

Projekt MAS

Salon Samochodowy

Dominik Piwowarczyk

1. Wprowadzenie
2. Wymagania użytkownika
3. Diagram przypadków użycia
4. Diagram klas – analityczny
5. Diagram klas – projektowy
6. Scenariusz przypadku użycia – Wyświetl historię serwisową
7. Diagram aktywności – Wyświetl historię serwisową
8. Diagram stanu - Zgłoszenie serwisowe
9. Projekt GUI
10. Decyzje projektowe i skutki analizy dynamicznej

1. Wprowadzenie

Projekt dotyczy systemu informatycznego wspomagającego pracę salonu samochodowego, który umożliwia zarządzanie ogłoszeniami samochodowymi oraz obsługę klientów autoryzowanego serwisu samochodowego.

2. Wymagania użytkownika

System musi spełniać następujące wymagania użytkownika:

1. Wymagania ogólne:
 - System powinien umożliwiać użytkownikowi logowanie oraz rejestrację konta
 - Wszelkie dane zawarte w systemie powinny być przechowywane w sposób trwały
 - System powinien umożliwić użytkownikowi zalogowanemu wylogowanie się z systemu
 - System powinien usuwać naprawy po 5 latach od zmiany statusu na
2. Wymagania dla użytkowników niezalogowanych:
 - Przeglądanie opublikowanych ogłoszeń samochodowych (tylko ogłoszenia ze statusem „aktywne”)
 - Użytkownik ma możliwość rejestracji konta lub zalogowania się
3. Wymagania dla użytkowników zalogowanych (klientów)
 - Użytkownik może umówić się na termin oglądania danego samochodu
 - Użytkownik ma możliwość dodania danego ogłoszenia do listy życzeń
 - Użytkownik ma możliwość dodania swojego pojazdu do bazy danych
 - Użytkownik ma możliwość umówienia swojego pojazdu do autoryzowanego serwisu
 - Użytkownik ma możliwość wyświetlenia historii serwisowej w naszym serwisie dla swojego pojazdu
4. Wymagania dla sprzedawców
 - Sprzedawca ma możliwość zarządzania ogłoszeniami (dodawanie, edytowanie, usuwanie)

- Sprzedawca ma możliwość dodania nowego pojazdu do bazy danych
- Sprzedawca ma możliwość wyświetlenia listy wszystkich ogłoszeń

