



**WYŻSZA SZKOŁA
INFORMATYKI i ZARZĄDZANIA**
z siedzibą w Rzeszowie

KOLEGIUM INFORMATYKI STOSOWANEJ

Kierunek: INFORMATYKA

Specjalność: **Inżynieria Danych**

Wojciech Armata
Nr albumu studenta w69760

System zarządzania pojazdami

Prowadzący: mgr inż. Ewa Żesławska

Praca projektowa programowanie obiektowe C#

Rzeszów 2025

Spis treści

Wstęp	4
1 Opis założeń projektu	5
1.1 Cele projektu	5
1.2 Wymagania funkcjonalne i нефункционалне	5
2 Opis struktury projektu	7
2.1 Wstęp	7
2.2 Język programowania, narzędzia i wymagania sprzętowe	7
2.3 Minimalne wymagania sprzętowe	7
2.4 Zarządzanie danymi i integracja z bazą danych SQL	8
2.5 Struktura tabel SQL	8
2.6 Struktura klas i hierarchia obiektów	9
2.7 Opis najważniejszych metod	10
3 Harmonogram realizacji projektu	11
4 Prezentacja warstwy użytkowej projektu	12
4.1 Opis warstwy użytkowej	12
4.1.1 Sposób działania	18
4.1.2 Zalety podejścia	18
4.2 Podsumowanie	18
5 Podsumowanie	19
5.1 Zrealizowane prace	19
5.2 Planowane dalsze prace rozwojowe	19
Spis rysunków	20

Wstęp

Efektywne zarządzanie flotą pojazdów to kluczowy element sprawnego funkcjonowania firm i instytucji. Często jednak pojawiają się problemy z nieuporządkowanymi danymi, trudnościami w wyszukiwaniu pojazdów czy koniecznością ręcznego prowadzenia ewidencji, co prowadzi do dezorganizacji, opóźnień i zwiększonego ryzyka błędów.

Stworzony system to aplikacja konsolowa, która usprawnia te procesy, zapewniając wygodne i uporządkowane zarządzanie danymi o pojazdach. Umożliwia szybkie dodawanie, edytowanie, usuwanie oraz wyszukiwanie pojazdów na podstawie numeru rejestracyjnego lub typu. Oprócz tego pozwala generować statystyki, co ułatwia analizę floty i podejmowanie decyzji związanych z jej eksploatacją.

Kluczowym elementem systemu jest integracja z bazą SQL oraz obsługa plików, co zapewnia trwałość i bezpieczeństwo danych. Dzięki temu użytkownicy mają stały dostęp do aktualnych informacji i mogą skutecznie zarządzać flotą. Aplikacja jest prosta w obsłudze, niezawodna i gotowa do dalszej rozbudowy. To praktyczne narzędzie dla firm, które chcą efektywnie kontrolować swoje zasoby transportowe bez konieczności korzystania z rozbudowanych systemów z interfejsem graficznym.

Rozdział 1

Opis założeń projektu

1.1 Cele projektu

Celem projektu jest stworzenie intuicyjnego i skutecznego systemu do zarządzania flotą pojazdów. System ma być prosty w obsłudze, a jednocześnie wydajny, umożliwiając uporządkowanie informacji o pojazdach i ułatwiając ich zarządzanie. Pozwala na sprawne dodawanie, edytowanie i wyszukiwanie pojazdów, a także generowanie raportów wspomagających podejmowanie decyzji. Problem, który projekt ma rozwiązać, dotyczy braku spójnego narzędzia do zarządzania flotą. Wiele firm korzysta z chaotycznych arkuszy kalkulacyjnych lub nawet papierowych notatek, co prowadzi do zagubienia danych, błędów i niepotrzebnych opóźnień. Zastosowanie tego systemu ma na celu automatyzację procesów i ułatwienie codziennej pracy. Dobrze zarządzana flota oznacza niższe koszty, lepszą organizację i większą efektywność. Brak aktualnych danych w firmach transportowych często skutkuje błędnymi decyzjami, co potwierdza potrzebę wprowadzenia spójnego narzędzia. System umożliwi eliminację tych problemów i usprawni zarządzanie pojazdami. Realizacja projektu wymaga zastosowania solidnej bazy danych SQL zapewniającej stabilność i bezpieczeństwo informacji. Kluczowe jest także odpowiednie zaprojektowanie funkcji programu w sposób intuicyjny i praktyczny. Proces wdrożenia obejmuje kilka etapów: stworzenie struktury bazy danych, implementację podstawowych funkcjonalności, testowanie oraz finalne wdrożenie. Wynikiem prac będzie gotowa do użytku aplikacja konsolowa, znacznie ułatwiająca zarządzanie flotą pojazdów.

