****

**Programowanie obiektowe**

*Tytuł projektu*

Prowadzący: Autor:

mgr inż. Ewa Żesławska Jakub Kruszyński

nr albumu

Kierunek: 3 IIZ/2021, grupa GP02

Rzeszów, r.a. 2022/2023

Spis treści

[1. Cel Projektu 3](#_Toc127241531)

[1.1. Wymagania funkcjonalne 3](#_Toc127241532)

[1.2. Wymagania niefunkcjonalne 3](#_Toc127241533)

[2. Opis techniczny projektu 3](#_Toc127241534)

[3. Harmonogram realizacji projektu 4](#_Toc127241535)

[4. Prezentacja warstwy użytkowej projektu 4](#_Toc127241536)

[4.1.Uruchomienie programu 4](#_Toc127241537)

[4.2. Wygląd aplikacji z poziomu admina 5](#_Toc127241538)

[4.3.Wygląd aplikacji z poziomu użytkownika 9](#_Toc127241539)

[4.4. Zabezpieczenia 13](#_Toc127241540)

[5. Repozytorium, system kontroli wersji 14](#_Toc127241541)

[6. Podsumowanie 14](#_Toc127241542)

[7. Literatura 14](#_Toc127241543)

# Cel Projektu

Celem mojego projektu było stworzenie aplikacji w języku Java połączonej z relacyjną bazą danych, która umożliwi zarządzanie przez właściciela wypożyczalnią samochodów oraz da możliwość korzystania z usług klientom tej oto wypożyczalni. Projekt został wykonany w formie aplikacji Webowej która umożliwia nam uruchomienie programu w przeglądarce, utworzenie wielu kont (z których każde przypisywane jest do użytkowników) , zalogowanie się oraz zarządzanie od strony admina całym systemem wypożyczalni.

## 1.1. Wymagania funkcjonalne

• Utworzenie nowych kont, zalogowania się na już istniejące oraz wylogowania się

• Dodawanie nowych samochodów do wypożyczenia, pod warunkiem że zostały utworzone pewne oddziały (od strony admina)

• Dodawanie nowych oddziałów z których będzie można wypożyczać samochody (od strony admina)

• Dodawanie Typu Samochodu(należy podać atrybuty, należy podać sposób identyfikacji).

• Sprawdzanie historii zgłoszeń złożonych przez użytkowników oraz możliwość edytowania ich (od strony admina)

• Przeglądanie istniejących ofert samochodów do wypożyczenia (od strony użytkownika)

• Wynajmowanie poprzez wybór daty początkowej i końcowej z kalendarza wraz z obliczaniem opłat za sumaryczną długość wypożyczenia auta

• możliwość sprawdzanie prywatnej historii zgłoszeń dla każdego użytkownika

## Wymagania niefunkcjonalne

• Program jest przyjazny, przejrzysty oraz prosty w użyciu przez użytkowników

• Aplikacja nawiązuje połączenie z bazą danych i używa rekordów w niej zapisanych

• Aplikacja stworzona jest w języku Java z frameworkiem Spring Boot

• Aplikacja działa na systemach, które posiadają przeglądarkę internetową.

# Opis techniczny projektu

Projekt jest zbudowany w oparciu o framework Spring Boot oraz biblioteke szablonów Thymeleaf)

