**Warsztat** **samochodowy**

**dokumentacja projektu**

**Cel projektu:**

Aplikacja bazodanowa z wykorzystaniem bazy danych Oracle 18c

**Opis projektu:**

Obsługa warsztatu samochodowego z użyciem funkcjonalności zaprojektowanych w bazie danych, z interfejsem graficznym stworzonym za pomocą języka Java i biblioteki JavaFX.

**Autor:**

Damian Paluch

**Prowadzący:**

Dr. Inż. Piotr Grochowalski

Spis treści

[1. Założenia. 3](#_Toc40804536)

[2. Schemat ERD 3](#_Toc40804537)

[3. Klasy zaimplementowane w projekcie 3](#_Toc40804538)

[4. Funkcjonalności aplikacji po stronie bazy danych 4](#_Toc40804539)

[5. Wyzwalacze 5](#_Toc40804540)

[6. Opis Interfejsu 5](#_Toc40804541)

[7. Instrukcja użytkowania 13](#_Toc40804542)

[a. Panel Logowania 13](#_Toc40804543)

[b. Panel Klienta 13](#_Toc40804544)

[c. Panel mechanika 13](#_Toc40804545)

[d. Panel administratora 13](#_Toc40804546)

[8. Wymagane loginy i hasła do sprawdzenia działania aplikacji 13](#_Toc40804547)

[a. Baza Danych 13](#_Toc40804548)

[b. Klient 14](#_Toc40804549)

[c. Mechanik 14](#_Toc40804550)

[d. Administrator 14](#_Toc40804551)

[9. Ograniczenia 14](#_Toc40804552)

[10. Lista błędów zdefiniowanych w projekcie: 14](#_Toc40804553)

# Założenia.

Tematem projektu jest aplikacja warsztatu samochodowego. Obsługiwać ją mogą klienci, mechanicy i administratorzy, którzy mają sprawować kontrolę nad działaniem całego systemu. Założeniem aplikacji było współdziałanie trzech paneli dedykowanych dla wymienionych użytkowników w celu obsługi warsztatu.

# Schemat ERD

Schemat znajduje się w pliku ERD.pdf znajdującym się w tym samym katalogu co dokumentacja.

Baza danych zawiera 10 tabel. Łączą je następujące relacje:

Pracownicy\_uzytkownicy – funkcja – relacja jeden do wielu, jedną funkcje może posiadać wielu użytkowników

Pracownicy – pracownicy\_uzytkownicy – relacja jeden do jednego, jeden pracownik może mieć jedno konto

Pracownicy – przeglady – relacja jeden do wielu, jeden pracownik może obsłużyć wiele przeglądów

Pracownicy – samochody\_uslugi – relacja jeden do wielu, jeden pracownik może obsłużyć wiele usług

Pracownicy – diagnozy – relacja jeden do wielu, jeden pracownik może obsłużyć wiele diagnoz

Klienci – klienci\_uzytkownicy – relacja jeden do jednego, jeden klient może mieć jedno konto

Klienci – samochody – relacja jeden do wielu, jeden klient może posiadać wiele samochodów

Samochody – diagnozy – relacja jeden do wielu, jeden samochód może mieć wiele diagnoz

Samochody – przeglądy – relacja jeden do wielu, jeden samochód może mieć wiele przeglądów

Samochody – usługi – relacja wiele do wielu za pomocą tabeli samochody\_uslugi, jeden samochód może mieć wiele usług oraz jedna usługa może być wykonana w wielu samochodach

# Klasy zaimplementowane w projekcie

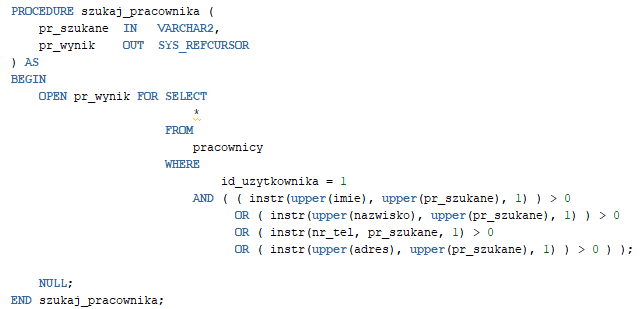
Pakiet „entity” zawiera klasy reprezentujące tabele bazy danych. Znajdują się tam konstruktory, gettery i settery.