1.2 Wymagania funkcjonalne i нефункционалне

Wymagania funkcjonalne

System "Zarządzanie Pojazdami" powinien umożliwiać kompleksowe zarządzanie danymi pojazdów poprzez funkcje dodawania, edytowania oraz usuwania wpisów na podstawie numeru rejestracyjnego. Każdy pojazd powinien być opisany za pomocą szczegółowych informacji, takich jak marka, model, rok produkcji oraz dodatkowe parametry zależne od jego typu. System powinien wspierać wyszukiwanie pojazdów według różnych kryteriów, takich jak numer rejestracyjny, marka, model czy rok produkcji, co pozwoli użytkownikowi na szybkie odnalezienie potrzebnych danych. Dodatkowo, aplikacja powinna umożliwiać generowanie statystyk, takich jak liczba pojazdów w systemie czy podział na kategorie, co pozwoli na lepszą analizę i organizację danych. Kolejną kluczową funkcjonalnością jest możliwość eksportowania i importowania danych w formacie CSV, co pozwoli na łatwe przenoszenie informacji między systemami lub ich archiwizację. Ważnym elementem systemu jest także integracja z bazą SQL, co zapewni trwałe przechowywanie danych oraz możliwość ich obsługi przez różne aplikacje. Dzięki temu system będzie skalowalny i przygotowany na dalszy rozwój.

Wymagania нефункционалне

Projekt "Zarządzanie Pojazdami" to aplikacja umożliwiająca dodawanie, edytowanie, usuwanie oraz wyszukiwanie pojazdów, przechowując dane zarówno w plikach CSV, jak i w bazie danych SQL. System został zaprojektowany zgodnie z zasadami programowania obiektowego, co zapewnia jego modularność i łatwość rozbudowy. Pod względem wymagań niefunkcjonalnych, aplikacja cechuje się wysoką wydajnością, umożliwiając szybkie operacje na dużych zbiorach danych, przy czym czas odpowiedzi na standardowe operacje CRUD nie powinien przekraczać 500 ms. System jest skalowalny, co pozwala na jego rozszerzenie o dodatkowe typy pojazdów oraz obsługę różnych baz danych. Aplikacja została zaprojektowana z myślą o łatwości utrzymania, dlatego kod jest modularny i zgodny z zasadami SOLID, co ułatwia jego rozwój i modyfikacje. System jest również przenośny, działając zarówno na systemach Windows, jak i Linux, oraz obsługując różne wersje SQL Server. W zakresie dostępności i niezawodności, aplikacja zapewnia stabilne działanie oraz umożliwia tworzenie kopii zapasowych poprzez eksport danych do plików CSV lub do bazy SQL.

Rozdział 2

Opis struktury projektu

2.1 Wstęp

Projekt „Zarządzanie Pojazdami” został zrealizowany jako aplikacja desktopowa stworzona w języku C# przy użyciu platformy .NET 8. Struktura projektu została zaprojektowana w sposób modułowy, co pozwala na łatwą rozbudowę i utrzymanie systemu. W niniejszym rozdziale omówiono techniczne aspekty projektu, jego architekturę, narzędzia użyte podczas implementacji oraz szczegóły dotyczące zarządzania danymi i integracji z bazą danych SQL.

2.2 Język programowania, narzędzia i wymagania sprzętowe

Język programowania

Aplikacja została napisana w języku C# z wykorzystaniem platformy .NET 8, co zapewnia wysoką wydajność oraz możliwość uruchamiania na systemach Windows i Linux. C# jako język obiektowy ułatwia zarządzanie pamięcią i wspiera nowoczesne techniki programowania, takie jak programowanie asynchroniczne. Zastosowanie .NET 8 pozwala na optymalizację wydajności, integrację z bazami danych SQL oraz łatwą rozbudowę aplikacji w przyszłości.

Narzędzia użyte podczas tworzenia projektu:

- Visual Studio – środowisko programistyczne wykorzystywane do implementacji i debugowania kodu.
- SQL Server Management Studio (SSMS) – narzędzie do zarządzania bazą danych SQL.
- Microsoft SQL Server – baza danych wykorzystywana do przechowywania informacji o pojazdach.