• Środowisko programistyczne Javy: Java JDK 19

• Środowisko programistyczne: IntelliJ IDEA Community Edition 2022.3.2

• Baza danych: MySQL

• Spring Boot w wersji 2.7.8

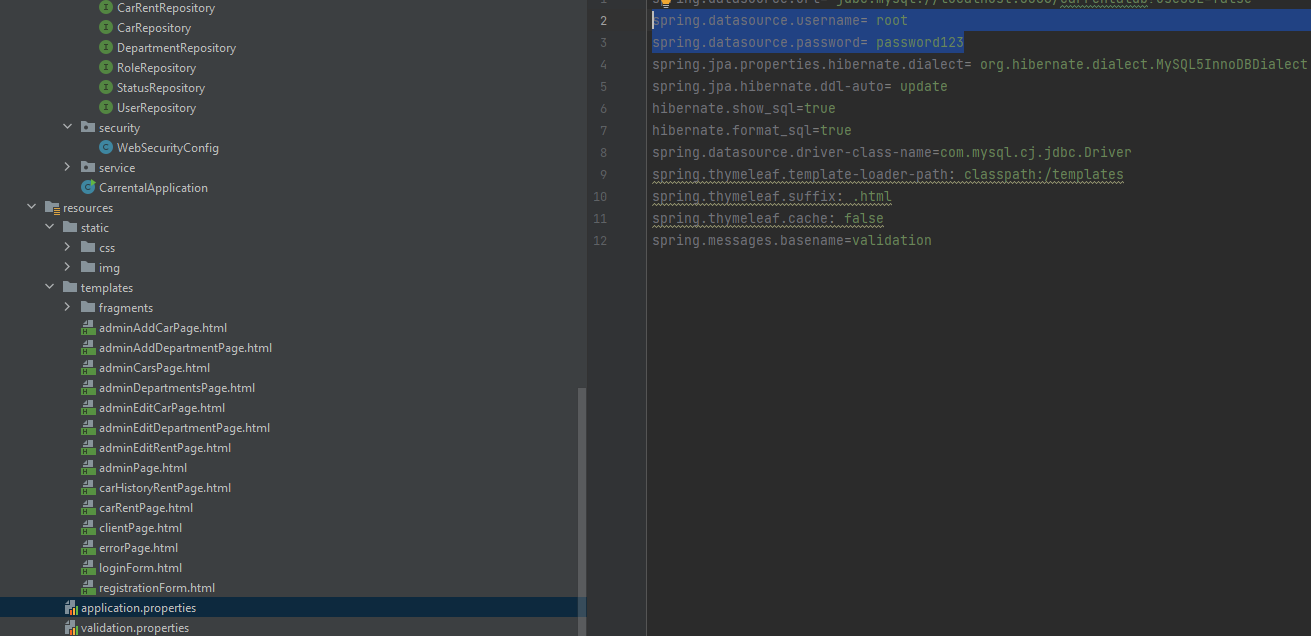
# Harmonogram realizacji projektu

# Prezentacja warstwy użytkowej projektu

## 4.1.Uruchomienie programu

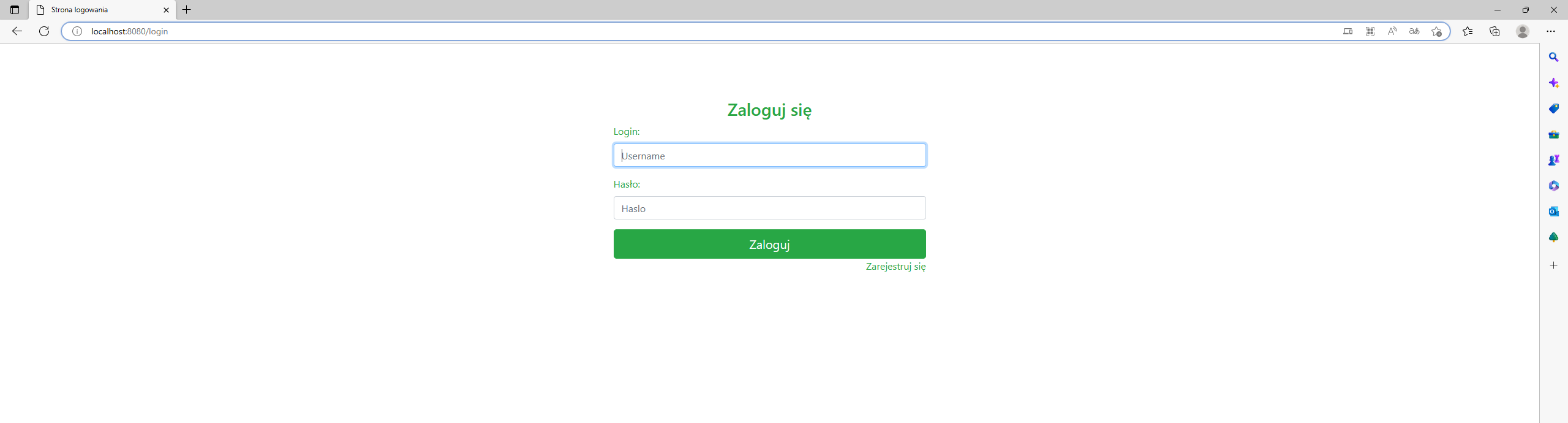
W celu uruchomienia aplikacji do obsługi systemu wypożyczalni samochodów należy wykonać następujące kroki:

• uruchomić oraz zalogować się do bazy danych MySQL, używając domyślnych danych bądź zmieniając w 2 i 3 linijce zakładki application.properties hasło oraz login



**Rysunek 1 .Dane do bazy danych**

Następnie wracamy do kodu programu w Javie i uruchamiamy go. Kolejnym krokiem jest otworzenie wybranej przeglądarki internetowej i wpisanie w niej formuły: **localhost:8080**

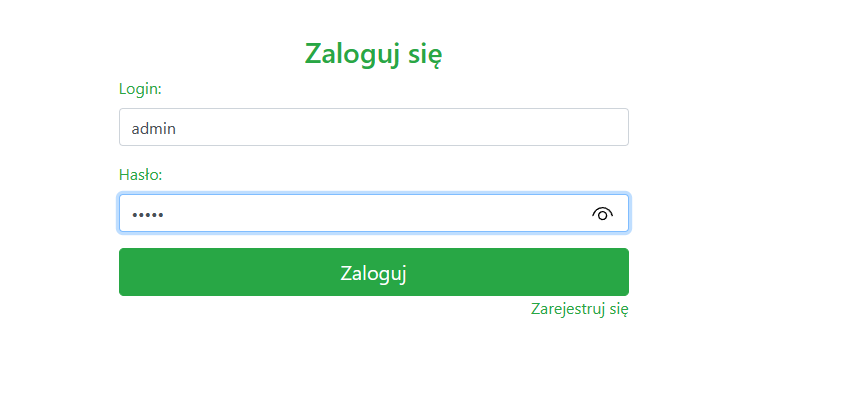


**Rysunek 2 . Panel logowania**

Jeśli wszystkie kroki zostały wykonane prawidłowo, to powinien nam się ukazać widok jak w zrzucie ekranu powyżej i możemy śmiało korzystać z aplikacji

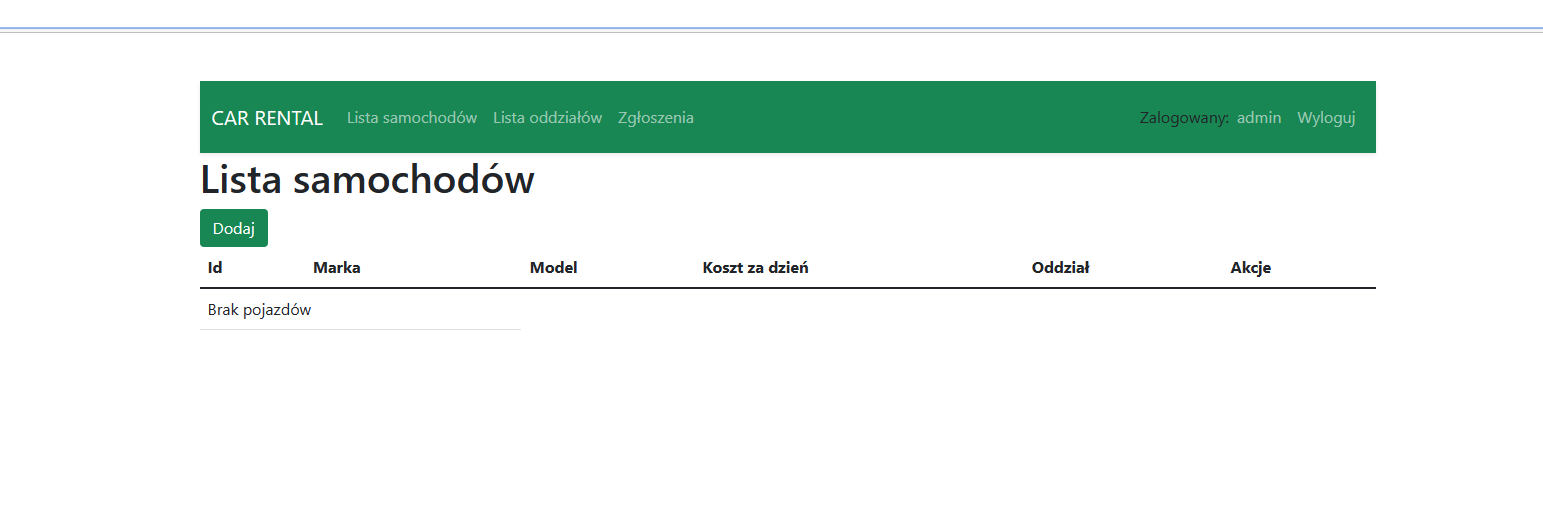
## 4.2. Wygląd aplikacji z poziomu admina

Po uzyskaniu dostępu do aplikacji należy zalogować się jako admin aby stworzyć oferty z których będą mogli skorzystać klienci wypożyczalni, gdyż zgodnie z założeniami jest ona standardowo pusta jednak każda zmiana wprowadzona później zostanie zapamiętana.