Klasy w „top” to w większości pliki .fxml oraz odpowiadające im kontrolery. Oprócz tego znajduje się tu plik „Warsztat.java”, który jest główną klasą aplikacji, „SceneMenager.java” odpowiada za przechodzenie między scenami, „Polaczenie.java” zawiera metody pozwalające uzyskać połączenie z bazą, „darkTheme.css” odpowiada za wygląd aplikacji oraz „2.jpg” to ikona aplikacji.

# Funkcjonalności aplikacji po stronie bazy danych

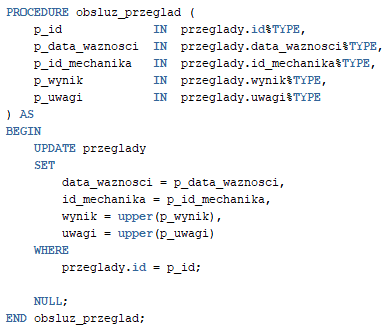
W projekcie zostało wykorzystanych wiele procedur i funkcji języka PL/SQL. Poniżej zostało przedstawionych kilka wraz z opisem.

Ryc. 1 Przykładowe procedury 1.



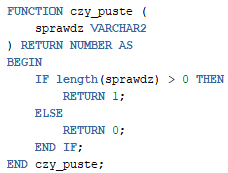
Na Ryc. 1 przedstawiona jest procedura przeszukująca tabele, jeśli znajdzie szukane słowo w którejś kolumnie to zwraca cały wiersz.

Ryc. 2 Przykładowe procedury 2.



Ryc. 2 zawiera procedurę obsługującą przegląd, na który umówił się klient

Ryc. 3 Przykładowe procedury 3.

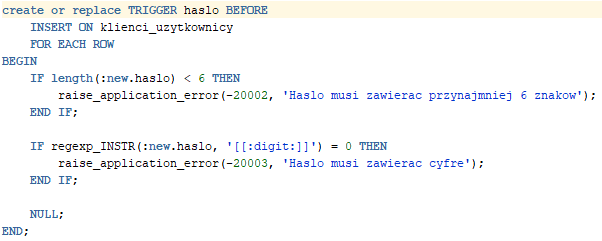


Na ryc. 3 jest funkcja sprawdzająca czy nie podajemy do bazy pustych rekordów. Jeśli taka sytuacja się wydarzy to po wciśnięciu przycisku odpowiedzialnego za dodanie/ edycji danych pola, które zostawiliśmy puste, a są wymagane zostaną podkreślone na czerwono.

# Wyzwalacze

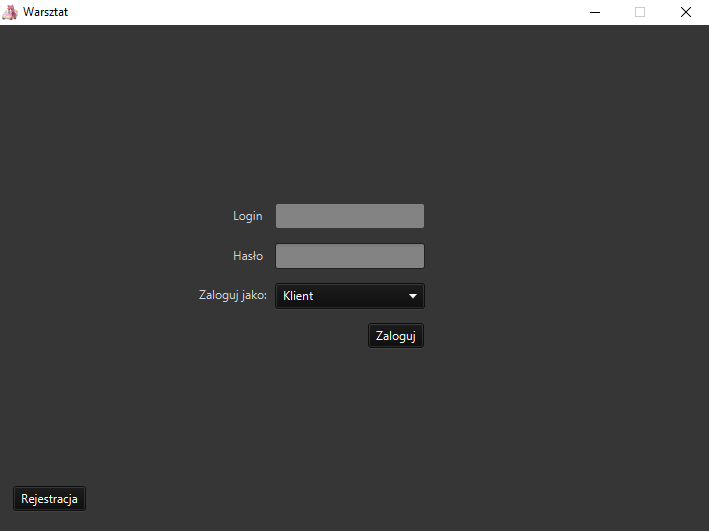
W projekcie jest dziewięć wyzwalaczy w celu walidacji podawanych danych. Przedstawiona poniżej Ryc.4 przedstawia jeden z nich, która sprawdza poprawność podawanego hasła.

Ryc. 4 Przykładowe procedury 4.

