Aplikacja do zarządzania flotą pojazdów



Kacper Mazur

Informatyka i ekonometria

131378

Spis treści

1. Wprowadzenie i ogólne założenia projektu
2. Instalacja
3. Interfejs użytkownika
4. Funkcjonalność
5. Bezpieczeństwo
6. Dokumentacja JavaDoc
7. Przydatne Informacje

**1.Wprowadzenie i ogólne założenia projektu:**

Celem aplikacji jest usprawnienie zarządzania flotą pojazdów w firmach transportowych oraz przedsiębiorstwach posiadających flotę samochodową. Aplikacja umożliwia efektywne zarządzanie danymi dotyczącymi pojazdów, kierowców, przeglądów technicznych, ubezpieczeń oraz informacji o kierowcach. Aplikacja JavaFX połączona z bazą danych umożliwiająca zarządzanie flotą pojazdów w firmie transportowej w celu uproszczenia logistycznego, zabieg ten ma na celu na celu łatwiejsze wyszukiwanie konkretnego pojazdu/pojazdów, kierowców, ubezpieczeń i przeglądów w celu sprawdzenia stanu jego przeglądu technicznego i informacji dodatkowych. Aplikacja została zaprojektowana jako narzędzie dla administratorów floty, którzy mogą za jej pomocą dodawać, usuwać oraz wprowadzać dane w przejrzystym interfejsie graficznym. Kluczowym elementem aplikacji jest jej zdolność do przechowywania informacji w relacyjnej bazie danych, co zapewnia integralność danych oraz możliwość łatwego ich przetwarzania i analizy. Poprzez dostępne w aplikacji pole powiadomień użytkownik na bieżąco informowany jest o nieważnym ubezpieczeniu oraz przeglądzie w celu prostej weryfikacji konieczności odnowienia tych danych. Dodatkowo do aplikacji zaimplementowana zostanie tabela kierowcy, która znacząco otworzy możliwości dalszej rozbudowy projektu.

Wymagania systemowe:

**Urządzenia i platformy** Aplikacja została zaprojektowana do pracy na komputerach stacjonarnych oraz laptopach z zainstalowanym systemem operacyjnym Windows, macOS lub Linux. Została napisana w języku Java, co zapewnia jej przenośność między platformami, pod warunkiem, że urządzenie spełnia wymagania środowiska uruchomieniowego Javy.

**Ograniczenia technologiczne** Technologia Javy wymaga zainstalowanego środowiska Java Runtime Environment (JRE) w wersji 8 lub wyższej. Nie przewiduje się obecnie wsparcia dla urządzeń mobilnych ani przeglądarek internetowych, ponieważ aplikacja została stworzona jako aplikacja desktopowa z wykorzystaniem JavaFX.

**Wymagania sprzętowe i programowe**

* Procesor: Minimum Intel Core i3 lub lepszy.
* Pamięć RAM: Minimum 4 GB.
* Miejsce na dysku: Minimum 500 MB wolnego miejsca.
* System operacyjny: Windows 10/11 lub wyższy, dowolna dystrybucja Linuxa z bibliotekami obsługującymi Javę.
* Zainstalowane oprogramowanie:
  + Java Development Kit (JDK) w wersji 8 lub wyższej.
  + MySQL w wersji 8.0 lub wyższej do obsługi bazy danych.
  + IntelliJ IDEA lub inne IDE do ewentualnej modyfikacji kodu źródłowego.

**Opis tabel**

1. **Tabela: kierowcy**
   * **PESEL** (bigint): Klucz główny jednoznacznie identyfikujący kierowcę.
   * **Imię** (varchar): Imię kierowcy.
   * **Nazwisko** (varchar): Nazwisko kierowcy.
   * **DataUrodzenia** (date): Data urodzenia kierowcy, potrzebna do identyfikacji wieku.
   * **NumerPrawoJazdy** (varchar): Numer prawa jazdy, wymagany do weryfikacji uprawnień.
   * **Uprawnienia** (varchar): Kategorie prawa jazdy posiadane przez kierowcę.

Typy danych wybrano z uwagi na ich przeznaczenie – varchar pozwala na przechowywanie tekstu, a bigint zapewnia odpowiednią przestrzeń dla numeru PESEL.

1. **Tabela: pojazdy**
   * **PojazdID** (int): Klucz główny.
   * **NumerVIN** (varchar): Numer identyfikacyjny pojazdu (unikalny).
   * **Marka** (varchar): Marka pojazdu.
   * **Model** (varchar): Model pojazdu.
   * **RokProdukcji** (int): Rok produkcji pojazdu.
   * **NumerRejestracyjny** (varchar): Numer rejestracyjny pojazdu.
   * **StanTechniczny** (varchar): Opis stanu technicznego.
   * **TypPojazdu** (varchar): Typ pojazdu (np. ciągnik siodłowy, samochód osobowy).

Wybrano varchar dla tekstu, int dla wartości liczbowych, co zapewnia optymalną wydajność.