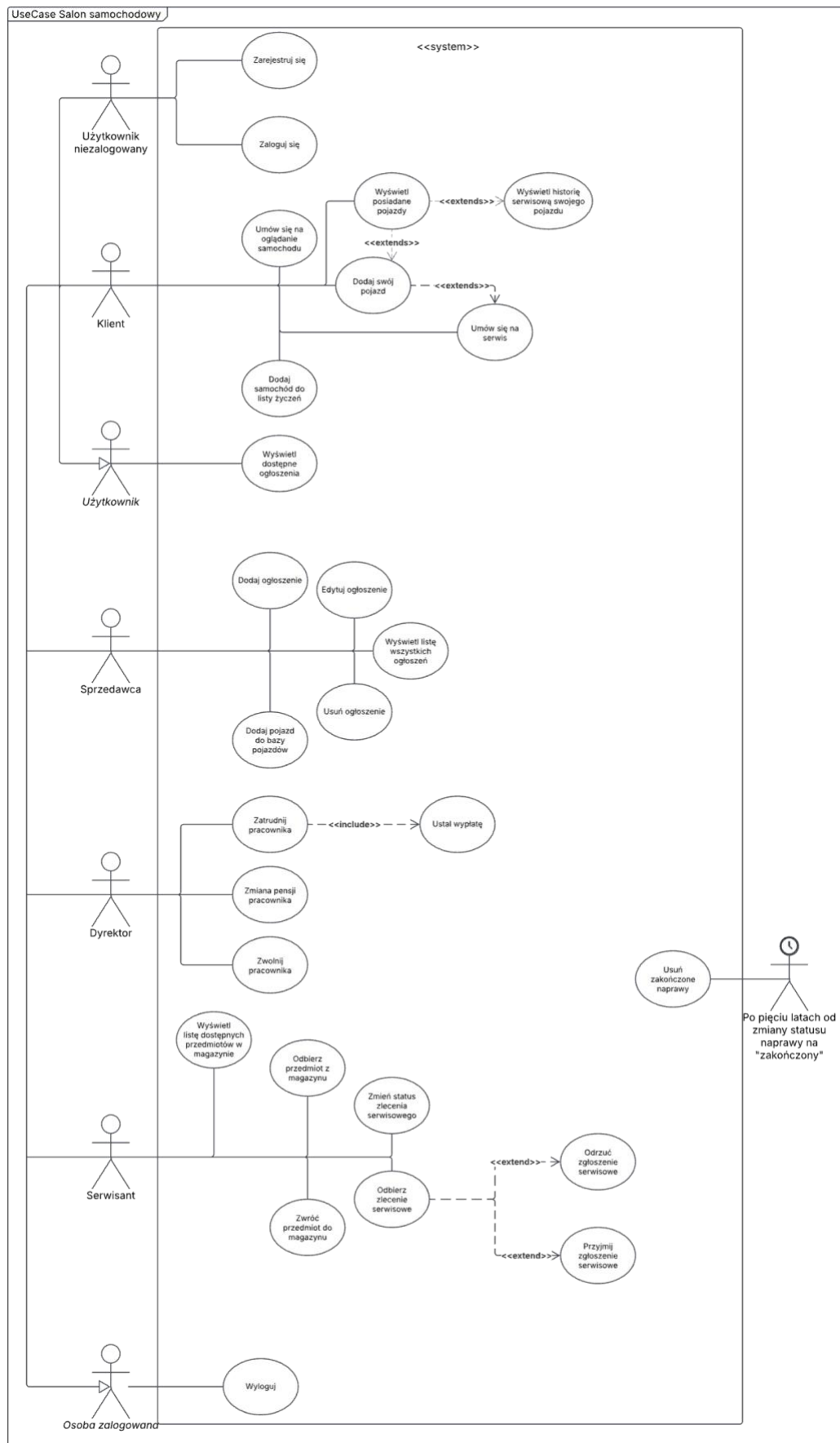
5. Wymagania dla dyrektora

- Dyrektor ma możliwość zarządzania pracownikami (dodawanie, usuwanie)
- Dyrektor ma możliwość edytowania pensji danego pracownika