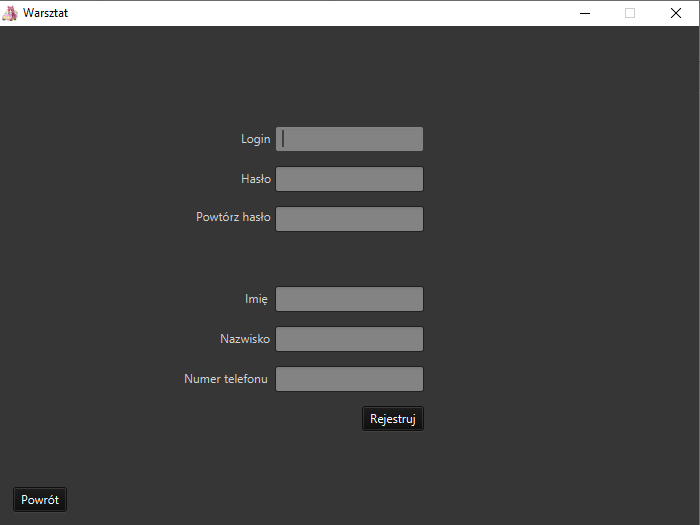


# Opis Interfejsu

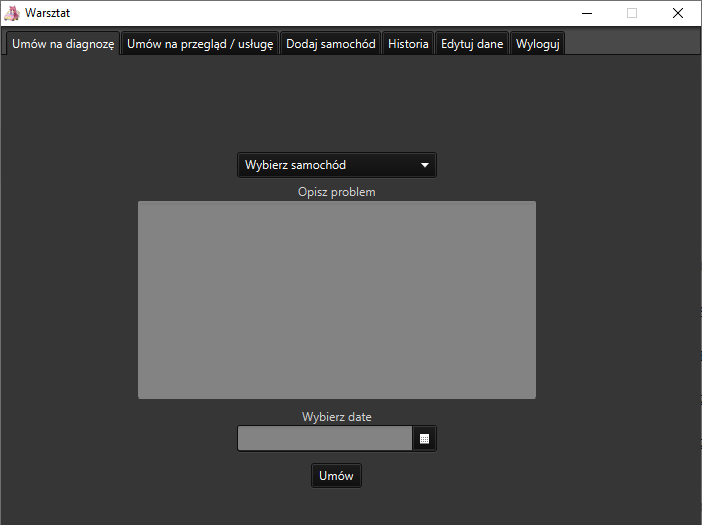
W aplikacji jest pięć głównych paneli: logowanie, rejestracja, klienci, mechanicy i administratorzy.



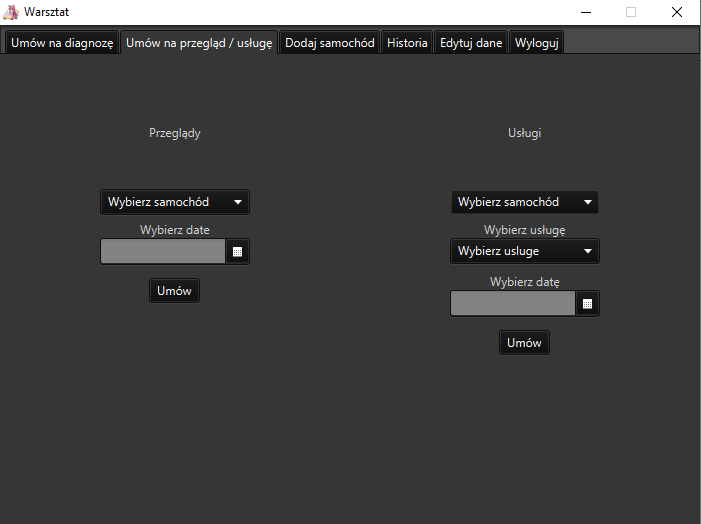
Jest to pierwszy panel, który się wyświetla po uruchomieniu aplikacji, mamy możliwość zalogowania się lub przejścia do panelu rejestracji.



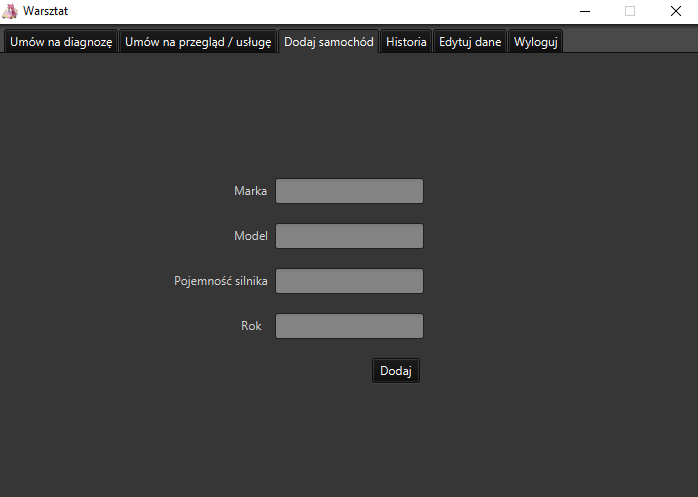
Panel rejestracji, zostały tu nałożone wiele ograniczeń: login nie może się powtarzać, login musi zawierać 6 znaków, hasła muszą być zgodne, hasło musi zawierać cyfrę oraz numer musi zawierać przynajmniej 9 cyfr.