1. **Tabela: przegladtechniczny**
   * **PrzegladID** (int): Klucz główny.
   * **PojazdID** (int): Klucz obcy wskazujący na tabelę pojazdy.
   * **DataPrzegladu** (date): Data wykonania przeglądu technicznego.
   * **WynikPrzegladu** (varchar): Wynik przeglądu (np. pozytywny, negatywny).
   * **Uwagi** (varchar): Dodatkowe informacje dotyczące przeglądu.
2. **Tabela: przypisania**
   * **PrzypisanieID** (int): Klucz główny.
   * **PojazdID** (int): Klucz obcy wskazujący na tabelę pojazdy.
   * **PESEL** (bigint): Klucz obcy wskazujący na tabelę kierowcy.
   * **DataPrzypisania** (date): Data przypisania pojazdu do kierowcy.
3. **Tabela: ubezpieczenia**
   * **UbezpieczenieID** (int): Klucz główny.
   * **PojazdID** (int): Klucz obcy wskazujący na tabelę pojazdy.
   * **RodzajUbezpieczenia** (varchar): Typ ubezpieczenia (np. OC, AC).
   * **DataRozpoczęcia** (date): Data rozpoczęcia ubezpieczenia.
   * **DataZakończenia** (date): Data zakończenia ubezpieczenia.
   * **Koszt** (decimal): Koszt ubezpieczenia.

Tabele zaprojektowano tak, aby wspierały spójność danych oraz łatwość ich przetwarzania w aplikacji.

1. Tabela pojazdy:

Zastosowanie klucza podstawowego jako PojazdID jest konieczne. Więcej niż jedno pole nadawałoby się aby być kluczem podstawowym, np. NumerVin lub NumerRejestracyjny. Postanowiłem jednak postawić na prostą metodę numerowania pojazdów po kolejności ich dodawania. Zabieg ten jest skuteczny i niezawodny więc komórka ta doskonale sprawdza się w roli PK a zastosowanie auto-inkrementacji pozwala na uniknięcie powtórzenia rekordu jak również konieczności ręcznego wpisywania kolejnego numeru na liście. NumerVin jest unikalny dla każdego pojazdu więc pole zawiera wartość UNIQUE i ograniczony jest do 17 znaków VARCHAR co pozwala na wpisanie pełnej długości bez nadmiarowych znaków. Zarówno UbezpieczenieID jak i PrzeglądID są wartościami UNIQUE ze względu na charakterystykę działania systemu firm ubezpieczeniowych 1:1 i przeglądów 1:1. W danym okresie może być ważne tylko jedno badanie techniczne. Pozostałe tabele posiadają pola zgodne z rzeczywistością godnie odwzorowując prawdziwe przypadki.

1. Tabela kierowcy

W tabeli kierowcy przechowywane są dane osobiste dotyczące pracowników w danej firmie. PESEL jest głównym kluczem podstawowym PK ograniczonym do 11 znaków w celu godnego odwzorowania rzeczywistości. NumerPrawoJazdy również posiada wartość UNIQUE aczkolwiek ciężko jednoznacznie powiedzieć kto jest właścielem danego prawa jazdy. Sama struktura numeru pesel mówi więcej niż tylko ciąg znaków. Brak zastosowania ID uzasadniam swoimi własnymi przemyśleniami aby oszczędzać dane a sama tabela posiada godnego zastępcę dla tego typu rozwiązania. Imię i Nazwisko są podstawowymi informacjami osobistymi o każdym kierowcy. DataUrodzenia posiada maskę wprowadzania co pozwala bezbłędnie wprowadzić dany rekord do tabeli (YYYY-MM-DD).

1. Tabela przegladtechniczny

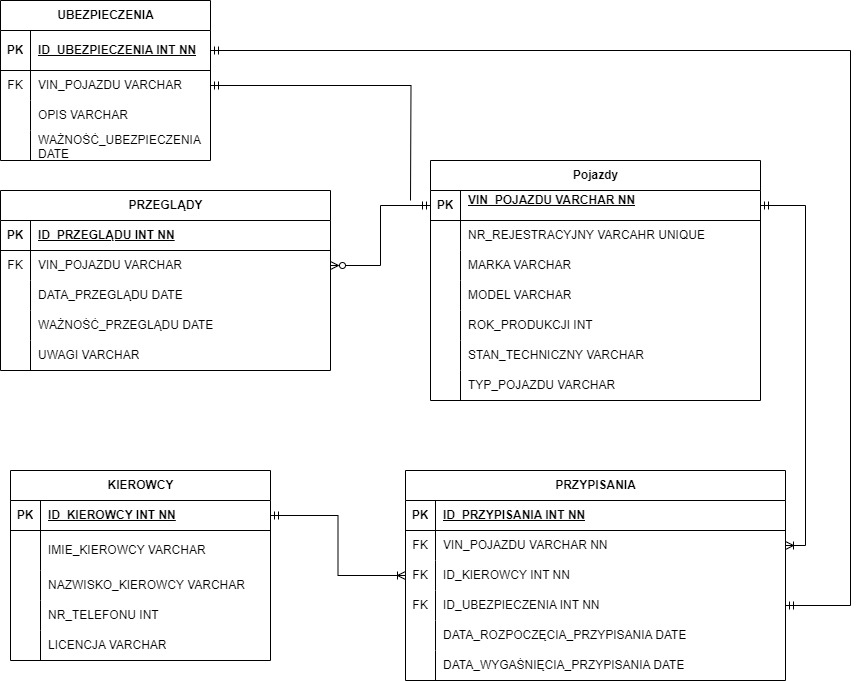
Charakterystyczną cechą tabeli jest zastosowanie jednej daty(DataPrzeglądu) będącej informacją o ostatnim przeglądzie danego pojazdu. Wyliczanie daty wygaśnięcia przeglądu obliczane jest na podstawie bieżącej daty urządzenia (YYYY-MM-DD), natomiast zastosowanie maski wprowadzania pozwala uchronić nas przed wpisaniem złego formatu daty. Kluczem głównym w tabeli jest PrzegladID reprezentuje on numer danego przeglądu. Każdy przegląd musi być powiązany z konkretnym pojazdem więc występuje tu pole PojazdID służące jako łącznik w zapytaniu INNER JOIN wyświetlającym dane odnośnie pojazdu na danym przeglądzie. Dodatkowo pole Uwagi zawiera informacje dotyczące wykonanych lub/i następnych napraw, wymian, przeglądów. WynikPrzegladu informuje użytkownika jak przebiegł dany przegląd i czy pojazd nadaje się do użytku.