2.3 Minimalne wymagania sprzętowe

- Procesor: Dwurdzeniowy, 1.8 GHz lub nowszy.
- Pamięć RAM: 2 GB (zalecane 4 GB dla lepszej wydajności).
- Dysk twardy: 250 MB wolnego miejsca na pliki aplikacji i bazę danych.
- System operacyjny: Windows 10/11 lub Linux z obsługą .NET 8.

2.4 Zarządzanie danymi i integracja z bazą danych SQL

System zarządza danymi pojazdów poprzez operacje na plikach TXT oraz bazie danych SQL.

- Pliki TXT służą do lokalnego przechowywania danych i eksportu/importu informacji.
- Baza SQL zapewnia centralne przechowywanie pojazdów, umożliwiając ich szybkie przeszukiwanie i manipulację.

W aplikacji zastosowano Microsoft.Data.SqlClient do komunikacji z bazą danych SQL, co zapewnia efektywne wykonywanie operacji CRUD (Create, Read, Update, Delete). Dodatkowo, aplikacja umożliwia importowanie danych z plików TXT do bazy SQL. Proces ten obejmuje:

- Odczytanie danych z plików tekstowych.
- Parsowanie i weryfikację poprawności rekordów.
- Wstawienie unikalnych rekordów do bazy SQL, aby uniknąć duplikacji.
- Obsługę błędów i logowanie wyników importu.

2.5 Struktura tabel SQL

Dane są przechowywane w bazie danych, gdzie każda encja posiada swoją tabelę.

- Pojazdy
 - ID - Klucz główny
 - Typ - Typ pojazdu (np. Samochód, Motocykl, Autobus)
 - Marka - Marka pojazdu
 - Model - Model pojazdu
 - RokProdukcji - Rok produkcji pojazdu
 - NumerRejestracyjny - Numer rejestracyjny pojazdu
 - DataDodania - Data dodania pojazdu do bazy
- Autobusy
 - ID
 - LiczbaMiejsc - Liczba miejsc w autobusie
 - Dlugosc - Długość autobusu
- Ciezarowki
 - ID
 - Ladownosc - Ładowność ciężarówki
 - Dlugosc - Długość ciężarówki
- Dostawczaki
 - ID
 - Ladownosc - Ładowność dostawczaka

- Motocykle
 - ID
 - Pojemnosc - Pojemność silnika motocykla
- SamochodyOsobowe
 - ID
 - LiczbaMiejsc - Liczba miejsc w samochodzie osobowym
- SamochodyElektryczne
 - ID
 - LiczbaMiejsc - Liczba miejsc w samochodzie elektrycznym
 - PojemnoscBaterii - Pojemność baterii w kWh
 - Zasięg - Zasięg pojazdu w km

2.6 Struktura klas i hierarchia obiektów

Interfejsy

- IPojazd – definiuje podstawowe operacje dla pojazdów.
- IBazaDanych – określa metody obsługi bazy danych.

Klasy abstrakcyjne

- Pojazd – klasa bazowa dla różnych typów pojazdów, zawierająca wspólne atrybuty i metody.
- BazaDanychBase – klasa abstrakcyjna dla różnych implementacji baz danych.

Klasy konkretne

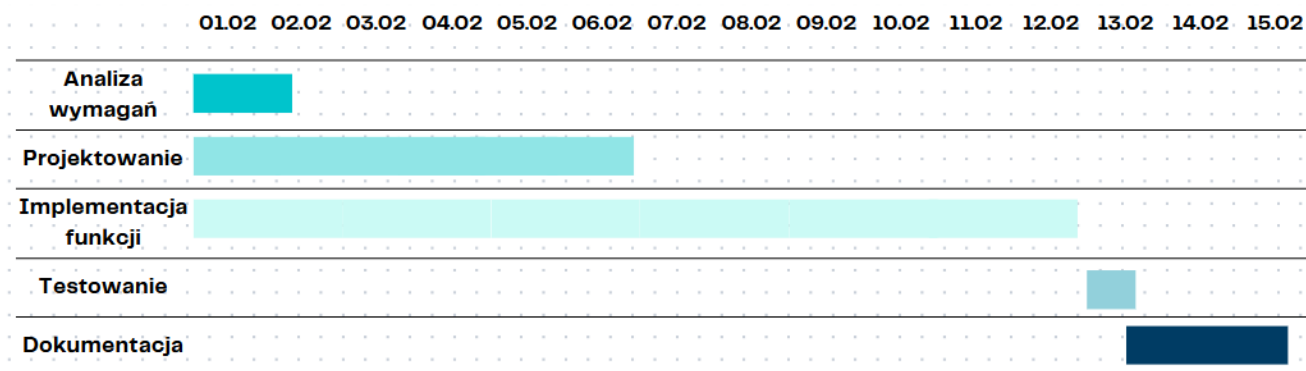
- Autobus, Ciezarowka, Motocykl, SamochodOsobowy, SamochodElektryczny, Dostawczak – klasy dziedziczące po Pojazd, zawierające specyficzne właściwości.
- BazaDanych – implementacja operacji na danych w plikach TXT.
- BazaDanychSQL – obsługa bazy SQL.
- BazaDanychHybrydowa – klasa łącząca obsługę plików TXT i bazy SQL.