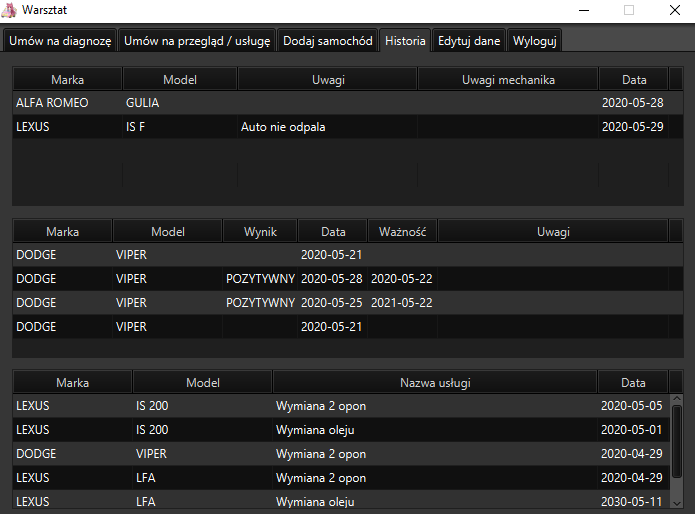
Jest to pierwszy panel po zalogowaniu się jako klient, możemy się tu umówić na przegląd.



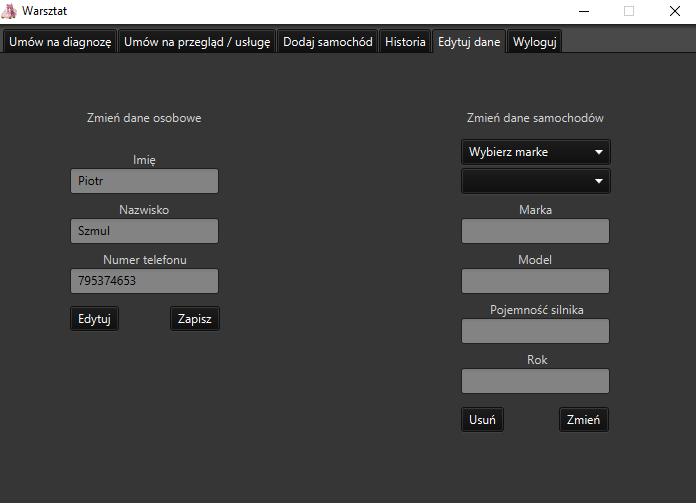
Ten panel jest analogiczny do poprzedniego, jedynie tutaj umawiamy się na przegląd/ usługe.



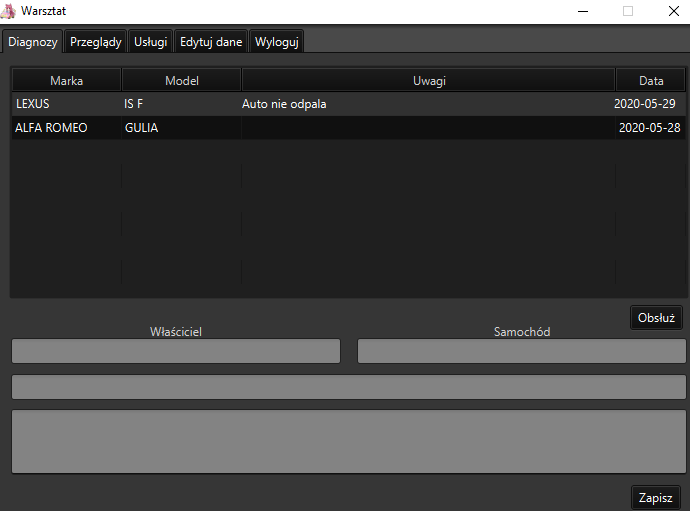
Okno dodawania samochodu, nie można dodać dwóch takich samych.