1. Ubezpieczenia

Tabela „Ubezpieczenia” została zaprojektowana z myślą o przechowywaniu informacji dotyczących ubezpieczeń przypisanych do pojazdów. Kluczowym elementem jest kolumna „UbezpieczenieID”, pełniąca rolę identyfikatora każdego ubezpieczenia w systemie. Pole to stanowi klucz główny PK i identyfikuje konkretne ubezpieczenie. Kolumna „PojazdID” odwołuje się do tabeli „Pojazdy”, łącząc dane o ubezpieczeniu z konkretnym pojazdem. Dzięki temu powiązaniu, system może precyzyjnie wskazać, które ubezpieczenie dotyczy jakiego pojazdu. Z kolei „RodzajUbezpieczenia” określa typ polisy, (OC lub AC) co pozwala na kategoryzowanie ubezpieczeń i umożliwia późniejsze analizy różnych rodzajów ochrony w przypadku kolizji.

Ważnym aspektem tej tabeli są również daty – „DataRozpoczęcia” i „DataZakończenia” – które wskazują na początek i koniec okresu ochrony ubezpieczeniowej. Posiadają maski wprowadzania(YYYY-MM-DD) co zapobiega błędnemu wprowadzeniu danych. Dzięki nim można dokładnie śledzić, kiedy zaczyna i kończy się obowiązywanie polisy, co ma kluczowe znaczenie przy monitorowaniu terminów płatności i przedłużania ubezpieczeń co w razie wygaśnięcia skutkować może wysokimi karami. Na koniec „Koszt” to pole, które przechowuje wartość, jaka została opłacona na dane ubezpieczenie. Ta informacja jest niezbędna do zarządzania finansami, ponieważ pozwala na kontrolowanie wydatków związanych z ubezpieczeniem pojazdów i w dalszej perspektywie pozwolić może na lepszą analizę wydatków w firmie

**Graficzne przedstawienie relacji i tabel projektu.**



**2. Instalacja**

**1. Instrukcja instalacji**

**Niezbędne oprogramowanie**

1. **Java Development Kit (JDK)**
   * Minimalna wersja: 8 lub nowsza
2. **XAMPP Controll Panel lub MySQL Server**
   * Minimalna wersja: 5.7 lub nowsza
3. **IDE (zintegrowane środowisko programistyczne)**
   * Przykład: IntelliJ IDEA (Community lub Ultimate) lub Eclipse IDE
4. **Git (opcjonalne)**
   * Do zarządzania wersjami kodu.

**Kroki instalacji**

1. Pobierz i zainstaluj **JDK**. Upewnij się, że zmienna środowiskowa JAVA\_HOME wskazuje na katalog instalacyjny JDK.
2. Zainstaluj **XAMPP Control Panel** lub inne środowisko do zarządzania bazami danych MySql.
3. Włącz plik znajdujący się w katalogu: Aplikacja\_Zarzadzanie\_flota\_pojazdow\out\artifacts\Aplikacja\_Zarzadzanie\_flota\_pojazdow\_jar. Włączenie tego pliku uruchomi procedurę startu i automatycznie zaimportuje bazę danych jeśli nie zostanie wykryta.