Diagram klas

Rozdział 3

Harmonogram realizacji projektu

Realizacja projektu "Zarządzanie pojazdami" została zaplanowana i przeprowadzona zgodnie z harmonogram realizacji projektu opartym na diagram Gantta. Przedstawia on poszczególne etapy projektu rozłożone w czasie.



Rysunek 3.1: Diagram Gauntta

Etapy realizacji projektu na podstawie diagramu Gantta:

- Analiza wymagań – określenie funkcjonalności i wymagań projektu.
- Projektowanie systemu – zaprojektowanie struktury klas, interfejsów i diagramu klas.
- Implementacja systemu – implementacja klas, interfejsów i testowanie funkcjonalności.
- Testowanie systemu – sprawdzenie poprawności działania systemu i naprawa błędów.
- Dokumentacja projektu – przygotowanie dokumentacji technicznej i użytkownika.

Repozytorium i system kontroli wersji

Projekt został zrealizowany z wykorzystaniem systemu kontroli wersji Git oraz platformy GitHub. Repozytorium projektu znajduje się pod adresem: <https://github.com/w69760/Labolatoria>

Rozdział 4

Prezentacja warstwy użytkowej projektu

4.1 Opis warstwy użytkowej

Warstwa użytkowa w omawianej aplikacji została zaimplementowana w postaci konsolowego programu (`Program.cs`). Użytkownik wchodzi w interakcję z systemem poprzez menu wyświetlane w konsoli, w którym dostępne są następujące opcje:

```
Polaczenie z baza danych dziala poprawnie.  
  
--- MENU ---  
1. Dodaj nowy pojazd  
2. Usun pojazd  
3. Edytuj rejestracje pojazdu  
4. Wyszukaj pojazd po rejestracji  
5. Wyświetl wszystkie pojazdy  
6. Wyświetl pojazdy danego typu  
7. Pokaz statystyki pojazdow  
8. Eksportuj wszystkie pojazdy z pliku do SQL  
0. Wyjście  
Wybierz opcje:
```

Rysunek 4.1: Menu główne aplikacji

- **Dodawanie nowego pojazdu** – użytkownik wprowadza dane pojazdu (marka, model, rok itp.), a program wywołuje odpowiednie metody warstwy dostępowej do bazy (np. `DodajNowyPojazd`).

```
Polaczenie z baza danych dziala poprawnie.

--- MENU ---
1. Dodaj nowy pojazd
2. Usun pojazd
3. Edytuj rejestracje pojazdu
4. Wyszukaj pojazd po rejestracji
5. Wyswietl wszystkie pojazdy
6. Wyswietl pojazdy danego typu
7. Pokaz statystyki pojazdow
8. Eksportuj wszystkie pojazdy z pliku do SQL
0. Wyjście
Wybierz opcje: 1

--- Wybierz typ pojazdu ---
1. Autobus
2. Ciezarowka
3. Dostawczak
4. Motocykl
5. Samochod osobowy
6. Samochod elektryczny
Wybierz numer (1-6): 4
Marka: Kawasaki
Model:
```