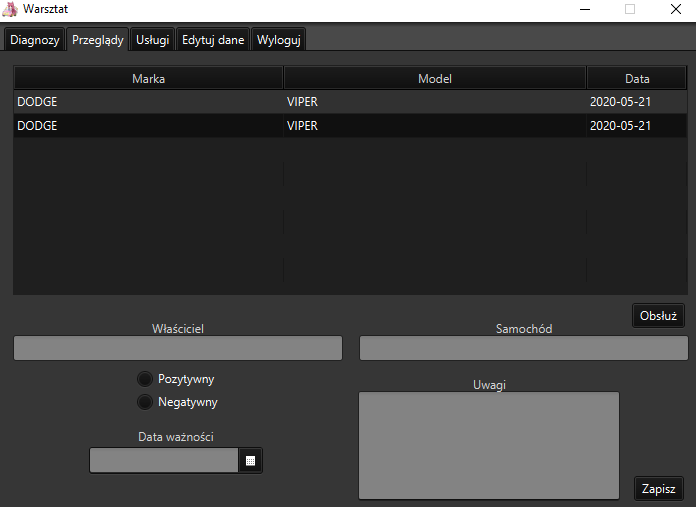
Tutaj jest przedstawiona historia wszystkich usług zalogowanego klienta.

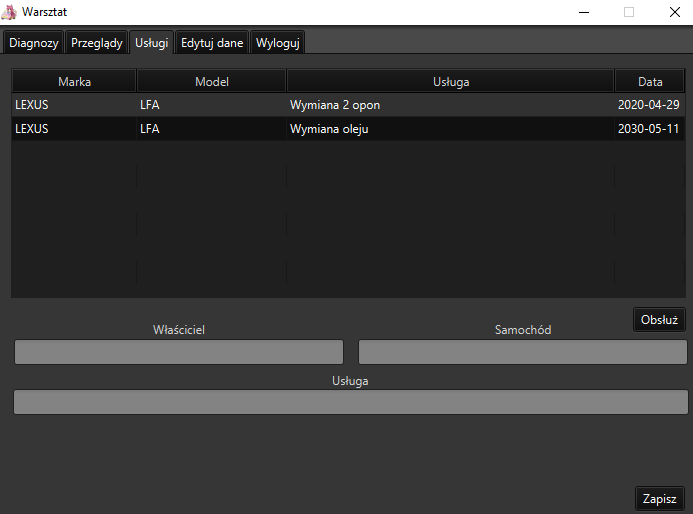


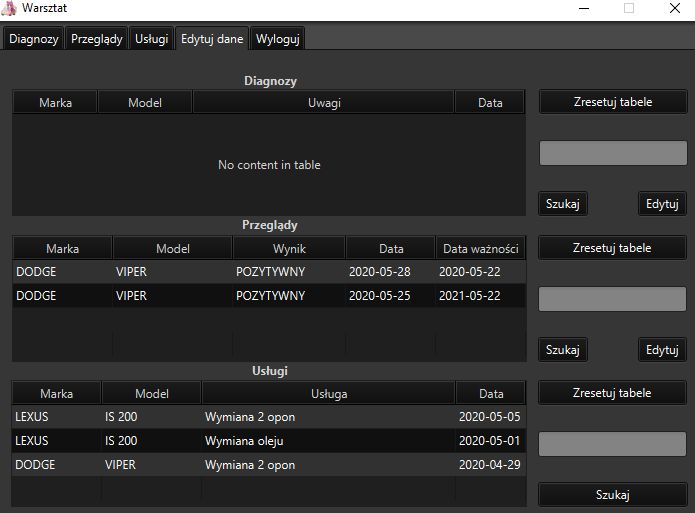
Tutaj mamy możliwość edycji/ usunięcia samochodu i edycji danych personalnych.