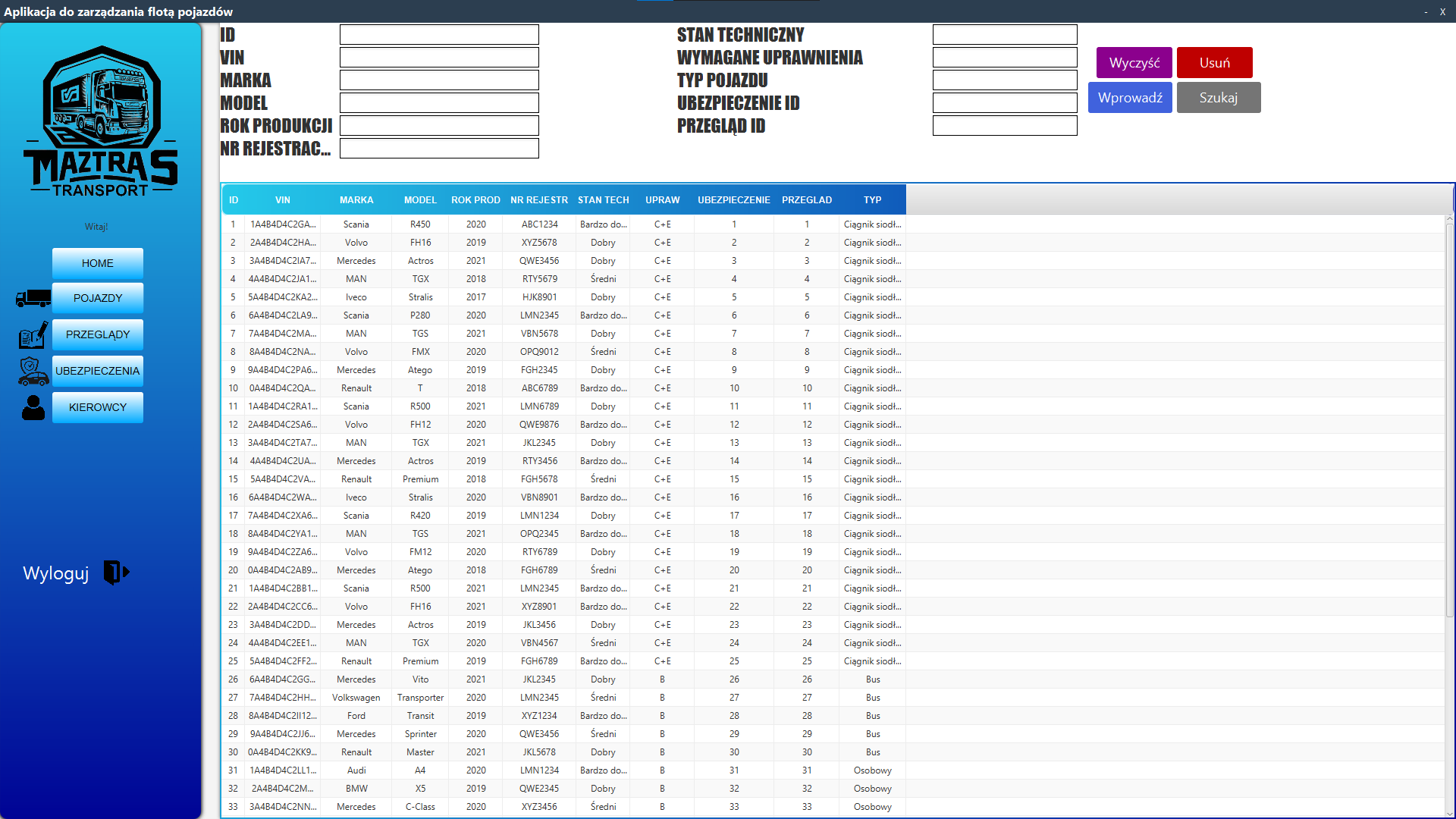
**3.Interfejs Użytkownika**

Wygląd graficzny i podział na okna:

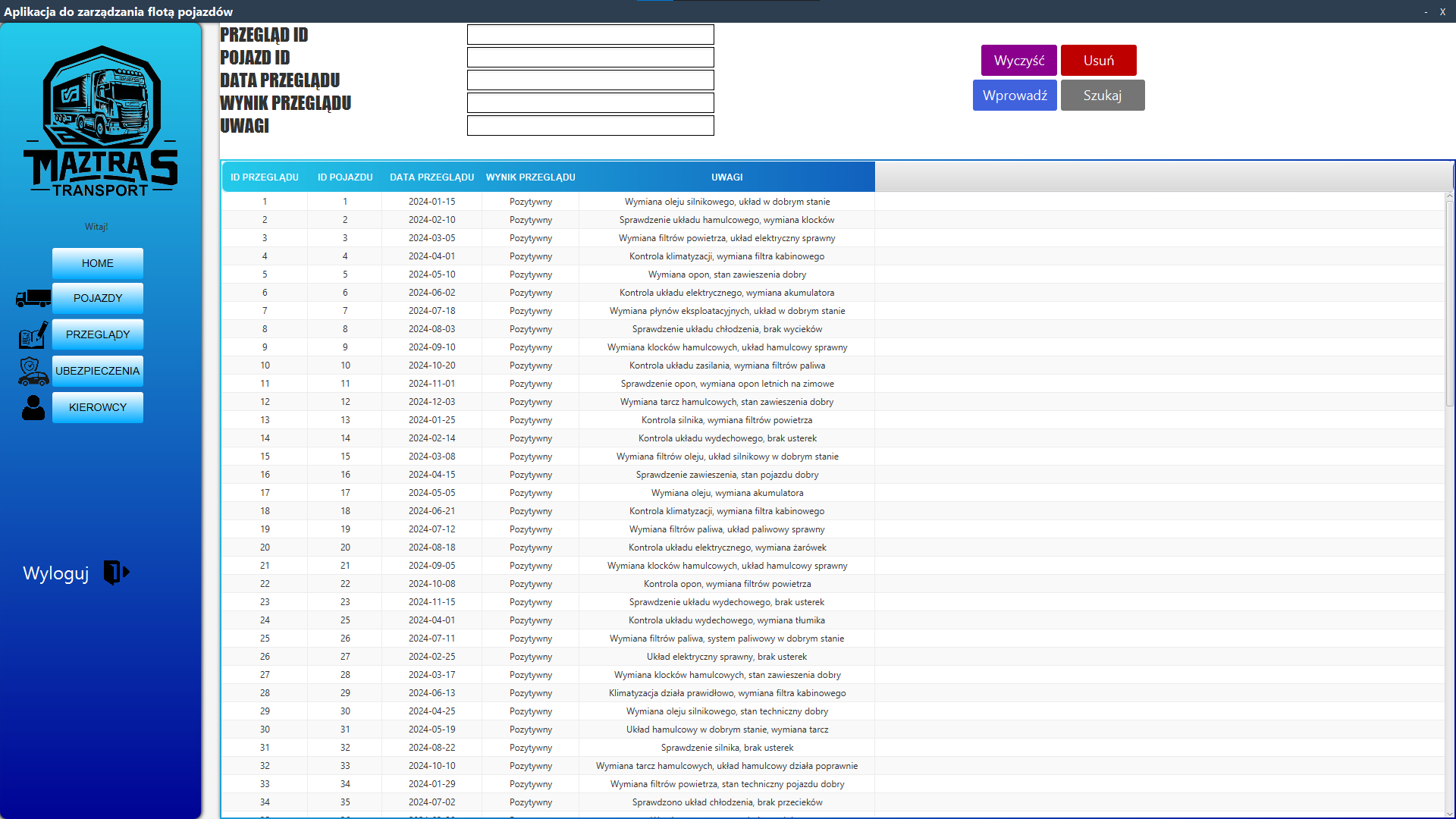
1. Logowanie - początkowe, startowe okno weryfikujące dane wpisane przez użytkownika