Rysunek 4.2: Dodawanie pojazdu

- **Usuwanie pojazdu** – użytkownik podaje numer rejestracyjny, a aplikacja usuwa pojazd z bazy plikowej i bazy SQL.

```
--- MENU ---
1. Dodaj nowy pojazd
2. Usun pojazd
3. Edytuj rejestracje pojazdu
4. Wyszukaj pojazd po rejestracji
5. Wyswietl wszystkie pojazdy
6. Wyswietl pojazdy danego typu
7. Pokaz statystyki pojazdow
8. Eksportuj wszystkie pojazdy z pliku do SQL
0. Wyjście
Wybierz opcje: 2
Podaj rejestracje pojazdu do usuniecia: PER620
Pojazd usuniety.
```

Rysunek 4.3: Usuwanie pojazdu

- **Edycja rejestracji** – umożliwia zmianę numeru rejestracyjnego istniejącego pojazdu.

```
--- MENU ---
1. Dodaj nowy pojazd
2. Usun pojazd
3. Edytuj rejestracje pojazdu
4. Wyszukaj pojazd po rejestracji
5. Wyswietl wszystkie pojazdy
6. Wyswietl pojazdy danego typu
7. Pokaz statystyki pojazdow
8. Eksportuj wszystkie pojazdy z pliku do SQL
0. Wyjście
Wybierz opcje: 3
Podaj stara rejestracje: BMW6789
Podaj nowa rejestracje: PLK100
Rejestracja zaktualizowana.
```

Rysunek 4.4: Edycja rejestracji

- **Wyszukiwanie pojazdu** – na podstawie rejestracji program wyświetla informacje o odna-

lezionejnym pojeździe (lub komunikat o braku pojazdu).

```
--- MENU ---
1. Dodaj nowy pojazd
2. Usun pojazd
3. Edytuj rejestracje pojazdu
4. Wyszukaj pojazd po rejestracji
5. Wyświetl wszystkie pojazdy
6. Wyświetl pojazdy danego typu
7. Pokaz statystyki pojazdow
8. Eksportuj wszystkie pojazdy z pliku do SQL
0. Wyjście
Wybierz opcje: 4
Podaj rejestracje: PLK100
ID: 28, Typ: SamochodOsobowy, Marka: BMW, Model: Seria 5, Rok: 2021, Rejestracja: PLK100, Data dodania: 18.02.2025, Liczba miejsc: 5
```