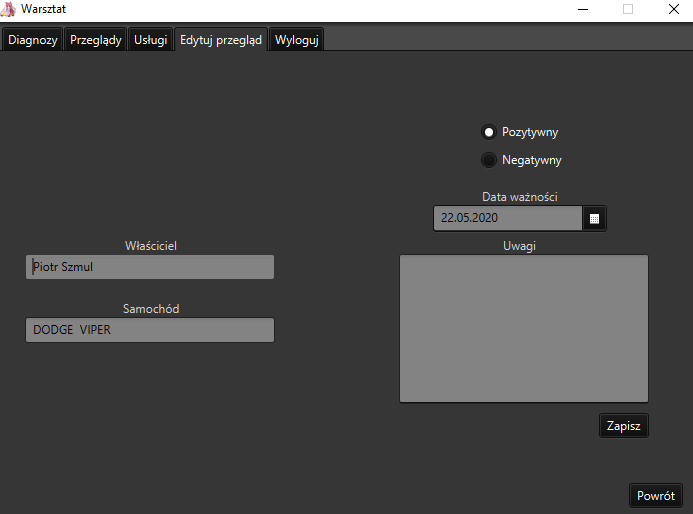
Przechodzimy do panelu mechanika, ten i następne dwa okna odpowiadają za obsługę wszystkich oczekujących wizyt.



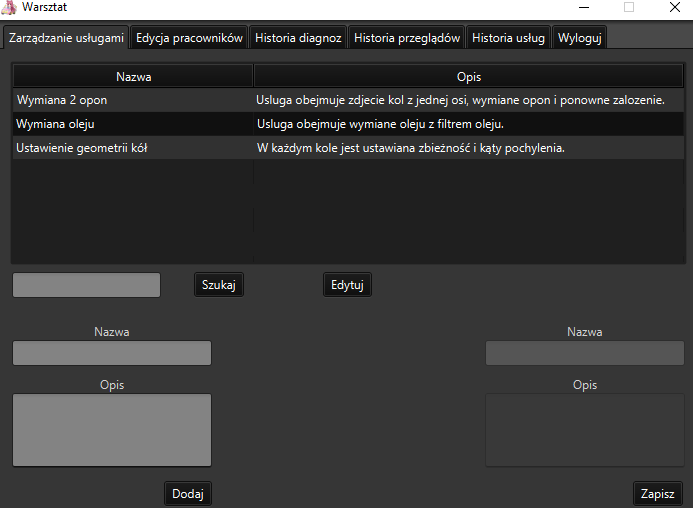




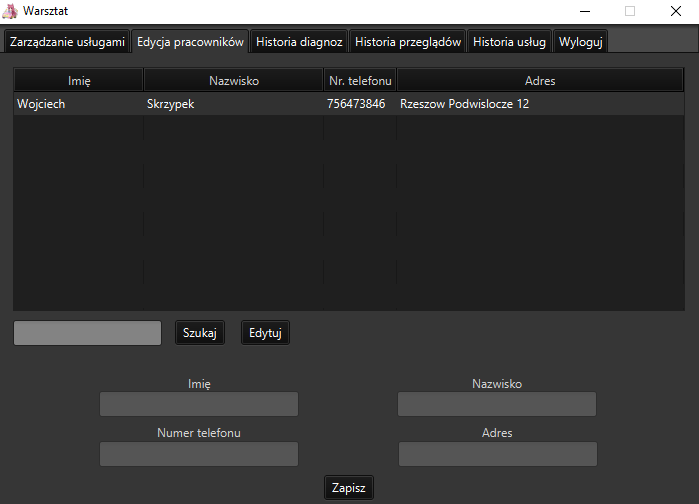
W panelu do edycji danych znajdują się wszystkie obsłużone wizyty przez zalogowanego mechanika, dodatkowo jest tu mechanizm wyszukiwania oraz przyciski do edycji, które wczytują inne sceny.



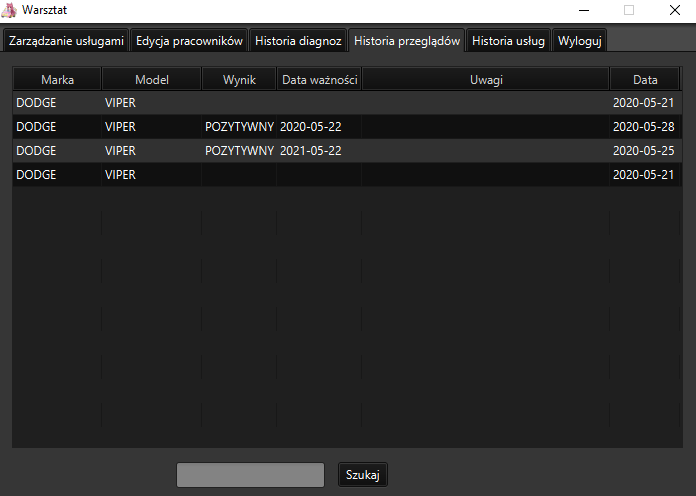
Edycja przeglądu.



