem. Projekt backendowy napisany w Javie, który udostępnie usługi przez API Restowe.

**Opracowywanie warstwy integracji.**

Na podstawie zaproponowanego modelu oraz wzorca projektowego opracowano warstwę integracji. Częścia odpowiedzialną za widok(VIEW) będzie projekt frontendowy napisany w Angularze. Model będzie przechowywał model danych projektu w tabelach w bazie MySql. Kontrolerem będzie aplikacja backendowa napisana w Javie, która za pomocą klas będzie mapowała encje bazodowe oraz odpowiednio je obsługiwała i zwracała do widoku poprzez API Restowe.

**Ocena możliwości zastosowań różnych sposobów modelowania**

MVP- Prezenter zastępujący kontroler może wpływać na zmiany pokazywane przez View. W naszym wypadku wystarczy, że widok będzie wyświetlał zmiany zachodzące w modelu.

MVVM- wspiera on dwukierunkowe przesyłanie informacji pomiędzy częściami View a View-Model. Nasz projekt nie jest na tyle skomplikowany by wymagał automatycznego propagowania zmian, np. wysyłania zmian z View-Model do części View. UI nie musi w naszym projekcie w żaden sposób wpływać na warstwę biznesową.

**Przedstawienie perspektyw wymagań**

Projekt może w przyszłości rozszerzyć się o nowe funkcjonalności:

-Sprzedaż innych produktów, o innej kategorii ale związanych z motoryzacją np. części samochodowe

-Sprzedaż nowych przedmiotów w innej branży, np. mieszkania, zabawki.

-Wprowadzenie systemu licytacji sprzedaży

Wprowadzenie sprzedaży produktów innych kategorii, nie powinno być problemem dla projektowanego systemu. Model danych oraz warstwy integracyjne są przygotowane na pojawienie się nowych kategorii i produktów. Projekt Angularowy może bez problemu zintegrować się z nowymi warstwami API prezentując nowe ‘zakładki’ czy nawet oddzielny system czy stronę internetową opartą na tym aktualnym projekcie. Aplikacja Javowa jest zaprojektowana w architekturze pozwalającej na dodanie nowych serwisów obsługujących nowe produkty. Z tego względu można stwierdzić, iż system jest gotowy na rozbudowę i mało kosztowne rozszerzenie swojej funckjonalności.