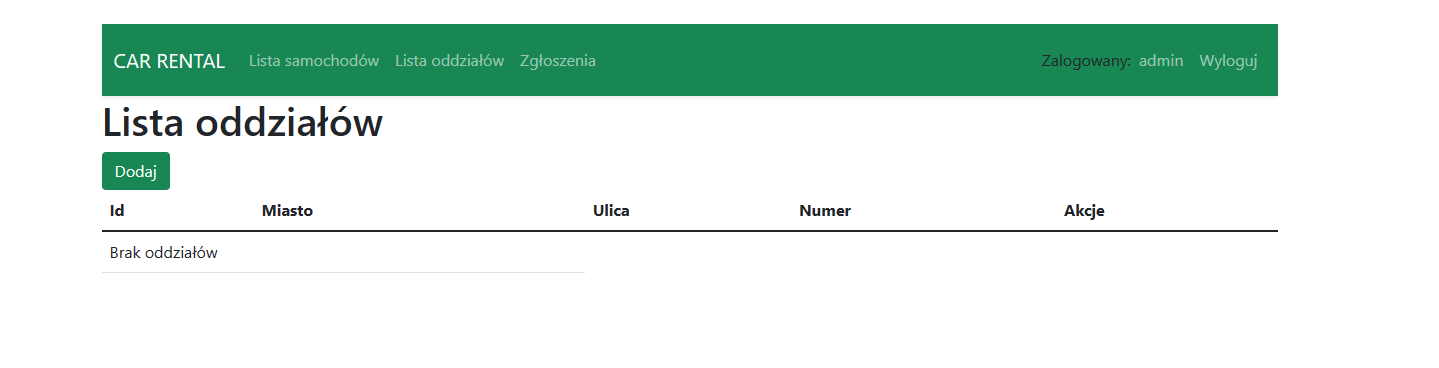
**Rysunek 3 . Logowanie jako admin**

Aby zalogować się jako admin, w polu login musimy wpisać: admin , a jako hasło wprowadzić także: admin

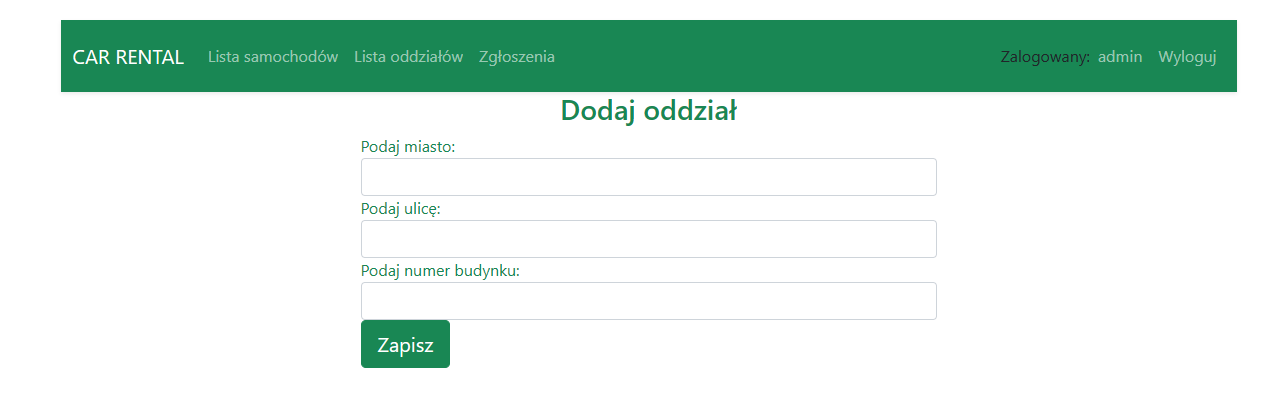


**Rysunek 4 .Menu główne**

Tak wygląda menu główne aplikacji. Zgodnie z tym co wcześniej wspomniałem, wszystkie rubryki są puste, dlatego zabieramy się za dodawanie, gdyż są one konieczne do dodania samochodu do listy

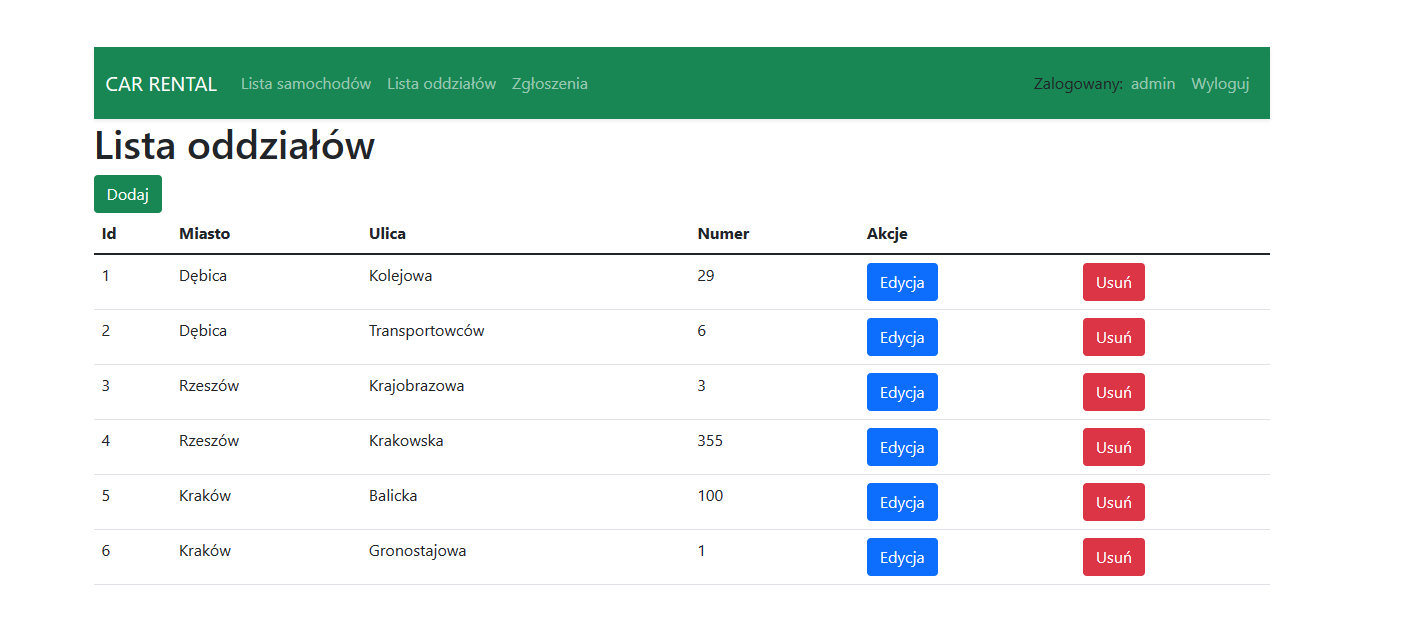


**Rysunek 5 . Pole oddziały**



**Rysunek 6 . Dodawanie oddziałów**

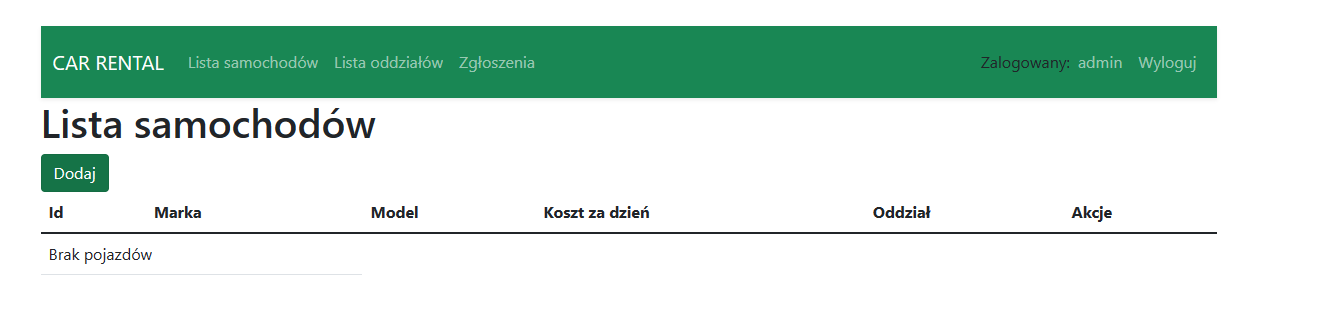
Tak wygląda menu dodawania oddziałów, teraz stworzę kilka różnych i pokażę jak wyglądają na liście.