Rysunek 4.5: Wyszukiwanie pojazdu po numerze rejestracyjnym

- **Wyświetlanie pojazdów** – aplikacja pobiera listę wszystkich pojazdów z warstwy dostępnej i prezentuje je w konsoli.

```
--- MENU ---
1. Dodaj nowy pojazd
2. Usun pojazd
3. Edytuj rejestracje pojazdu
4. Wyszukaj pojazd po rejestracji
5. Wyświetl wszystkie pojazdy
6. Wyświetl pojazdy danego typu
7. Pokaz statystyki pojazdow
8. Eksportuj wszystkie pojazdy z pliku do SQL
0. Wyjście
Wybierz opcje: 5
ID: 1, Typ: Autobus, Marka: MAN, Model: Lions Coach, Rok: 2015, Rejestracja: MAM1234, Data dodania: 18.02.2025, Miejsca: 50, Dlugosc: 12 m
ID: 2, Typ: Autobus, Marka: Solaris, Model: Urbino, Rok: 2020, Rejestracja: KRK5678, Data dodania: 18.02.2025, Miejsca: 50, Dlugosc: 12 m
ID: 3, Typ: Autobus, Marka: Mercedes, Model: Tourismo, Rok: 2018, Rejestracja: POZ8765, Data dodania: 18.02.2025, Miejsca: 50, Dlugosc: 12 m
ID: 4, Typ: Ciezarowka, Marka: Scania, Model: R500, Rok: 2019, Rejestracja: GDA3456, Data dodania: 18.02.2025, Ladownosc: 10 t, Dlugosc: 8 m
ID: 5, Typ: Ciezarowka, Marka: Volvo, Model: FH16, Rok: 2021, Rejestracja: SGP1234, Data dodania: 18.02.2025, Ladownosc: 10 t, Dlugosc: 8 m
ID: 6, Typ: Ciezarowka, Marka: MAN, Model: TGX, Rok: 2020, Rejestracja: WRO6543, Data dodania: 18.02.2025, Ladownosc: 10 t, Dlugosc: 8 m
ID: 7, Typ: Dostawczak, Marka: Opel, Model: Vivaro, Rok: 2017, Rejestracja: LUI2345, Data dodania: 18.02.2025, Ladownosc: 3,5 t
ID: 8, Typ: Dostawczak, Marka: Renault, Model: Master, Rok: 2020, Rejestracja: PO32145, Data dodania: 18.02.2025, Ladownosc: 3,5 t
ID: 9, Typ: Motocykl, Marka: Yamaha, Model: MT-07, Rok: 2021, Rejestracja: DUA123, Data dodania: 18.02.2025, Pojemnosc silnika: 600 ccm
ID: 10, Typ: Motocykl, Marka: Honda, Model: CBR500, Rok: 2020, Rejestracja: LUB4567, Data dodania: 18.02.2025, Pojemnosc silnika: 600 ccm
ID: 11, Typ: Motocykl, Marka: Suzuki, Model: GSX-R750, Rok: 2019, Rejestracja: KAT7654, Data dodania: 18.02.2025, Pojemnosc silnika: 600 ccm
ID: 12, Typ: SamochodOsobowy, Marka: Toyota, Model: Corolla, Rok: 2018, Rejestracja: WAR7890, Data dodania: 18.02.2025, Liczba miejsc: 5
ID: 13, Typ: SamochodOsobowy, Marka: Volkswagen, Model: Golf, Rok: 2019, Rejestracja: GDA1237, Data dodania: 18.02.2025, Liczba miejsc: 5
ID: 14, Typ: SamochodOsobowy, Marka: Skoda, Model: Octavia, Rok: 2020, Rejestracja: WRO4568, Data dodania: 18.02.2025, Liczba miejsc: 5
ID: 15, Typ: SamochodElektryczny, Marka: Tesla, Model: Model 3, Rok: 2022, Rejestracja: TES1234, Data dodania: 18.02.2025, Liczba miejsc: 5, Pojemnosc baterii: 60 kWh, Zasięg: 400 km
ID: 16, Typ: SamochodElektryczny, Marka: Nissan, Model: Leaf, Rok: 2021, Rejestracja: NIS5678, Data dodania: 18.02.2025, Liczba miejsc: 5, Pojemnosc baterii: 60 kWh, Zasięg: 400 km
ID: 17, Typ: SamochodElektryczny, Marka: BMW, Model: i3, Rok: 2020, Rejestracja: BMW7891, Data dodania: 18.02.2025, Liczba miejsc: 5, Pojemnosc baterii: 60 kWh, Zasięg: 400 km
ID: 18, Typ: SamochodElektryczny, Marka: Audi, Model: e-tron, Rok: 2023, Rejestracja: AUD4567, Data dodania: 18.02.2025, Liczba miejsc: 5, Pojemnosc baterii: 60 kWh, Zasięg: 400 km
ID: 19, Typ: Ciezarowka, Marka: DAF, Model: XF, Rok: 2022, Rejestracja: DAF7890, Data dodania: 18.02.2025, Ladownosc: 10 t, Dlugosc: 8 m
ID: 20, Typ: Dostawczak, Marka: Mercedes, Model: Sprinter, Rok: 2019, Rejestracja: SPR6543, Data dodania: 18.02.2025, Ladownosc: 3,5 t
ID: 21, Typ: Autobus, Marka: Iveco, Model: Crossway, Rok: 2021, Rejestracja: IVE6789, Data dodania: 18.02.2025, Miejsca: 50, Dlugosc: 12 m
ID: 22, Typ: SamochodElektryczny, Marka: Porsche, Model: Taycan, Rok: 2023, Rejestracja: POR9876, Data dodania: 18.02.2025, Liczba miejsc: 5, Pojemnosc baterii: 60 kWh, Zasięg: 400 km
ID: 23, Typ: Motocykl, Marka: Ducati, Model: Panigale V4, Rok: 2022, Rejestracja: DUC5432, Data dodania: 18.02.2025, Pojemnosc silnika: 600 ccm
ID: 24, Typ: Autobus, Marka: Neoplan, Model: Skyliner, Rok: 2018, Rejestracja: NEO3214, Data dodania: 18.02.2025, Miejsca: 50, Dlugosc: 12 m
ID: 25, Typ: Dostawczak, Marka: Fiat, Model: Ducato, Rok: 2019, Rejestracja: DUC7890, Data dodania: 18.02.2025, Ladownosc: 3,5 t
ID: 26, Typ: Ciezarowka, Marka: Iveco, Model: Stralis, Rok: 2020, Rejestracja: IVE5678, Data dodania: 18.02.2025, Ladownosc: 10 t, Dlugosc: 8 m
ID: 27, Typ: Autobus, Marka: Setra, Model: S415, Rok: 2022, Rejestracja: SET4567, Data dodania: 18.02.2025, Miejsca: 50, Dlugosc: 12 m
ID: 28, Typ: SamochodOsobowy, Marka: BMW, Model: Seria 5, Rok: 2021, Rejestracja: PLK100, Data dodania: 18.02.2025, Liczba miejsc: 5
ID: 29, Typ: SamochodElektryczny, Marka: Hyundai, Model: Kona EV, Rok: 2022, Rejestracja: HYU1234, Data dodania: 18.02.2025, Liczba miejsc: 5, Pojemnosc baterii: 60 kWh, Zasięg: 400 km
ID: 30, Typ: Motocykl, Marka: Kawasaki, Model: Ninja 650, Rok: 2021, Rejestracja: KAW9876, Data dodania: 18.02.2025, Pojemnosc silnika: 600 ccm
```