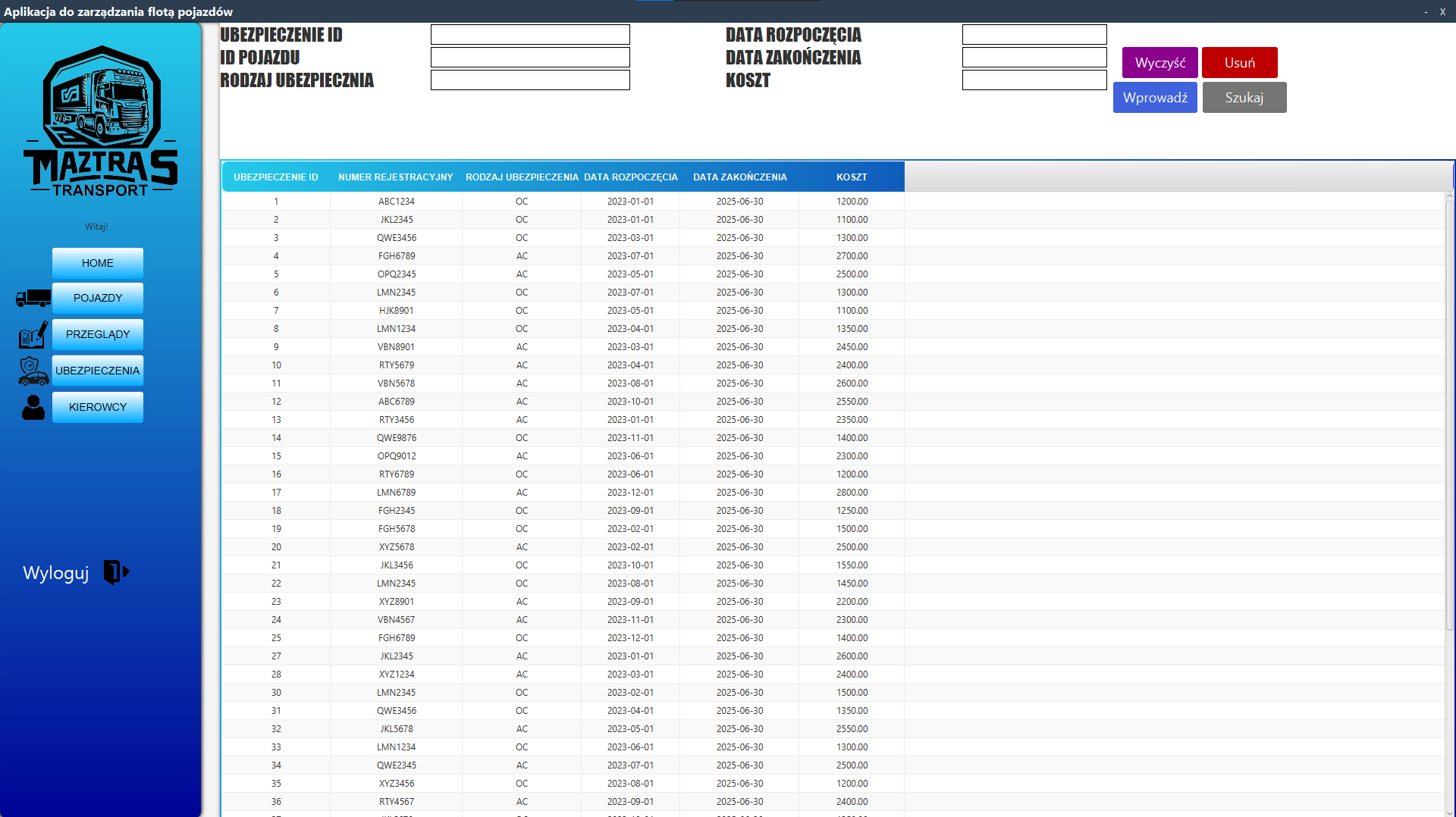
1. Pojazdy - zakładka pozwalająca sprawdzić informacje danym pojeździe bądź grupie pojazdów