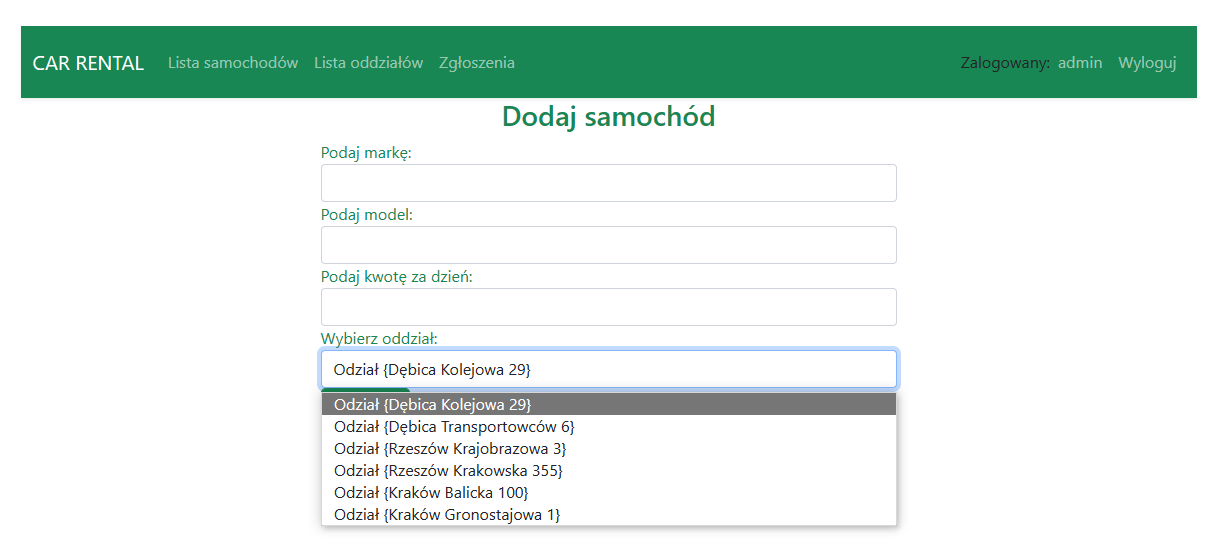
**Rysunek 7 . Gotowe oddziały**

Admin może dodatkowo edytować każdy z istniejących już oddziałów, bądź go usuwać.

Teraz przechodzę do stworzenia samochodów możliwych do wypożyczenia.

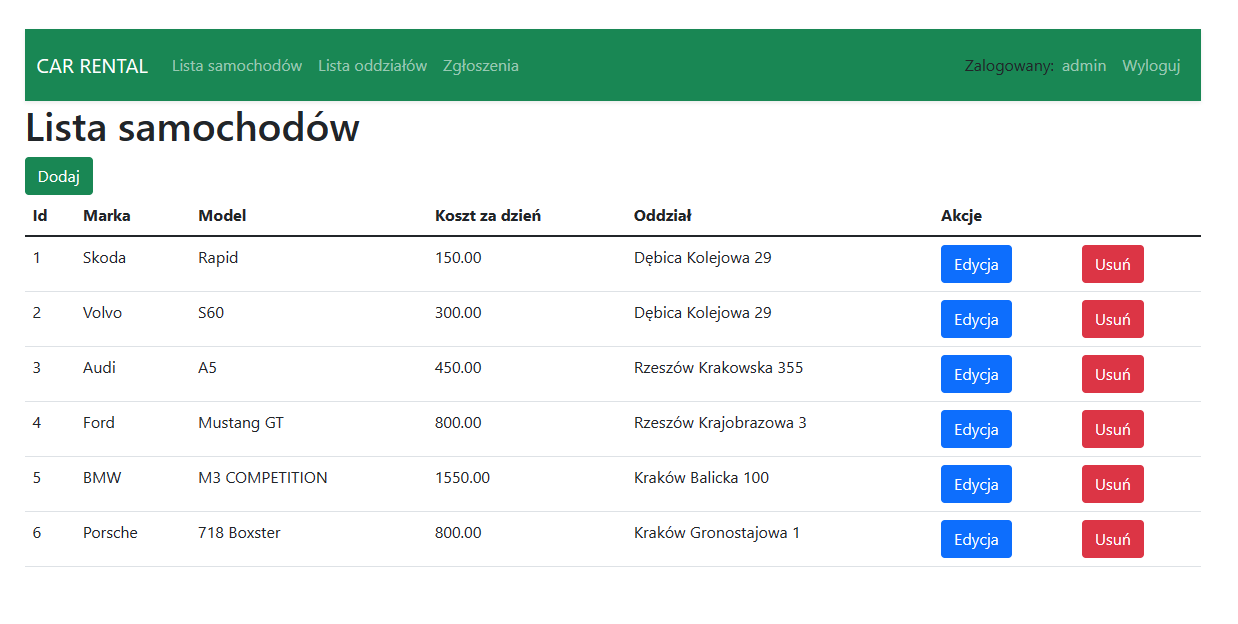


**Rysunek 8 . Lista samochodów**



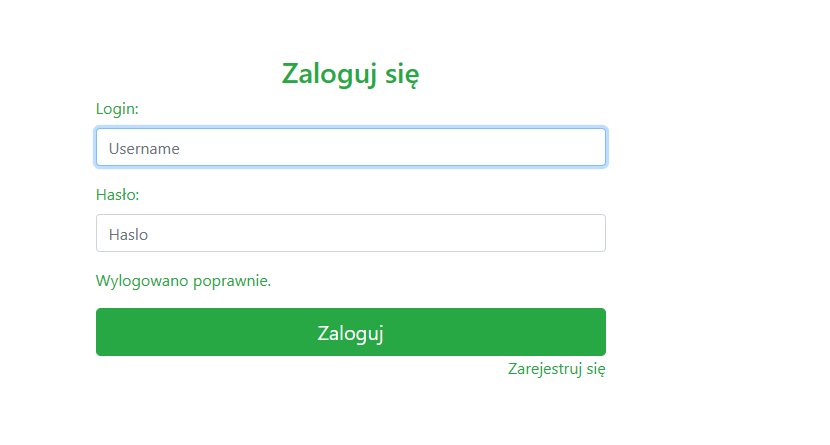
**Rysunek 9 . Menu dodawania samochodów**

Dodatkową funkcją podczas podawania kwoty jest oprócz możliwości wpisania ręcznie, opcja zwiększania bądź zmniejszania kwoty o 1 za pomocą suwaka który pojawia się dopiero po najechaniu na pole. Podczas najechania na pole z „Wybierz oddział” natomiast rozwija nam się lista z możliwością wybrania konkretnego z utworzonych oddziałów oraz nie ma możliwości ręcznego wpisania oddziału w to pole. Oryginalnie pod rubryką „Wybierz oddział” jest przycisk Zapisz, jednak chciałem pokazać jak zachowuje się pole z oddziałami.