Ostatni panel – mechanika. W pierwszej zakładce mam możliwość zarządzania wszystkimi usługami.



Edycja pracowników



Ostatnie trzy wyglądają podobnie, różnią się tylko zawartością tabel. Znajduje się tu historia wszystkich wizyt.

# Instrukcja użytkowania

## Panel Logowania

Należy wprowadzić dane, po czym nacisnąć przycisk zaloguj. W przypadku podania niepoprawnych danych pojawi się odpowiedni komunikat.

## Panel Klienta

* Aby umówić się potrzebujemy dodać samochód, po wprowadzeniu poprawnych danych samochód zostanie dodany i wszystkie pola wymagające auta się uzupełnią.
* Teraz możemy umówić się na diagnozę wybierając samochód, podając swoje uwagi (opcjonalne) i wybierając datę.
* Zapisy na przegląd i usługę wyglądają podobnie, z tą różnicą, że nie podajemy uwag, a do usług dodatkowo wybieramy ją z rozwijanej listy.
* W celu edycji danych osobowych klikamy przycisk edytuj, który odblokowuje pola. Po ich zmianie klikamy zapisz.
* Edycja i usuwanie samochodów polega na wybraniu marki samochodu, wtedy uzupełni się pole model, z którego wybieramy interesujące nas auto, wszystkie pola się wtedy uzupełnią i możemy usunąć lub zmienić dane.

## Panel mechanika

* Pierwsze trzy zakładki polegają na tym samym. Wybieramy interesujący nas rekord w tabeli i klikamy obsłuż. Zostają wtedy uzupełnione informacje podane przez klienta i możemy wtedy go obsłużyć.
* W ostatniej zakładce są wyświetlone wszystkie obsłużone przez nas wizyty w odpowiednich tabelach. Mamy możliwość ich przeszukiwania, które polega na wpisaniu w pole interesującego nas słowa i użyciu przycisku szukaj. Zostanie wtedy odświeżona tabela rekordami zawierającymi wpisany ciąg znaków. Dodatkowo jest możliwość edycji diagnoz i przeglądów. Po wciśnięciu przycisku edytuj przenosi nas do innej sceny.
* W nowej scenie do edycji zostaną wczytane wszystkie informacje o wybranym z tabeli rekordzie. Zmieniamy interesujące nas wartości i klikamy zapisz (pola wpisane przez klienta pozostają nieaktywne).

## Panel administratora

* W pierwszym panelu administratora mamy możliwość wyszukania usług. Po wybraniu jednej i kliknięciu edytuj zostaną wczytane do pól po prawej dane usługi, którą możemy edytować. Pola po lewej służą do dodawania nowych usług.
* Obsługa pracowników wygląda bardzo podobnie z wyjątkiem braku możliwości dodania nowego.
* Ostatnie trzy zakładki przedstawiają historię warsztatu z możliwością szukania.

# Wymagane loginy i hasła do sprawdzenia działania aplikacji

## Baza Danych

* Użytkownik: c##warsztat
* Sid: xe
* Hasło: admin
* Host: localhost
* Port: 1521

Zmiana danych na potrzeby testów możliwa jest w pliku Polaczenie.java

## Klient

* Login: klient
* Hasło: klient1

## Mechanik

* Login: mechanik
* Hasło: mechanik1

## Administrator

* Login: administrator
* Hasło: administrator1

# Ograniczenia

* Jednego dnia można się zapisać na 3 diagnozy, 10 przeglądów i 15 usług,
* Jeden klient nie może dodać dwóch takich samych samochodów,
* Nie ma możliwości zapisania się w datą, która już minęła,
* Podobnie nie da się zapisać w weekend,
* Ograniczenia złożoności loginu i hasła.

# Lista błędów zdefiniowanych w projekcie:

* -20000 - Błędne dane logowania
* -20001 - Login musi zawierać przynajmniej 6 znaków
* -20002 - Hasło musi zawierać przynajmniej 6 znaków
* -20003 - Hasło musi zawierać cyfrę
* -20004 - Numer telefonu musi zawierać 9 cyfr
* -20005 - Hasła się różnią
* -20006 - Data z przeszłości
* -20007 - Nie pracujemy w weekend
* -20008 - Samochód istnieje w bazie