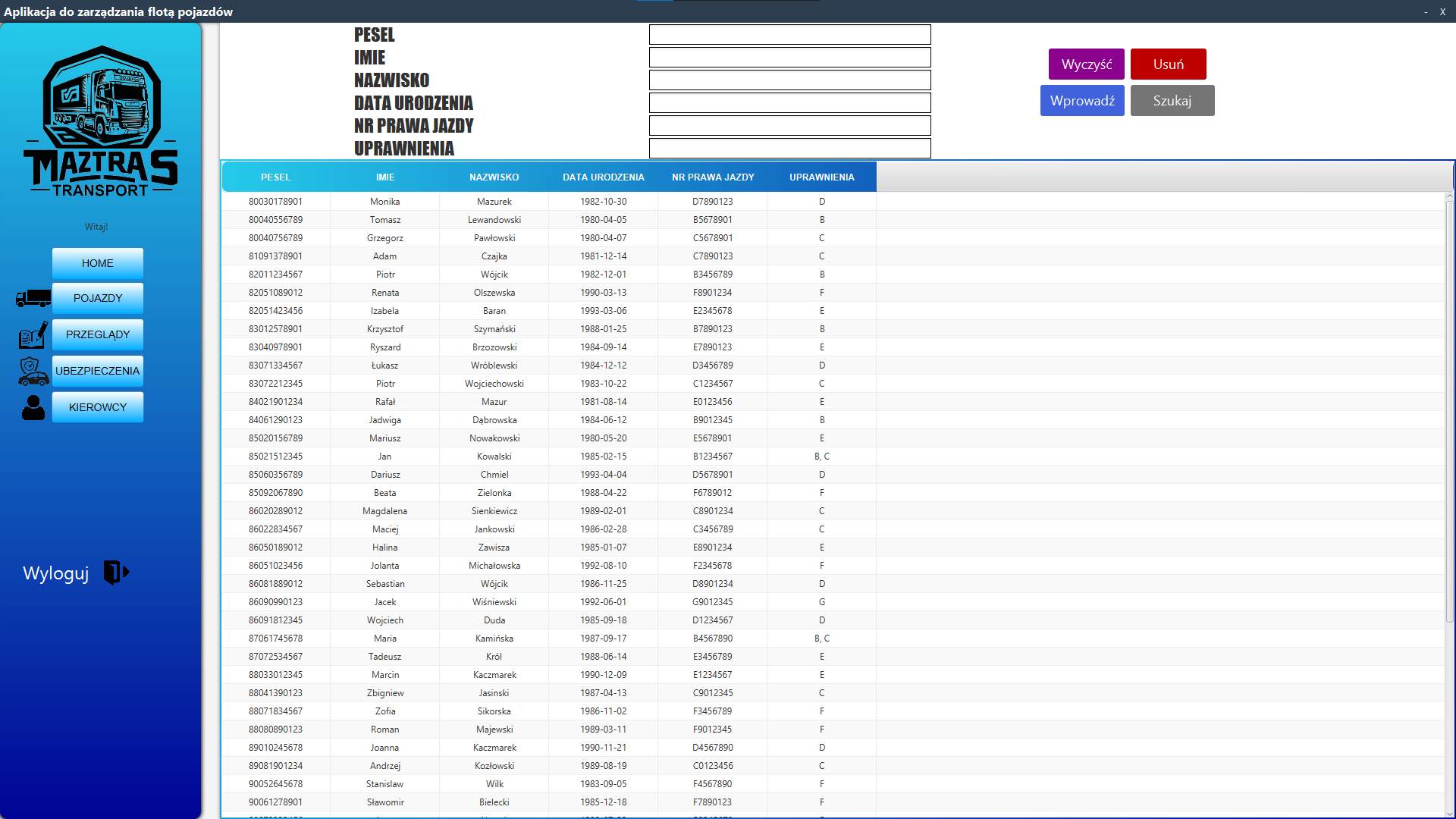
1. Przegląd - zakładka przedstawiać będzie dogłębne informacje o przeglądzie i ubezpieczeniu konkretnego pojazdu bądź grupie pojazdów.