**Rysunek 10 . Przykładowo utworzona lista samochodów**

Tak wygląda stworzona przeze mnie lista samochodów, w której pozycje admin może później edytować tak samo jak w przypadku oddziałów. Jak na razie prezentację rubryki „Zgłoszenia” pomijam z powodów oczywistych – nie istnieją a admin jako osoba nadrzędna może jedynie edytować istniejące zgłoszenia a nie tworzyć własne, gdyż od tej drugiej funkcji odpowiedzialni są klienci.

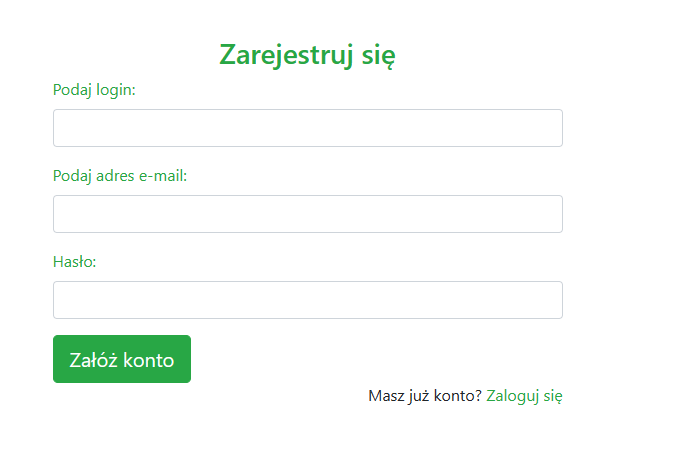


**Rysunek 11 . Menu po wylogowaniu się**

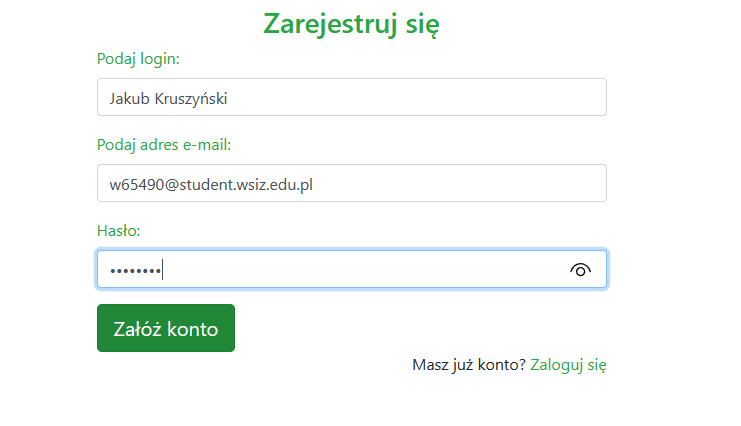
W celu przejścia do następnej części wylogowuję się i przechodzę do Użytkownika.

## 4.3.Wygląd aplikacji z poziomu użytkownika

Na sam początek należy kliknąć na napis „Zarejestruj” się w celu stworzenia konta na które będziemy mogli później się logować. Wpisywanie danych w pola „Login” i ”Hasło” jak na razie nie dadzą żadnego rezultatu.