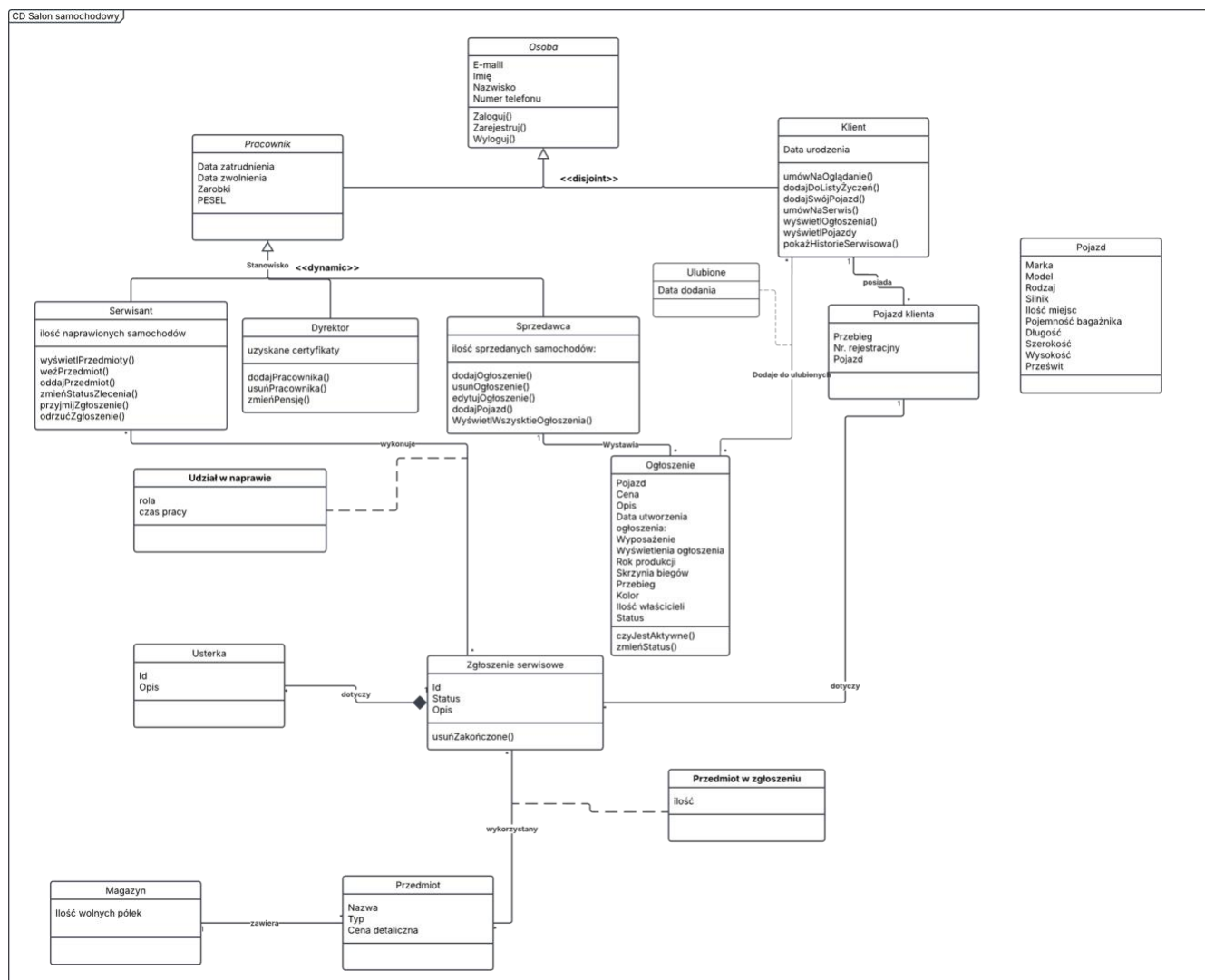
6. Wymagania dla serwisanta

- Serwisant ma możliwość wyświetlenia wszystkich przedmiotów znajdujących się w magazynie
- Serwisant ma możliwość odebrania oraz zwrotu przedmiotu z magazynu
- Serwisant ma możliwość zmiany statusu zlecenia serwisowego
- Serwisant ma możliwość przyjęcia lub odrzucenia danego zlecenia serwisowego