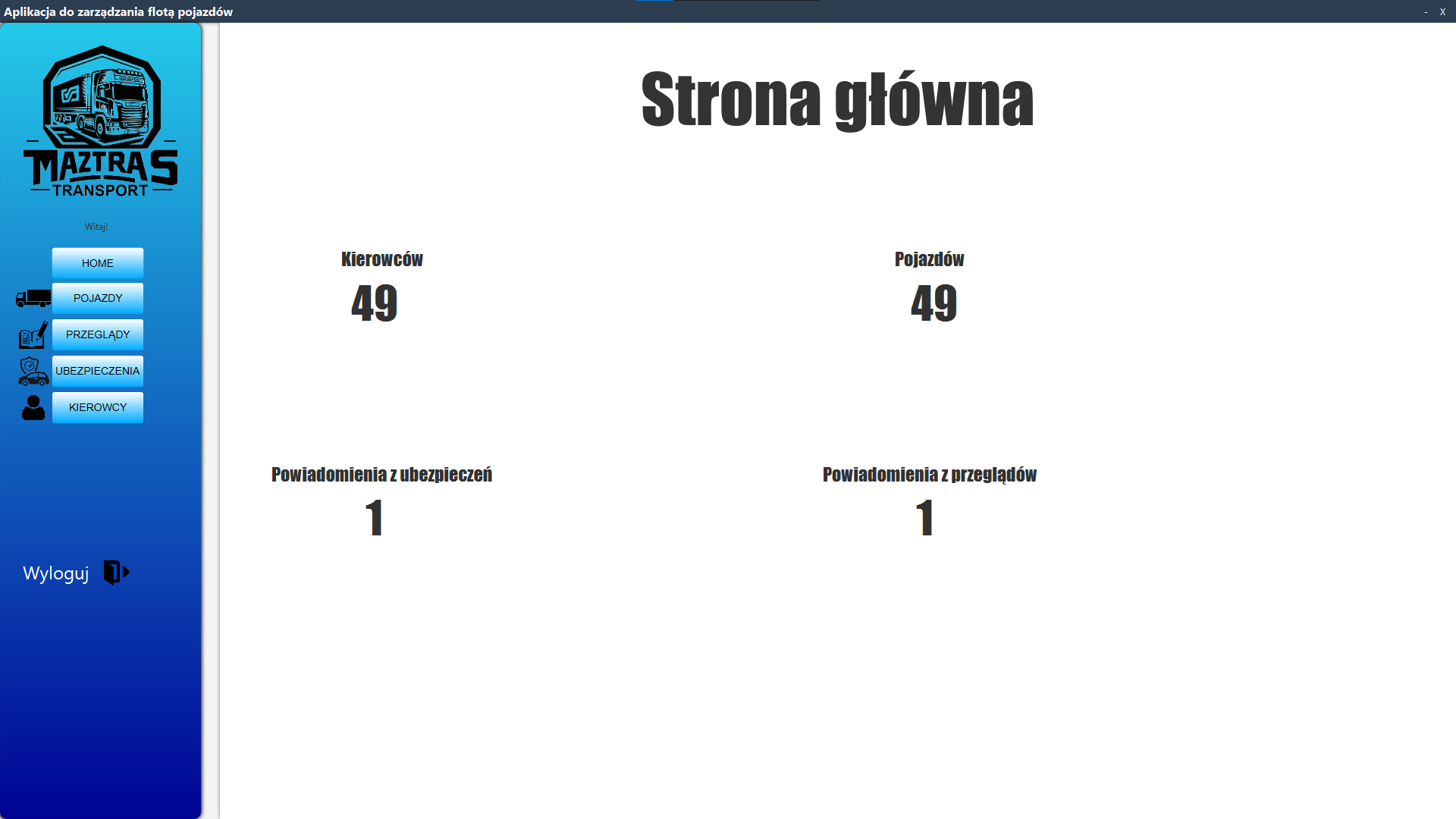
1. Ubezpieczenia- wyświetlenie konkretnych informacji zwrotnych dla użytkownika dotyczących ubezpieczeń.



1. Kierowcy – zakładka przedstawia kierowców i informacje o nich.



1. Powiadomienia – zakładka pozwalająca w prosty sposób określić liczbę zatrudnionych kierowców, bieżącą liczbę pojazdów oraz zasygnalizować nieważne przeglądy i ubezpieczenia w celu weryfikacji i reakcji przez administratora w celu odesłania konkretnego pojazdu na wymagany zabieg kontroli.



**4.Funkcjonalności:**

**a. Lista funkcjonalności**

1. **Zarządzanie pojazdami:**
   * Dodawanie nowych pojazdów do bazy danych.
   * Edycja danych pojazdów, takich jak marka, model, numer rejestracyjny.
   * Przeglądanie historii pojazdu (ubezpieczenia, przeglądy, przypisanie do kierowcy).
2. **Zarządzanie dokumentami:**
   * Dodawanie nowych ubezpieczeń i przeglądów technicznych.
   * Weryfikacja ważności ubezpieczenia i przeglądów.
   * Powiadomienia o zbliżających się terminach ważności.
3. **Zarządzanie kierowcami:**
   * Dodawanie i edycja danych kierowców.
   * Przypisywanie kierowców do pojazdów.

**b. Opis krok po kroku, jak wykonać daną czynność**

**Zarządzenie dodawaniem (na przykładzie dodania nowego pojazdu):**

1. Użytkownik loguje się do systemu jako administrator.
2. W panelu głównym wybiera sekcję „Pojazdy”.
3. W formularzu wprowadza szczegóły pojazdu:
   * VIN
   * Marka
   * Model
   * Rok produkcji
   * Numer rejestracyjny
   * Stan techniczny
   * Wymagane uprawnienia
   * ID ubezpieczenia
   * ID przeglądu
   * Typ pojazdu
4. Kliknięcie „Wprowadź” powoduje aktywowanie procedury sprawdzania danych.
5. System potwierdza dodanie pojazdu i wyświetla nowy wpis na liście pojazdów.

**Zarządzanie ubezpieczeniem pojazdu (dodanie ubezpieczenia):**

1. Użytkownik loguje się do systemu jako administrator.
2. W panelu głównym wybiera sekcję „Ubezpieczenia”.
3. W formularzu wprowadza szczegóły ubezpieczenia:
   * ID ubezpieczenia
   * ID pojazdu
   * Rodzaj ubezpieczenia
   * Data rozpoczęcia.
   * Data zakończenia
   * Koszt.
4. Klika „Wprowadź”.
5. System weryfikuje dane i zapisuje ubezpieczenie.
6. Powiadomienie o dodaniu ubezpieczenia jest wysyłane do użytkownika.

**c. Diagramy przepływu informacji**

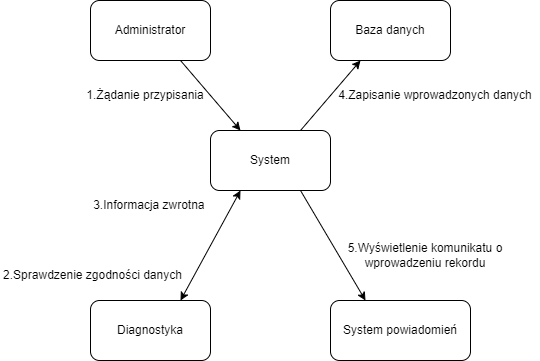
**Przykład: Przypisywanie przeglądu do pojazdu:**

* Administrator przypisuje dane przeglądu poprzez panel zarządzania.
* System weryfikuje, czy użytkownik wprowadza dla przeglądu odpowiednie atrybuty w poszczególne kolumny (np. Unikalny Klucz Podstawowy).
* Po zatwierdzeniu dane są zapisywane w bazie.