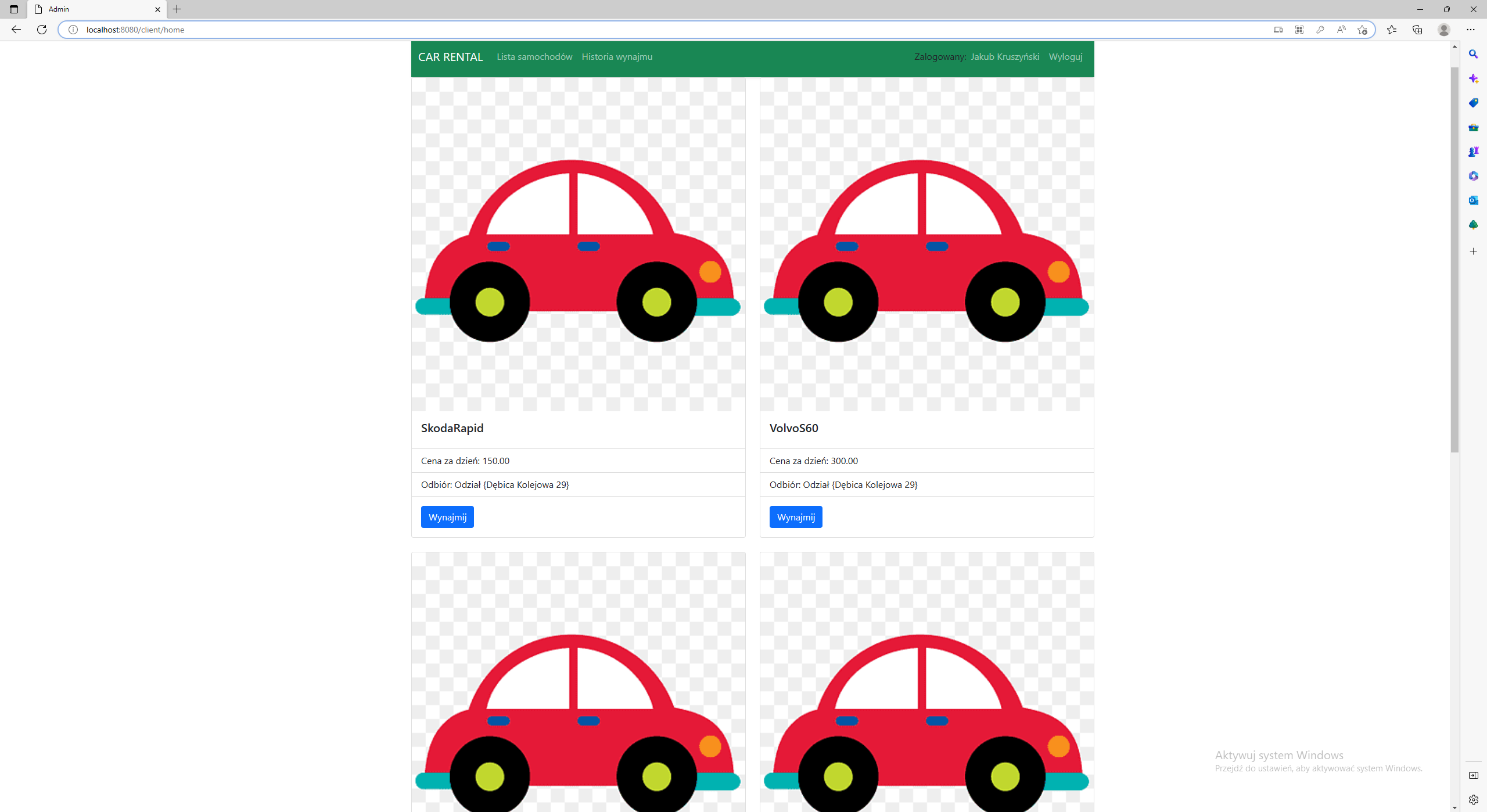


**Rysunek 12 .Menu tworzenia nowego konta**

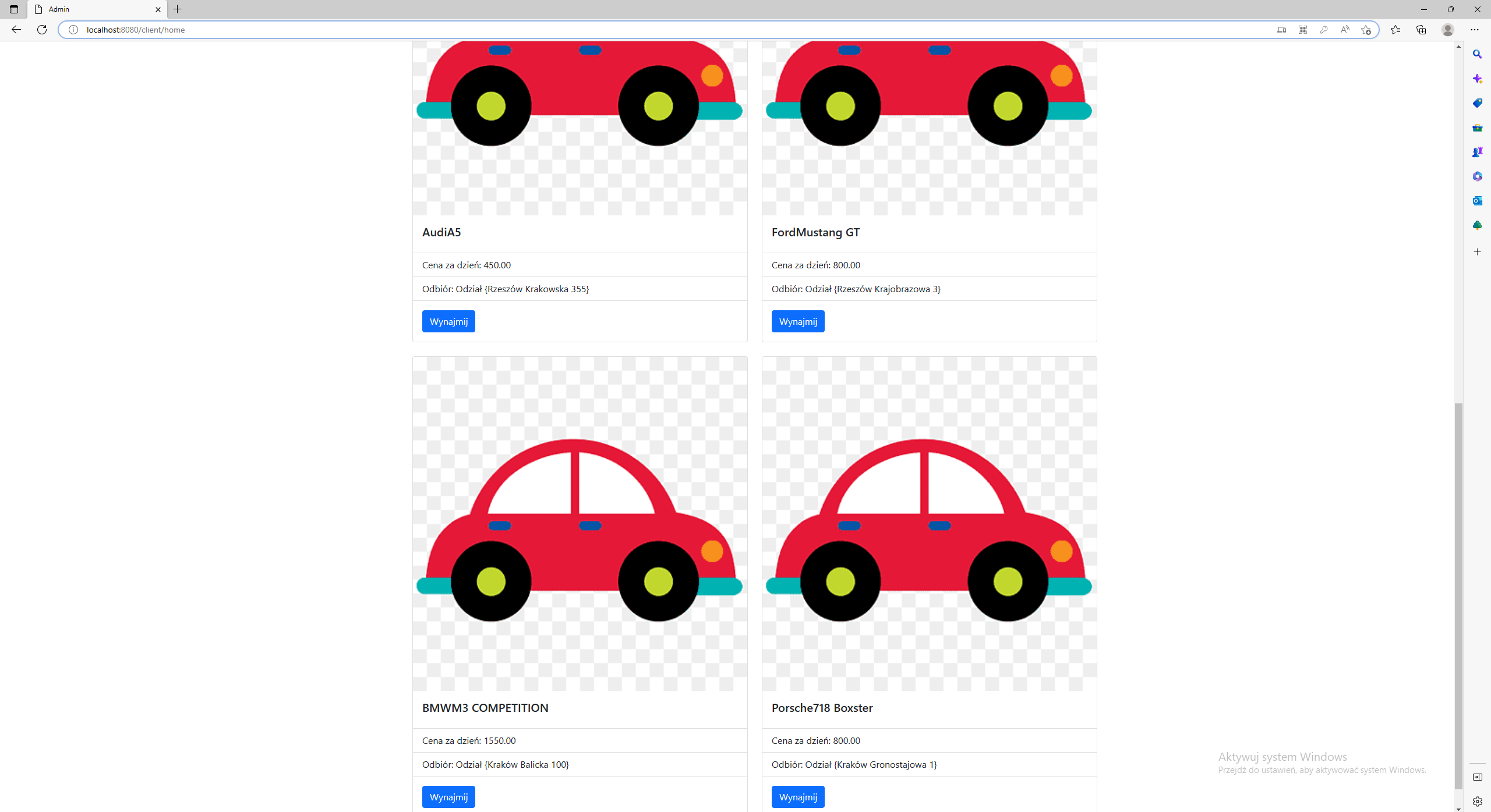


**Rysunek 13 .Przykładowe konto**

Po utworzeniu konta zostajemy przekierowani ponownie do menu logowania, jednak teraz z możliwością zalogowania się i wejścia na stronę wypożyczalni którą utworzył admin. Element ponownego zalogowania pomijam.