**Diagramy UML**

Diagram przypadków użycia

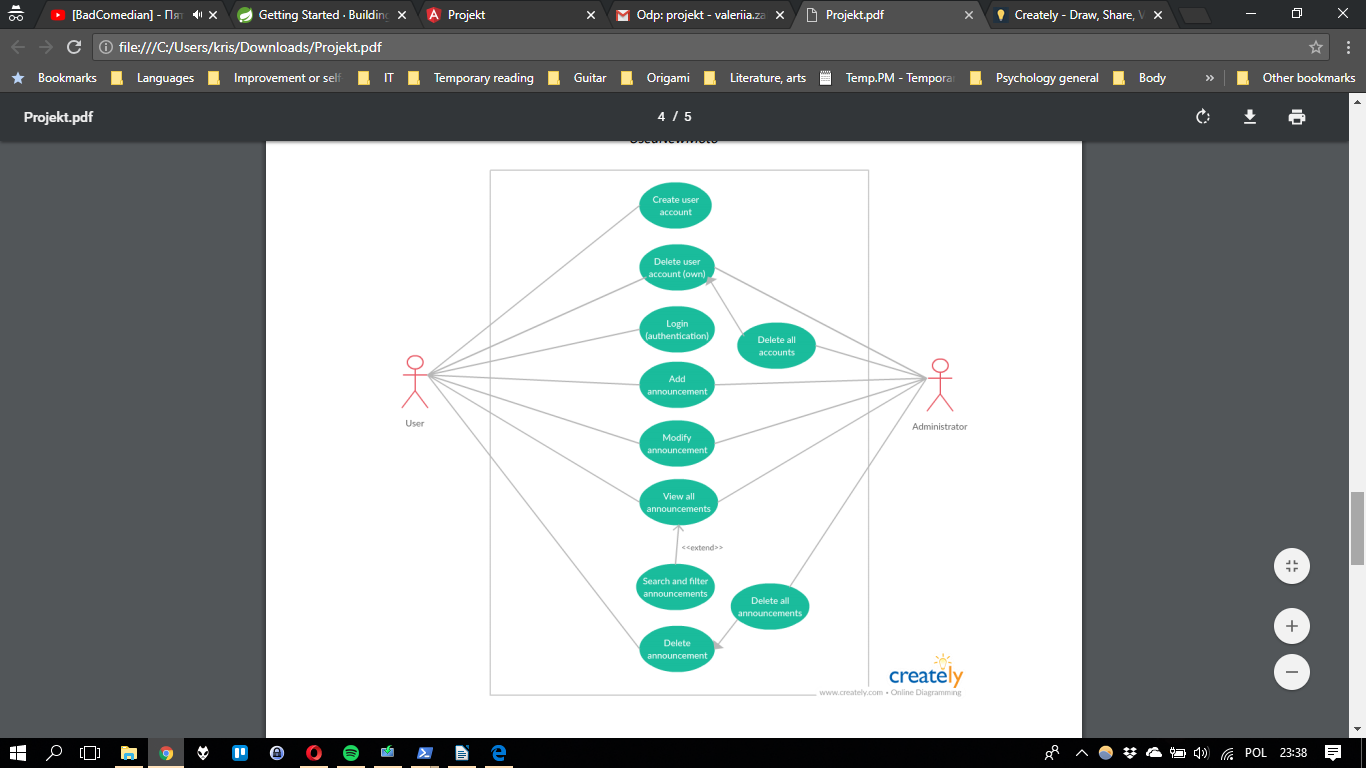


Diagram klas:

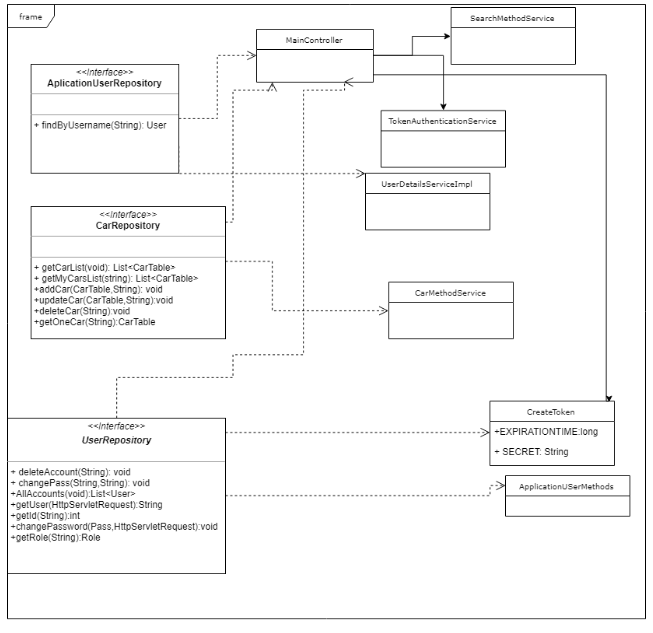


Diagram pakietów

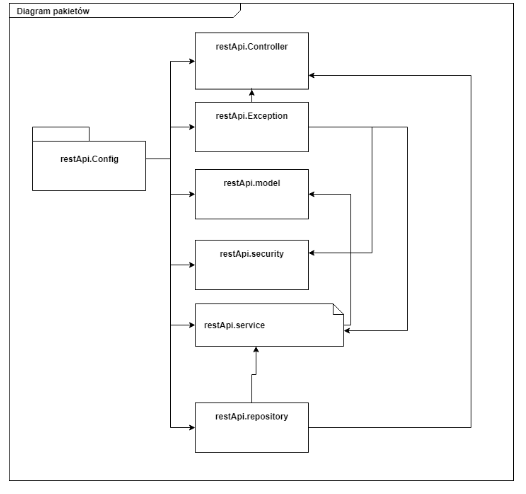


Diagram wdrożeniowy

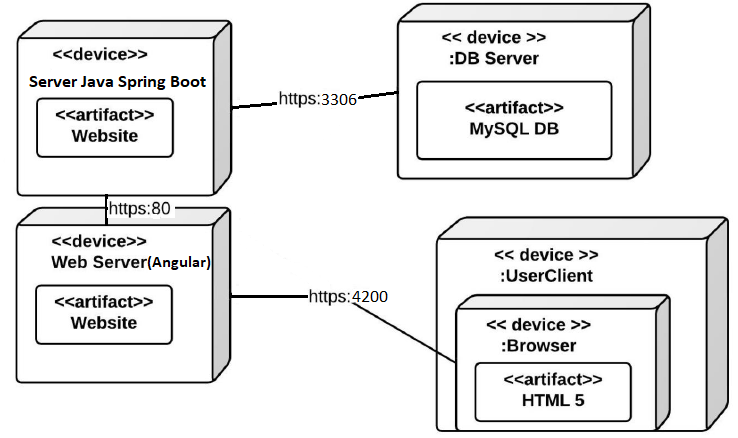


Diagram struktur połączonych

Ze względu na duże podobieństwo do diagramu wdrożeniowego oraz brak wnoszenia użytecznych informacji został pominięty.

Diagram czynności dla dodania oferty:

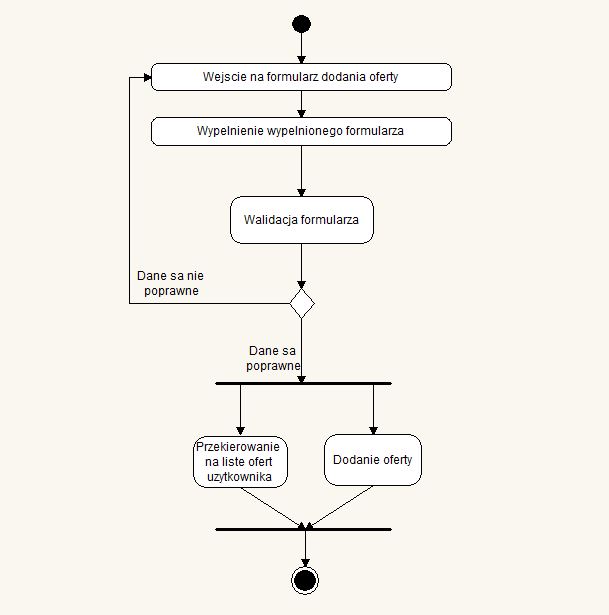


Diagram czynności dla rejestracji:

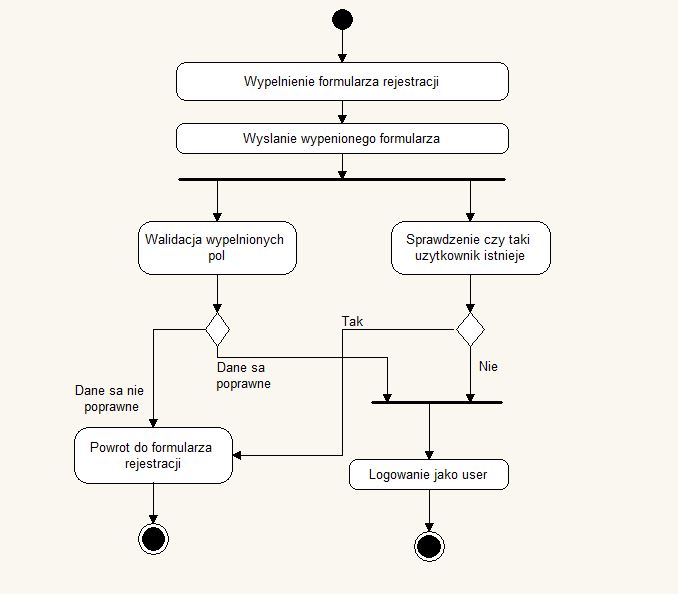


Diagram stanów dla logowania:

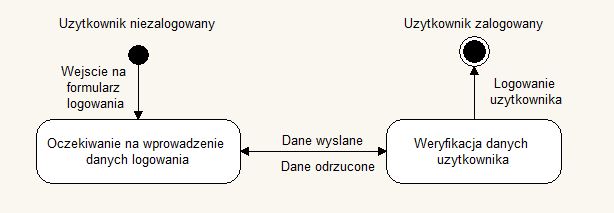
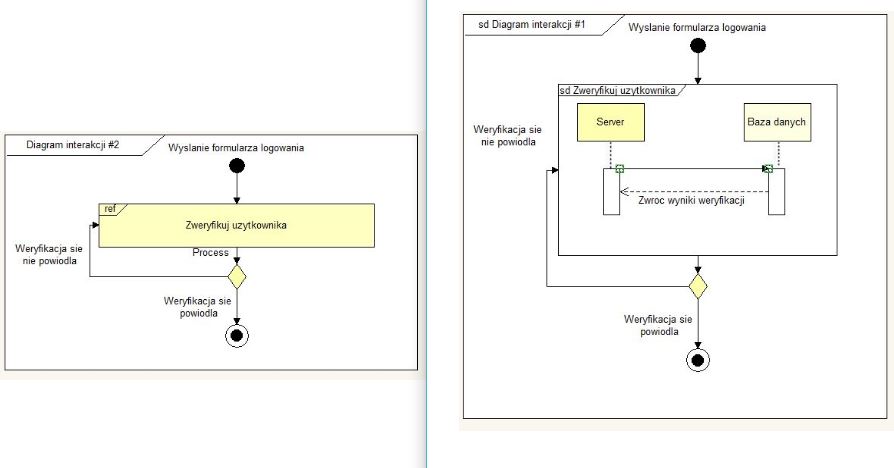
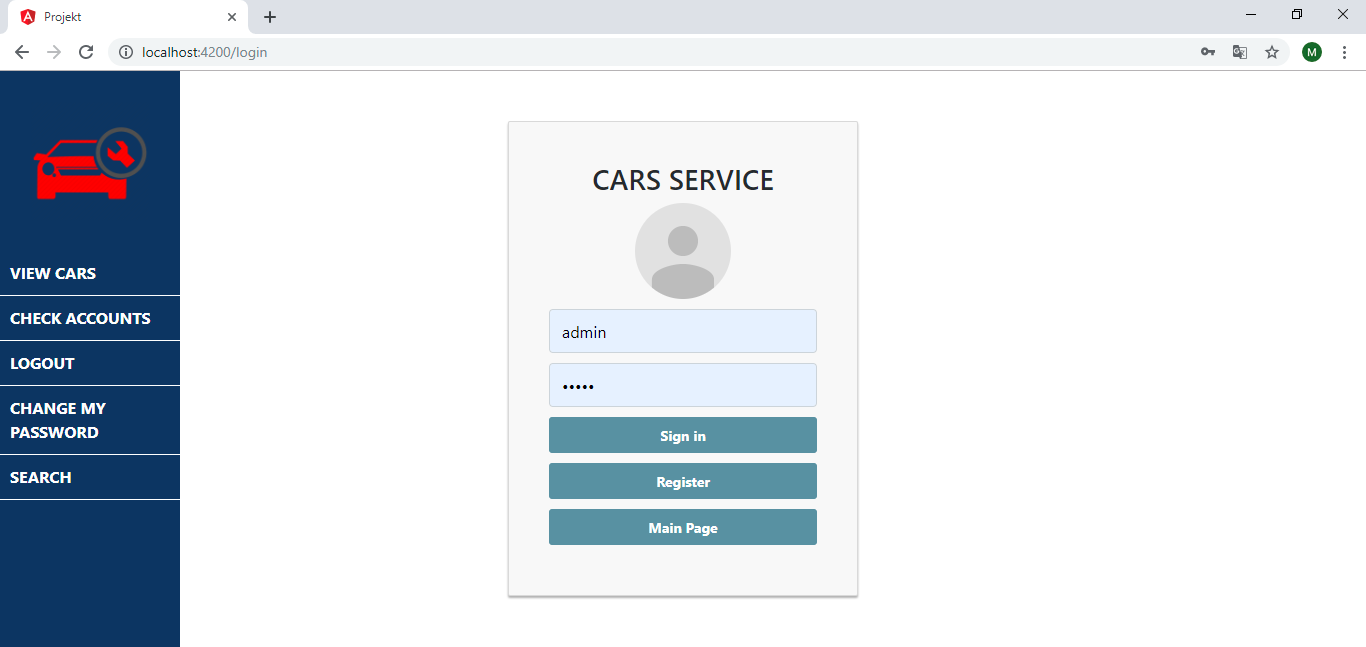


Diagram interakcji dla logowania:



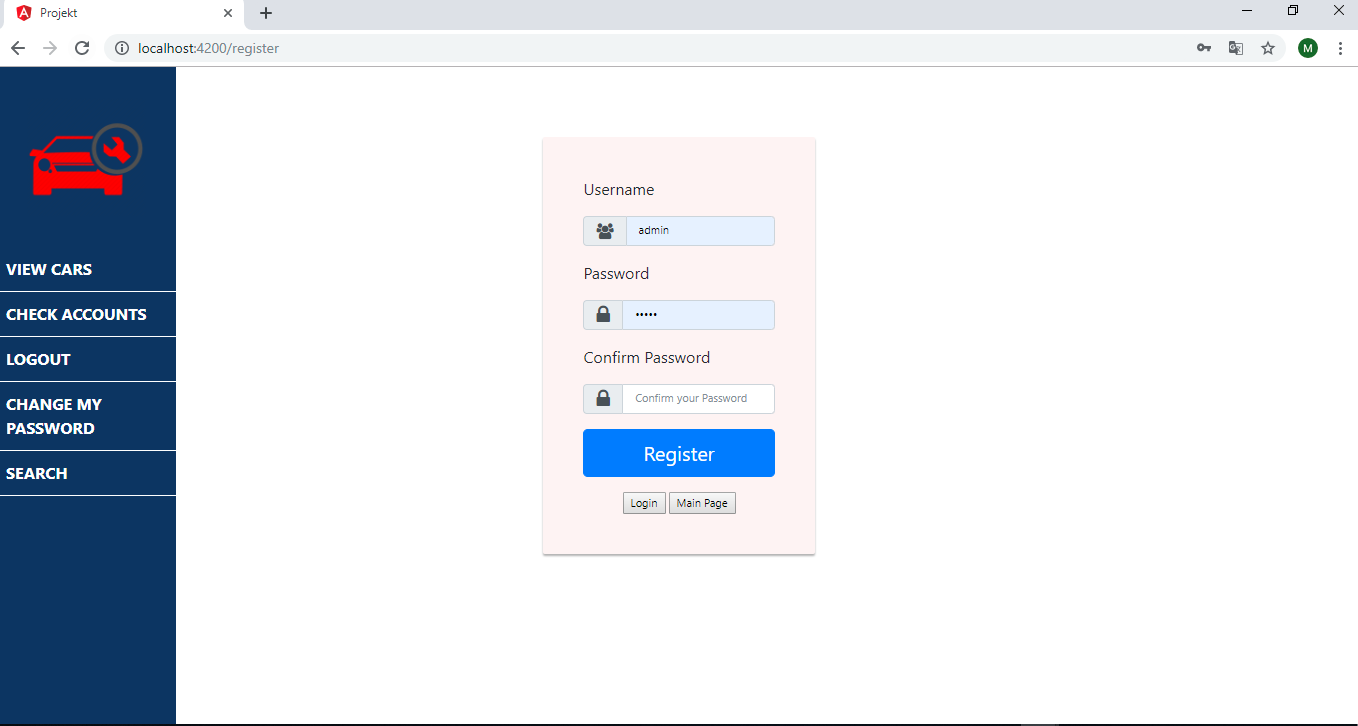
Opracowanie warstwy prezentacji:

Screen strony głównej

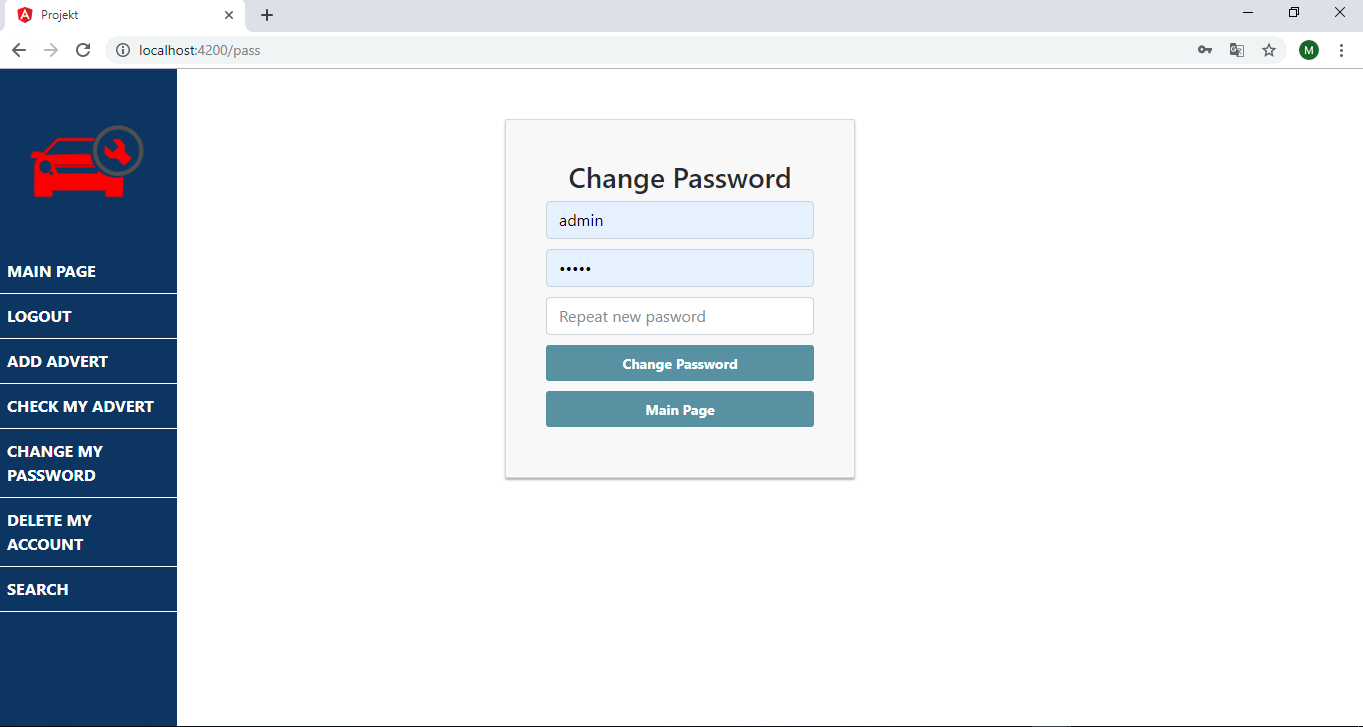


Przedstawiono panel logowania, zgodnie z wymaganiami klienta dotyczącymi rozmieszczenia elementów na stronie, koloru grafiki i typu czcionki.

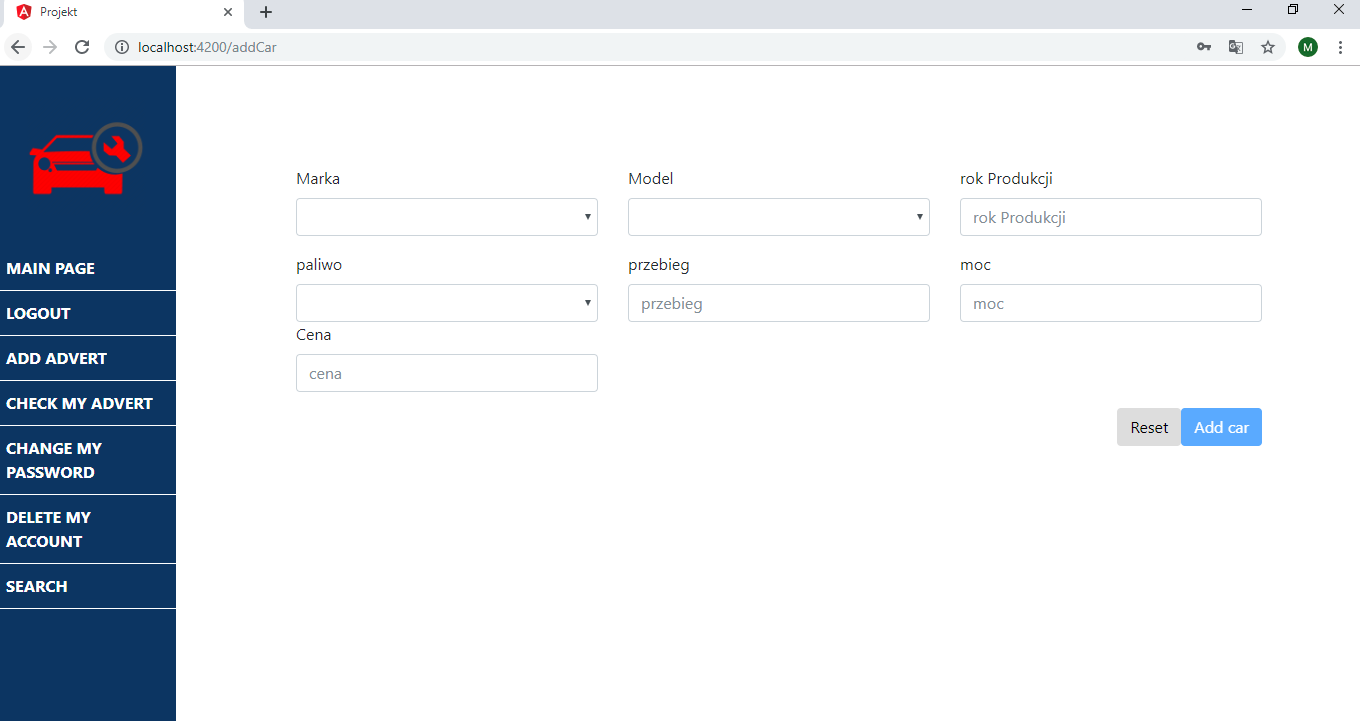
Poniżej znajduje się panel rejestracji:



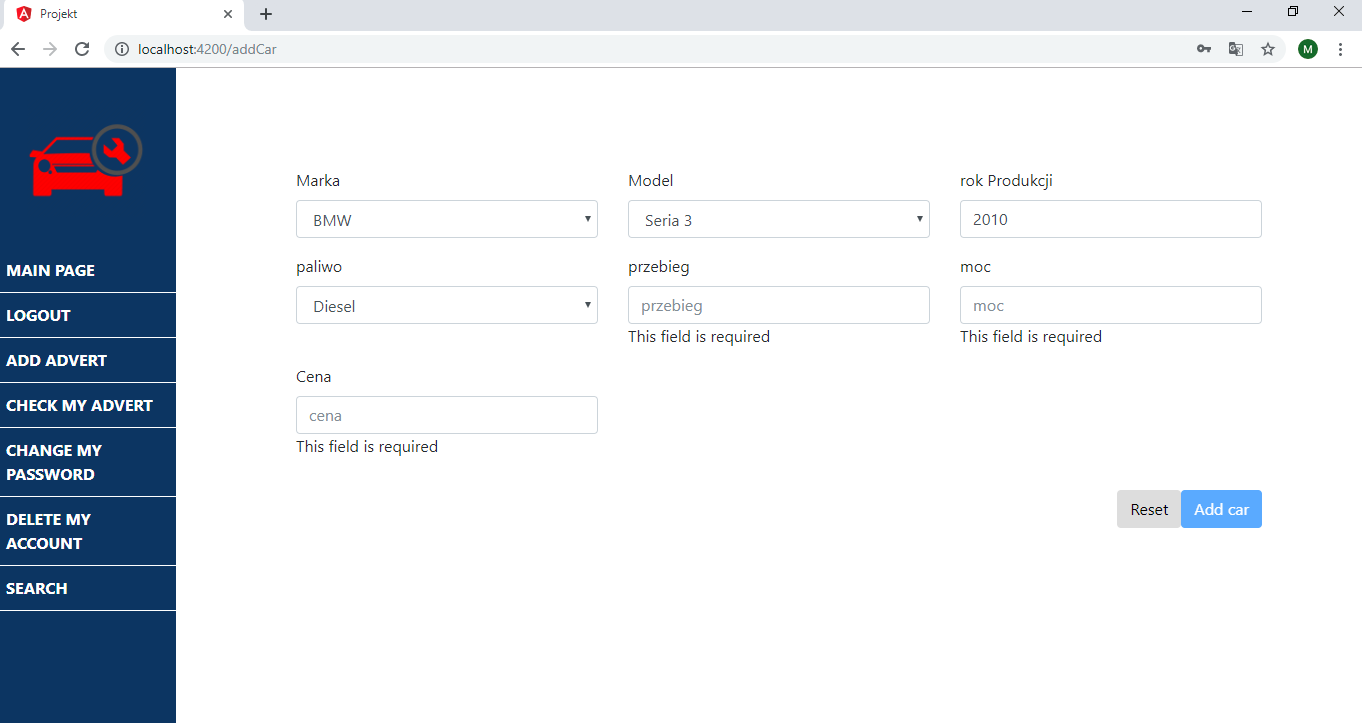
Dla zalogowanego użytkownika istnieje możliwość zmiany swoje hasła poprzez formularz, przedstawiony poniżej:



Strona formularza dodania nowej oferty dla załogowanego użytkownika:



Walidacja formularza na froncie nie pozwała dodawać ogłoszenia bez wszystkich wypełnionych pół:



Strona z formularzem dla wyszukiwania ofert