**Przykładowy diagram przepływu:**

1. Użytkownik (Administrator) -> System -> Baza danych: Dodanie przeglądu.
2. System -> Diagnostyka: Sprawdzenie wprowadzonych danych.
3. System -> Powiadomienie: Potwierdzenie przypisania.

Analogicznie dla pozostałych danych.



**5.Bezpieczeństwo**

1. Zabezpieczenie przed dostaniem się osoby trzeciej do bazy danych działające na zasadzie wprowadzenia poprawności loginu i hasła do aplikacji.
2. Zabezpieczenia SQL:
3. Tabela pojazdy:

* PojazdID – PrimaryKey,NN,auto\_increment
* NumerVIN-UNIQUE
* UbezpieczenieID-UNIQUE
* PrzeglądID-MUL

1. Tabela kierowcy:

* PESEL- PrimaryKey,NN
* NumerPrawoJazdy-UNIQUE

1. Tabela ubezpieczenia:

* UbezpieczenieID- PrimaryKey,NN
* PojazdID-UNIQUE

1. Tabela przegladtechniczny:

* PrzegladID- PrimaryKey,NN
* PojazdID-MUL

W celu zabezpieczenia spójności danych niemożliwe jest dodanie przeglądu i ubezpieczenia do nieistniejącego pojazdu, zatem kolejność dodawania nowego pojazdu do bazy danych wygląda następująco:

Pojazd->Ubezpieczenie->Przegląd

W przeciwnym razie dodanie nowego rekordu okaże się niemożliwe!

Cała struktura tej tabeli ma na celu zapewnienie pełnej informacji o ubezpieczeniach pojazdów, umożliwiając ich efektywne zarządzanie, kontrolowanie terminów oraz analizowanie kosztów związanych z utrzymaniem polis ubezpieczeniowych.

1. **Dokumentacja dostępna w JavaDoc**

**7.Przydatne informacje**

**Ograniczenia aplikacji**

1. **Brak obsługi wielu użytkowników**:

* Aplikacja obsługuje jedynie jednego użytkownika (administratora), co może być problematyczne w środowisku wieloosobowym.

1. **Brak zaawansowanego systemu uprawnień**:

* Każdy użytkownik (lub administrator) ma pełny dostęp do wszystkich danych w aplikacji. Brakuje mechanizmów kontrolowania dostępu (np. uprawnienia dla różnych ról).

1. **Brak funkcji eksportu danych**:
   * Aplikacja nie oferuje możliwości eksportu danych (np. do plików CSV, PDF), co ogranicza możliwość współpracy z innymi systemami.
2. **Brak funkcji raportowania**:
   * Aplikacja nie posiada funkcji generowania raportów lub analiz (np. roczny raport przeglądów czy wygasłych ubezpieczeń).
3. **Brak możliwości pracy w trybie online**:
   * Aplikacja wydaje się działać w środowisku lokalnym, co ogranicza jej zastosowanie w większych organizacjach wymagających zdalnego dostępu.

**Zalety aplikacji**

1. **Przejrzysty interfejs użytkownika**:

Dashboard umożliwia łatwe zarządzanie danymi pojazdów, kierowców, ubezpieczeń i przeglądów w jednym miejscu.

1. **Prosta obsługa danych**:
   * Aplikacja oferuje intuicyjne funkcje dodawania, edycji, wyszukiwania i usuwania rekordów, co ułatwia zarządzanie bazą danych.
2. **Integracja z bazą danych**:
   * Pełne połączenie z bazą danych umożliwia przechowywanie dużej ilości danych i ich zarządzanie w sposób strukturalny.
3. **Automatyczna inicjalizacja**:
   * Po uruchomieniu aplikacja automatycznie ładuje dane do odpowiednich tabel i aktualizuje statystyki (np. liczba pojazdów, wygasłe ubezpieczenia).
4. **Funkcje wyszukiwania**:
   * Użytkownik może wyszukiwać dane w każdej sekcji (pojazdy, kierowcy, ubezpieczenia, przeglądy) na podstawie wprowadzonych kryteriów.
5. **Walidacja podstawowa**:
   * Aplikacja sprawdza, czy wszystkie wymagane pola zostały wypełnione, co zmniejsza ryzyko wprowadzenia niekompletnych danych.
6. **Obsługa błędów podczas operacji na bazie danych**:
   * W przypadku błędów (np. brak połączenia z bazą), aplikacja wyświetla komunikaty ostrzegawcze, co pomaga w diagnozowaniu problemów.
7. **Kolorowe wskaźniki statusu**:
   * Przeglądy i ubezpieczenia są wyróżniane kolorami w tabeli (np. czerwony kolor dla wygasłych rekordów), co zwiększa czytelność i efektywność pracy.
8. **Prosty system statystyk**:
   * Aplikacja wyświetla statystyki, takie jak liczba pojazdów, kierowców, wygasłych ubezpieczeń czy nieważnych przeglądów, co pozwala użytkownikowi szybko uzyskać kluczowe informacje.
9. **Możliwość rozbudowy**:
   * Kod aplikacji jest napisany w modularny sposób, co pozwala na łatwe dodawanie nowych funkcji i rozszerzeń w przyszłości.