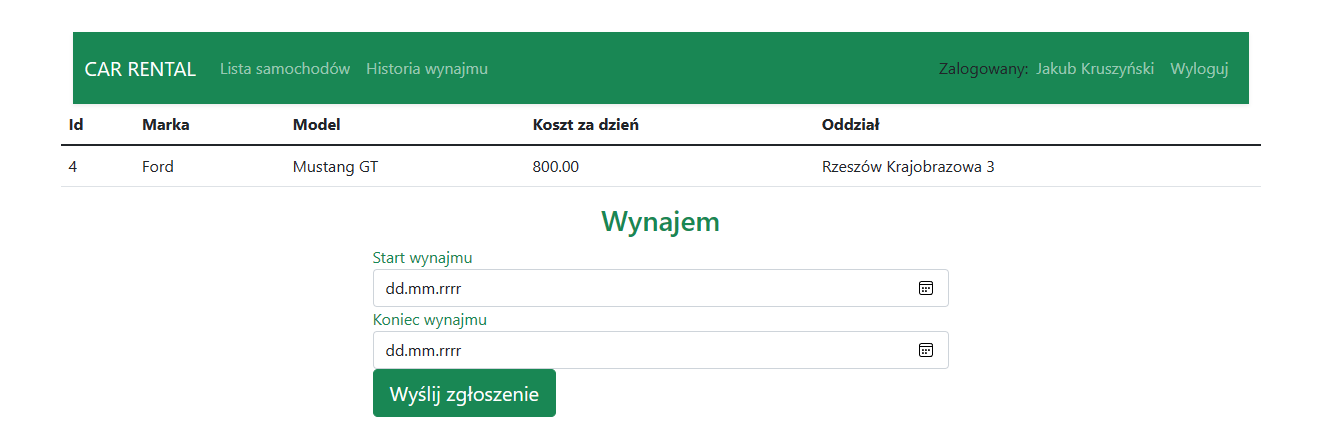


**Rysunek 14 .Menu wypożyczalni ; Lista samochodów\_v1**

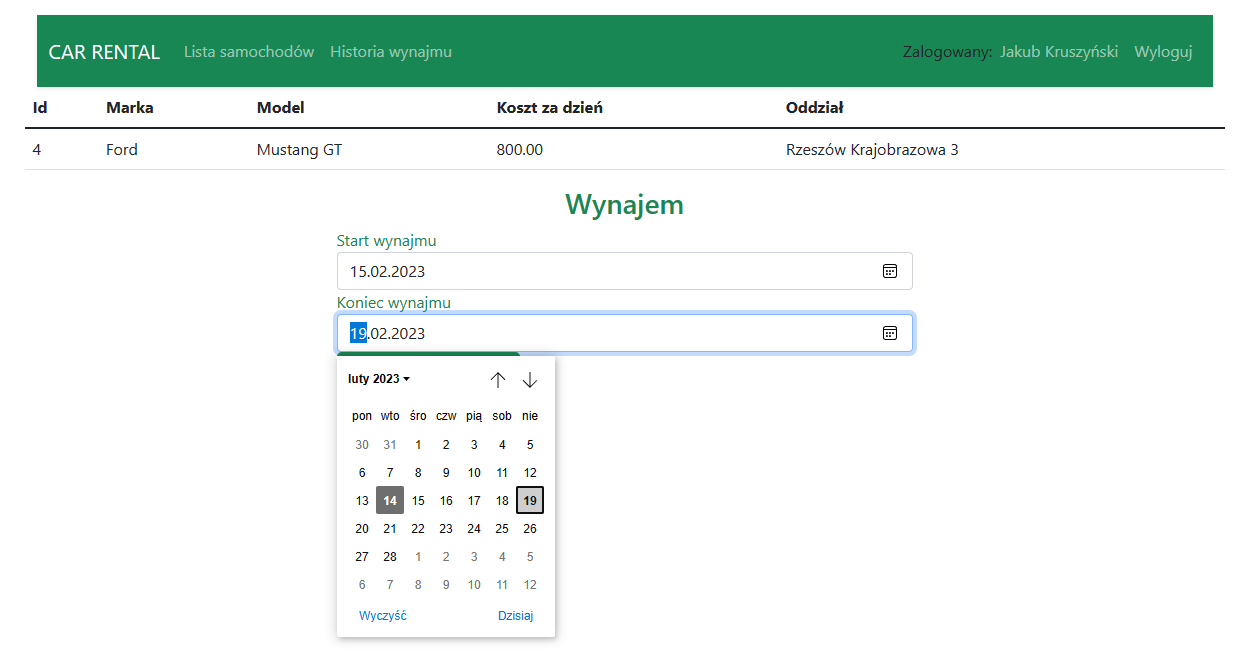


**Rysunek 15 .Menu wypożyczalni ; Lista samochodów\_v2**

Jak widać na zamieszczonych zrzutach ekranu, użytkownik może zarezerwować samochody a więc wypożyczam jeden z nich – mój wybór to Ford Mustang GT



**Rysunek 16 .Menu wynajmowania auta**



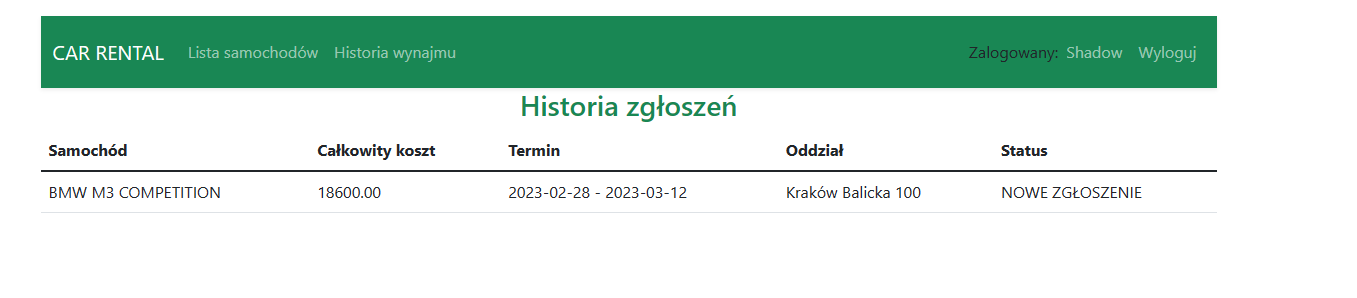
**Rysunek 17 .Przykładowe wynajmowanie samochodu + wygląd kalendarza**

Jak widać na zrzucie ekranu powyżej, użytkownik może wybrać zakres dat lub ręcznie je wprowadzić. Zakres dat musi być prawidłowy bo inaczej wyskoczy ostrzeżenie o błędzie. (**zrzut ekranu powiadomienia o błędzie wraz z innymi zabezpieczeniami umieszczam w punkcie „Zabezpieczenia”** ) Składam kolejne zamówienie na kolejne auto – Volwo S60 i przechodzę do historii wynajmu.



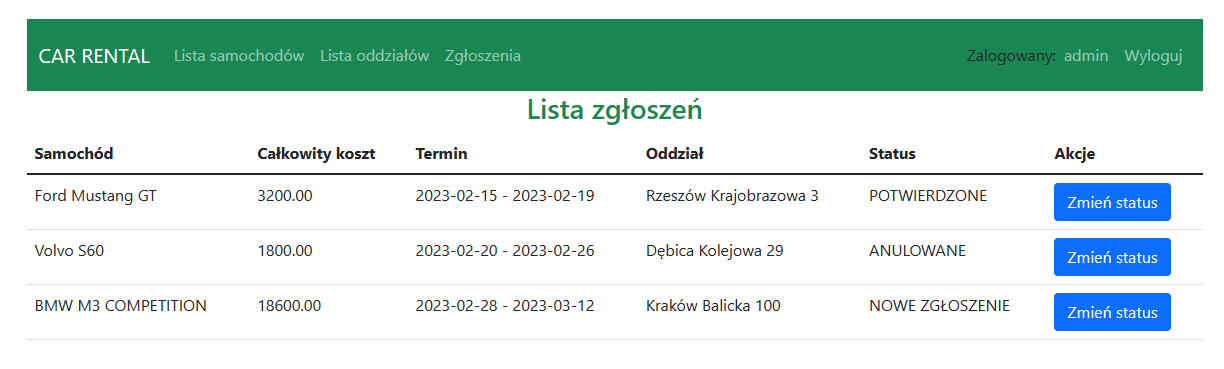
**Rysunek 18 . Historia zgłoszeń pierwszego użytkownika**

Tak wygląda historia zgłoszeń. W celu pokazania dokładnego jej działania utworzę nowe konto i wynajmę kolejne auto.



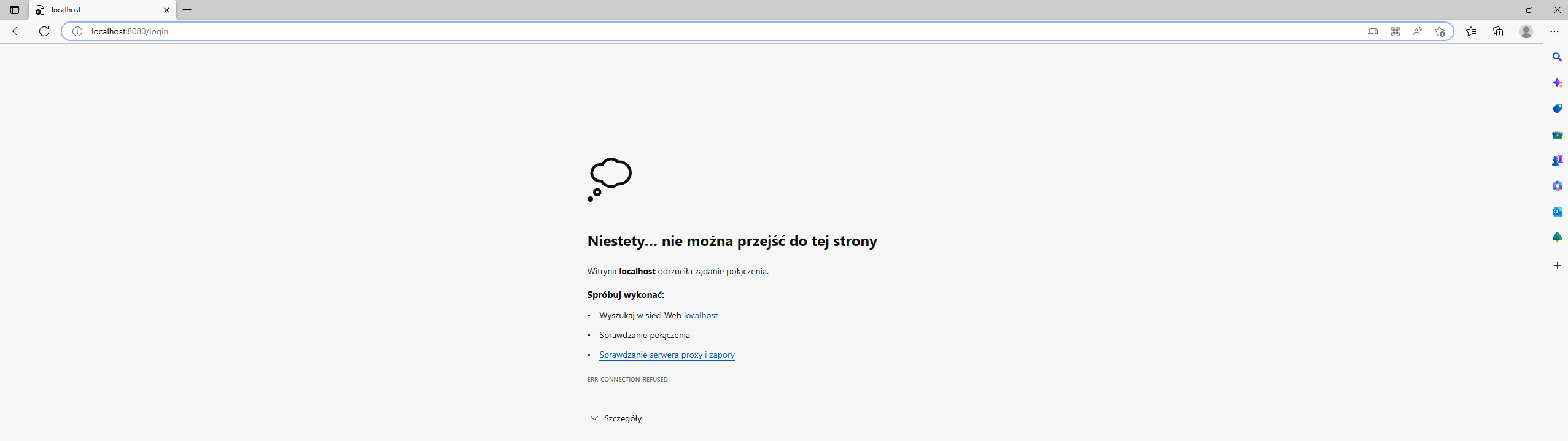
**Rysunek 19 .Historia zgłoszeń drugiego użytkownika**