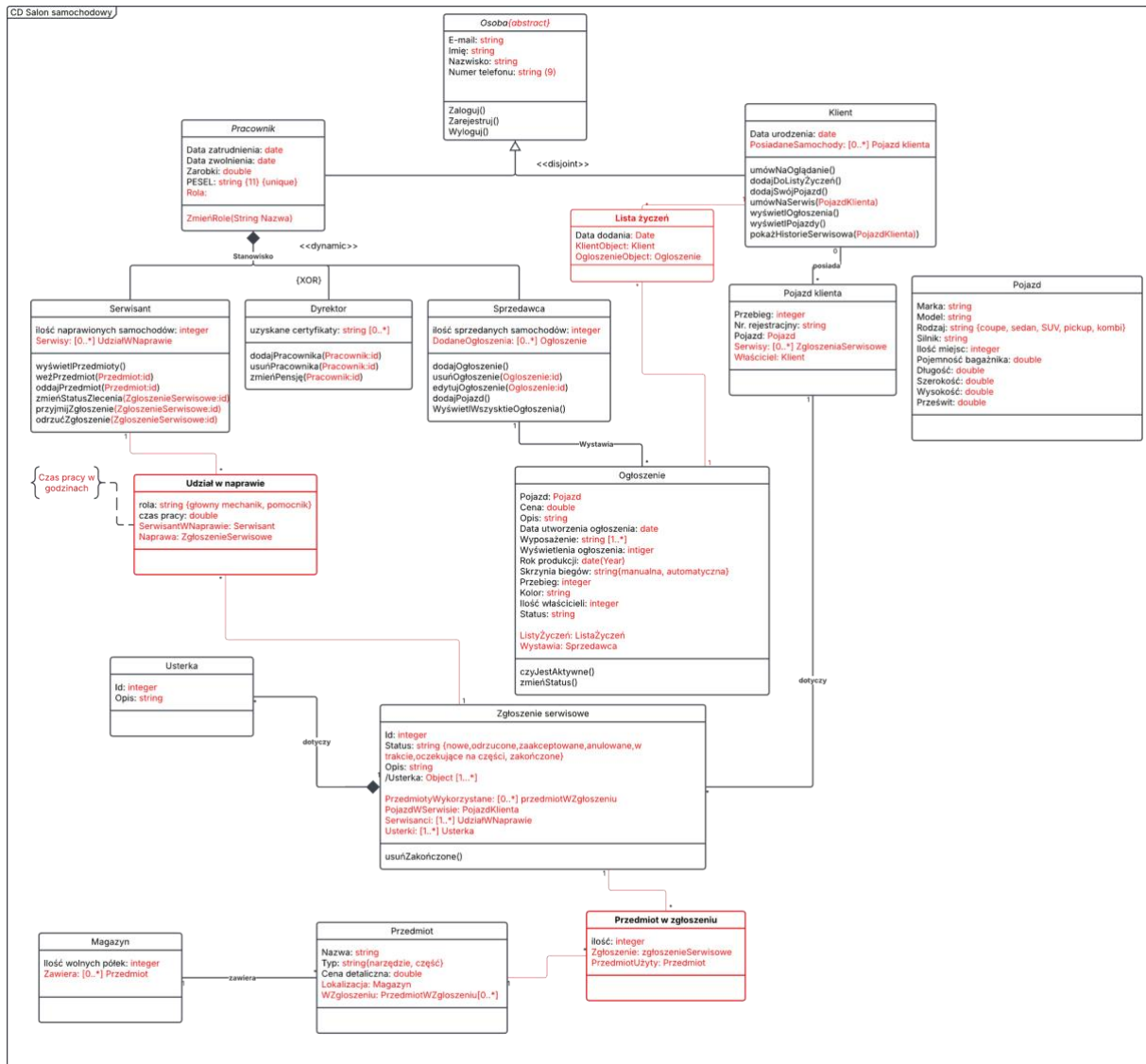
3. Diagram przypadków użycia



4. Diagram klas analityczny



5. Diagram klas – projektowy

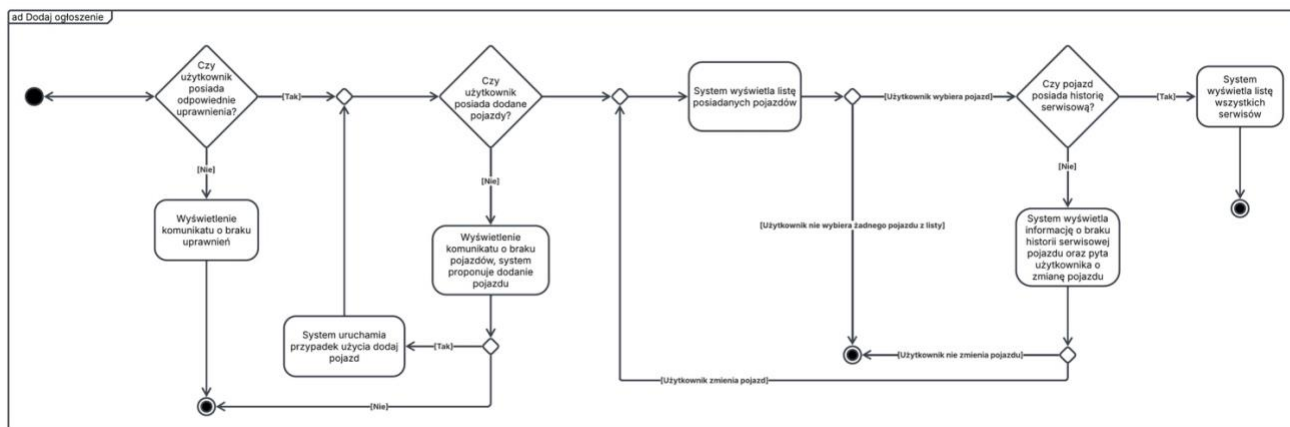


6. Scenariusz przypadku użycia – Wyświetl historię serwisową

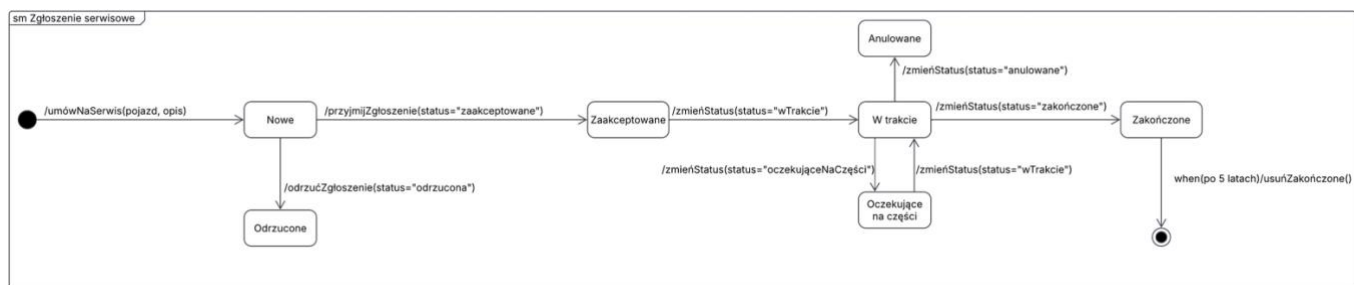
Nazwa przypadku użycia	Wyświetl historię serwisową swojego pojazdu	Id 1
Opis	Przypadek użycia umożliwia klientowi wyświetlenie całej zapisanej historii serwisowej swojego pojazdu znajdującej się w naszym serwisie	
Aktor główny	Klient	
Aktorzy pomocniczy	Brak	
Warunki początkowe	Użytkownik posiada pojazd dodany do systemu oraz wybiera dany pojazd z listy swoich pojazdów	
Scenariusz		
Krok	Akcja	
1	Użytkownik uruchamia przypadek użycia „Wyświetl listę swoich pojazdów”	
2	System wyświetla użytkownikowi listę posiadanych pojazdów	
3	Użytkownik wybiera dany pojazd z listy posiadanych pojazdów i klika „Pokaż historię serwisową	
4	System pobiera historię serwisową z bazy danych i wyświetla listę wszystkich serwisów użytkownikowi	
Scenariusz alternatywny		
2a	Użytkownik nie posiada dodanych pojazdów, system proponuje dodanie pojazdu	
2aa	Użytkownik zgadza się na dodanie pojazdu, system przechodzi do przypadku użycia „dodaj swój pojazd”	
2ab	Użytkownik odrzuca, system kończy przypadek użycia	
3a	Użytkownik nie naciska „Pokaż historię serwisową” i kończy przypadek użycia	
4a	Pojazd nie posiada historii serwisowej w bazie, system wyświetla użytkownikowi „Pojazd nie posiada historii serwisowej” i proponuje zmianę pojazdu	
4aa	Użytkownik zgadza się na zmianę pojazdu, system wraca do punktu 2.	
4ab	Użytkownik nie zgadza się na zmianę pojazdu, system kończy przypadek użycia	

Warunki końcowe	System poprawnie wyświetlił historię serwisową danego pojazdu klienta
-----------------	---