Jak widać, inny użytkownik może przeglądać jedynie swoje zgłoszena



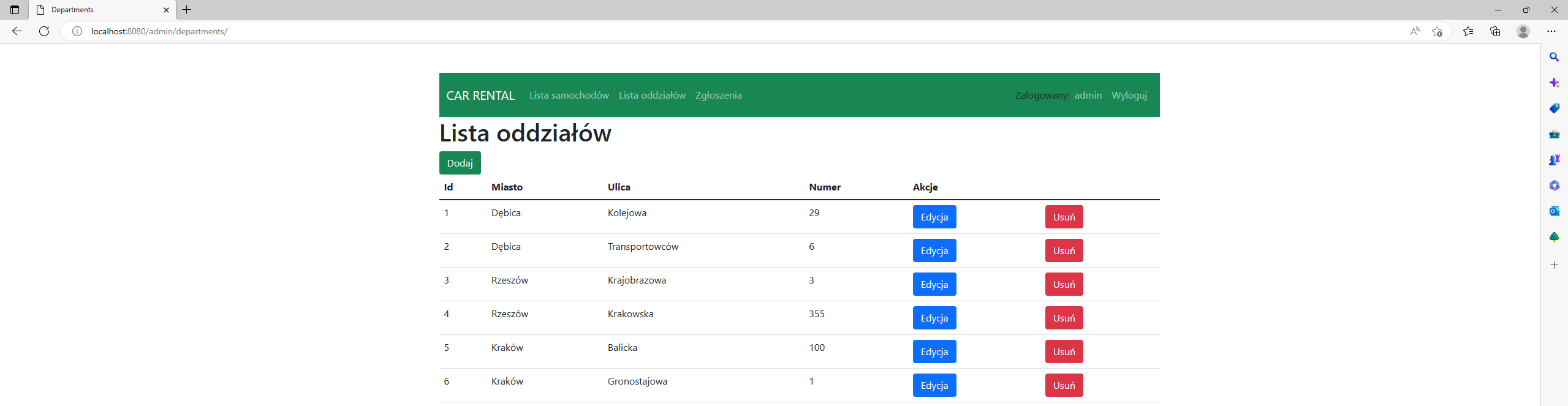
Rysunek 20 .Historia zgłoszeń admina

Admin z kolei widzi wszystkie zgłoszenia i może nadać im jeden z trzech statusów (POWIERDZONE, ANULOWANE, NOWE ZGŁOSZENIE) z czego NOWE ZGŁOSZENIE jest ustawiane domyślnie po pojawieniu się nowego zgłoszenia. Przy Anulowanym celowo nie jest zerowana kwota ponieważ jest to jedynie historia zgłoszeń.



**Rysunek 21 .Strona po wyłączeniu programu**

Po wyłączeniu programu w Javie, nie możemy już dłużej korzystać z wypożyczalni. Teraz zajmę się ponownym uruchomieniem programu w celu sprawdzenia stanu danych na stronie wypożyczalni.



**Rysunek 22 .Strona po ponownym uruchomieniu programu**

Jak widać, pomimo zresetowania programu, program pamięta dane i można nadal zalogować się na każdego z Użytkowników

## 4.4. Zabezpieczenia

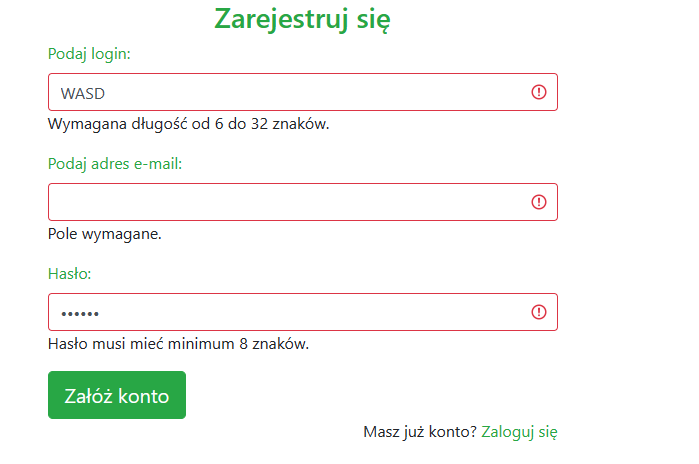
W tej części podam przykłady gdzie zastosowałem walidację danych w celu zabezpieczenia przed wprowadzeniem niepoprawnych danych oraz przed wykonaniem czynności mogących uszkodzić strukturę danych (nie zamieszczam wszystkich zrzutów błędów):

• Podczas zakładania nowego konta, dane Użytkownika mają spełniać następujące warunki (login – od 6 do 32 znaków; e-mail – musi być podany; hasło – minimum 8 znaków)

• Podczas rezerwowania przez Użytkownika danego auta, program sprawdza zakres dat który został zaznaczony porównując do daty aktualnej (data startowa nie może być mniejsza niż dzień aktualny, data końcowa tak samo)

• Program blokuje usuwanie działów, które są w użytku od strony admina (admin może usunąć jedynie te działy do których nie jest przypisane a kiedy spróbuje usunąć dział w użyciu to wyskoczy błąd)

• Program blokuje usuwanie samochodów, które są w użytku od strony admina



# Repozytorium, system kontroli wersji

Projekt został zrealizowany z wykorzystaniem systemu kontroli wersji Git. Dokumentacja oraz projekt został umieszczony w repozytorium dostępnym pod adresem:

<https://github.com/JakubProject/car-rental>

# Podsumowanie

Aplikacja spełnia założone wymagania oraz została wykonana według technicznych zaleceń.

Stworzona aplikacja ma wyraźny, spokojny i przyjemny dla oka interfejs graficzny oraz działa w sposób zadowalający i mogący ułatwić zarządzanie oraz obsługę wypożyczalnią samochodów.

# Literatura

* <https://www.baeldung.com/spring-boot-crud-thymeleaf>
* <https://docs.oracle.com/en/java/>
* <https://www.javatpoint.com/spring-boot-tutorial>
* <https://nullpointerexception.pl/kurs-java-dla-poczatkujacych/>