7. Diagram aktywności – Wyświetl historię serwisową

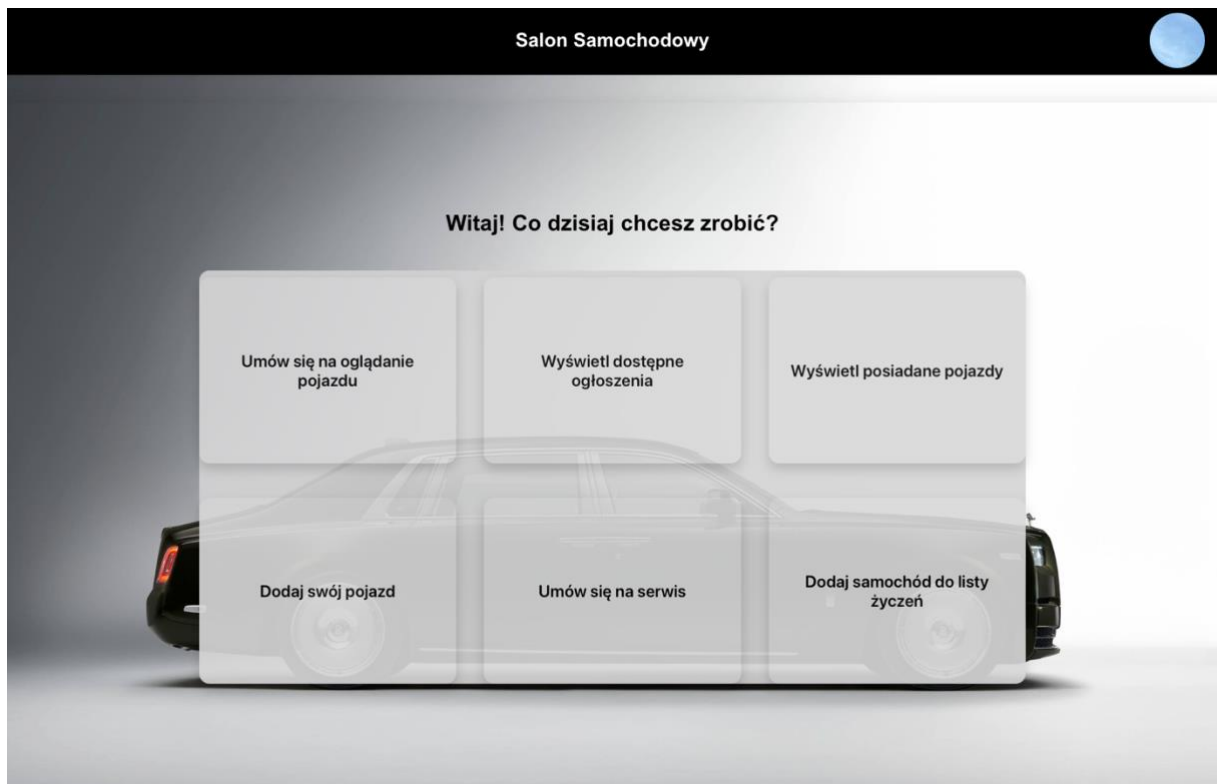


8. Diagram stanu – Zgłoszenie serwisowe



9. Projekt GUI

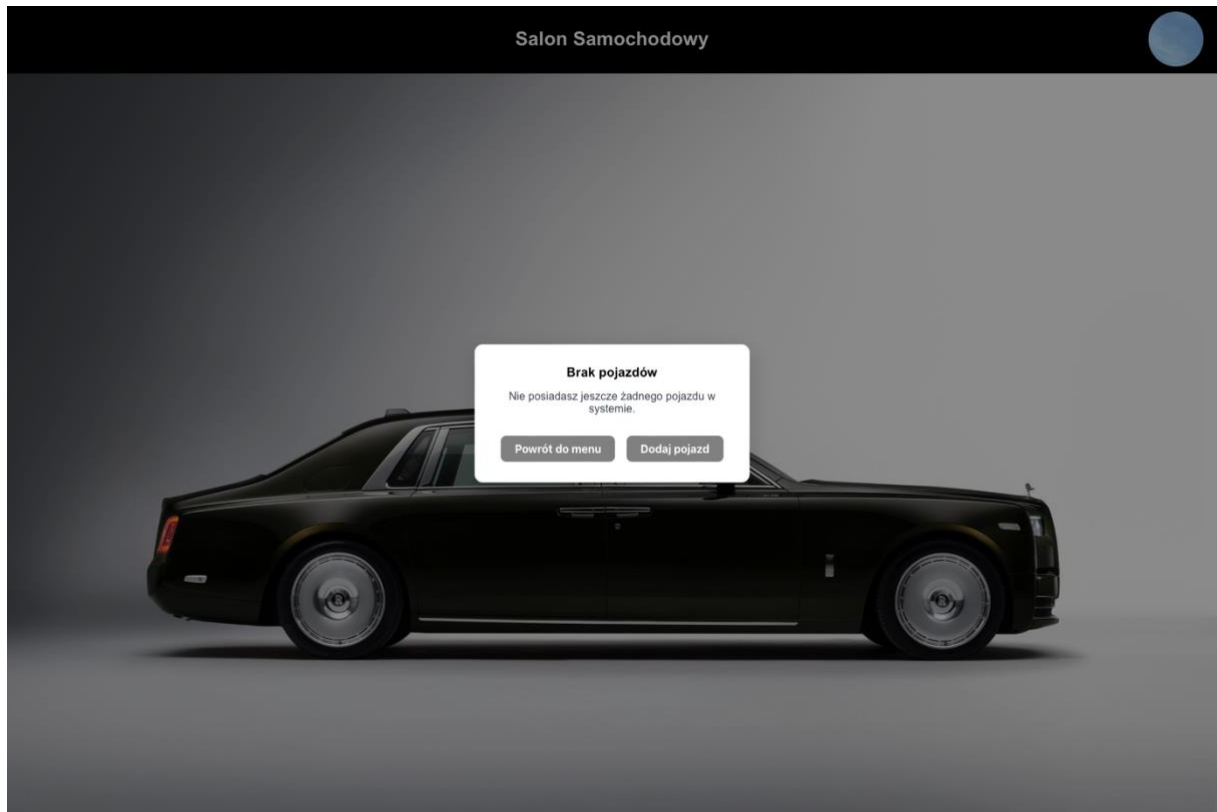
9.1 Menu główne:



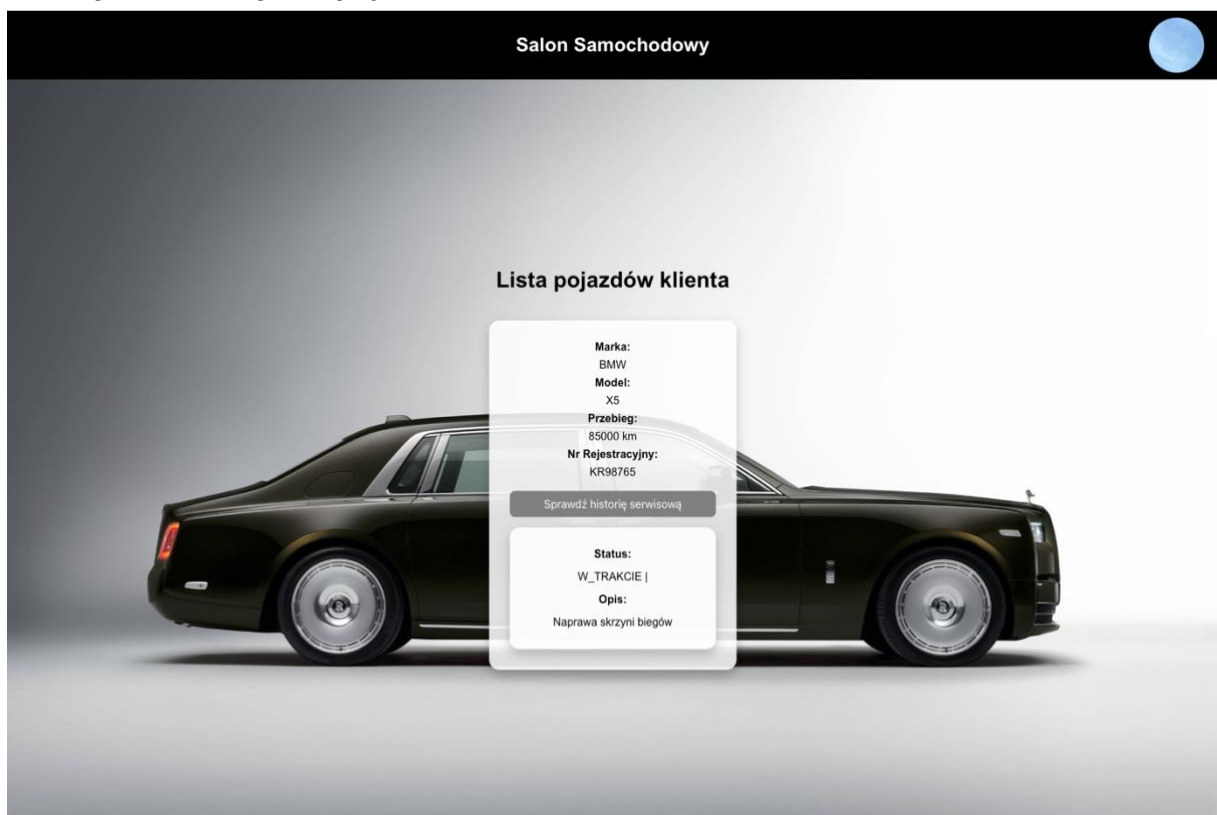
9.2 Menu wyboru pojazdu



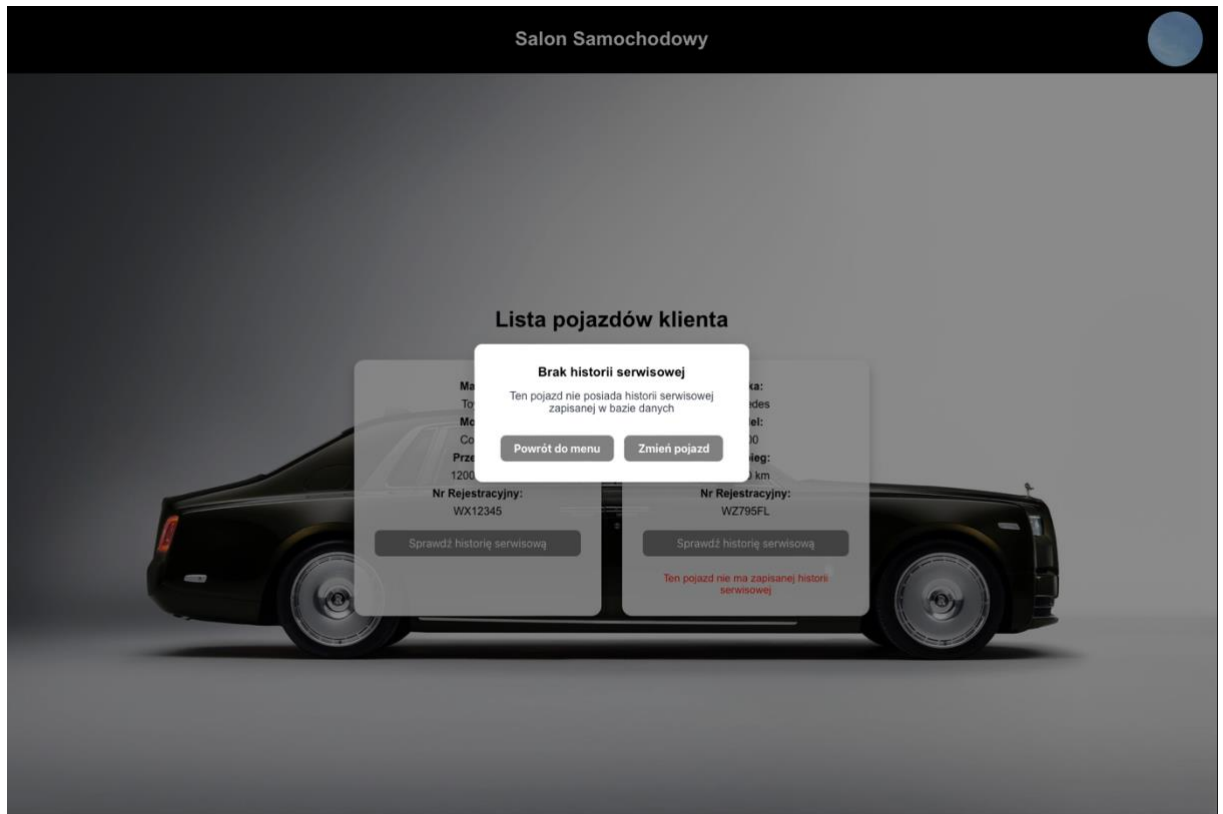
9.2.1 Użytkownik nie posiada pojazdu



9.3 Użytkownik wybrał pojazd



9.3.1 Pojazd użytkownika nie ma historii serwisowej



10. Decyzje projektowe i skutki analizy dynamicznej

Projekt „Salon samochodowy” zrealizowany będzie jako aplikacja webowa. Za frontend odpowiedzialny będzie biblioteka React.js, backend zostanie wykonany w Java Spring. Baza danych zostanie zrealizowana w H2.

Asocjacje zostaną zaimplementowane w następujące sposoby:

1. **Asocjacja 1-*:** Po stronie „1” zostanie utworzona lista obiektów (*), zaś po stronie „*” zostanie utworzona zmienna odwołująca się do obiektu (1)
2. **Asocjacja *-*:** Asocjacje wiele-wiele zostaną zrealizowane przez utworzenie klasy pośredniczącej zawierającej zmienne odwołujące się do obiektów po obu stronach asocjacji, zaś w klasach pierwotnych zostaną utworzone listy obiektów klasy pośredniczącej
3. **Kompozycja:** (na przykładzie klas usterka-zgłoszenie serwisowe) Dla klasy „usterka” zostanie utworzona zmienna referencyjna przechowująca dane zgłoszenie serwisowe, zaś dla klasy „zgłoszenie serwisowe” zostanie utworzony set zawierający wszystkie usterki dotyczące danego zgłoszenia.

Dziedziczenia zostaną zaimplementowane w następujące sposoby:

1. **Disjoint** – Zostanie utworzona klasa abstrakcyjna, klasy dziedziczące otrzymają wspólny, zawierający najważniejsze dane konstruktor oraz najważniejsze metody.
2. **Dynamic** – Wykorzystane zostanie „sprytne kopiowanie”. W klasach nadrzędnej zostanie utworzony konstruktor kopiujący, zaś w klasach podrzędnych zostaną utworzone konstruktory kopiujące najważniejsze dane z klasy ponad nimi oraz dodające dodatkowe atrybuty ważne dla danego przypadku.