Rysunek 4.6: Wyświetlanie wszystkich pojazdów

- **Wyświetlanie pojazdów danego typu** – użytkownik wybiera typ (np. *Autobus*, *Ciezarowka*, *Motocykl*), a program wyświetla pasujące rekordy.

```
--- MENU ---
1. Dodaj nowy pojazd
2. Usun pojazd
3. Edytuj rejestracje pojazdu
4. Wyszukaj pojazd po rejestracji
5. Wyświetl wszystkie pojazdy
6. Wyświetl pojazdy danego typu
7. Pokaz statystyki pojazdow
8. Eksportuj wszystkie pojazdy z pliku do SQL
0. Wyjście
Wybierz opcje: 6
--- Wybierz typ pojazdu ---
1. Autobus
2. Ciezarowka
3. Dostawczak
4. Motocykl
5. Samochod osobowy
6. Samochod elektryczny
Wybierz numer (1-6): 4
ID: 9, Typ: Motocykl, Marka: Yamaha, Model: MT-07, Rok: 2021, Rejestracja: DUA123, Data dodania: 18.02.2025, Pojemnosc silnika: 600 ccm
ID: 10, Typ: Motocykl, Marka: Honda, Model: CBR500, Rok: 2020, Rejestracja: LUB4567, Data dodania: 18.02.2025, Pojemnosc silnika: 600 ccm
ID: 11, Typ: Motocykl, Marka: Suzuki, Model: GSX-R750, Rok: 2019, Rejestracja: KAT7654, Data dodania: 18.02.2025, Pojemnosc silnika: 600 ccm
ID: 23, Typ: Motocykl, Marka: Ducati, Model: Panigale V4, Rok: 2022, Rejestracja: DUC5432, Data dodania: 18.02.2025, Pojemnosc silnika: 600 ccm
ID: 30, Typ: Motocykl, Marka: Kawasaki, Model: Ninja 650, Rok: 2021, Rejestracja: KAW9876, Data dodania: 18.02.2025, Pojemnosc silnika: 600 ccm
```

Rysunek 4.7: Wyświetlanie danego typu pojazdow

- **Statystyki** – pozwala na wyświetlenie informacji statystycznych (np. liczba pojazdów w każdej kategorii).

```
--- MENU ---
1. Dodaj nowy pojazd
2. Usun pojazd
3. Edytuj rejestracje pojazdu
4. Wyszukaj pojazd po rejestracji
5. Wswietl wszystkie pojazdy
6. Wswietl pojazdy danego typu
7. Pokaz statystyki pojazdow
8. Eksportuj wszystkie pojazdy z pliku do SQL
0. Wyjście
Wybierz opcje: 7
Autobus: 6 pojazdow
Ciezarowka: 5 pojazdow
Dostawczak: 4 pojazdow
Motocykl: 5 pojazdow
SamochodElektryczny: 6 pojazdow
SamochodOsobowy: 4 pojazdow
Autobus: 6 pojazdow
Ciezarowka: 5 pojazdow
Dostawczak: 4 pojazdow
Motocykl: 5 pojazdow
SamochodOsobowy: 8 pojazdow
SamochodElektryczny: 6 pojazdow
```

Rysunek 4.8: Statystyki pojazdów

- **Eksport do SQL** – umożliwia przeniesienie danych z bazy plikowej do bazy SQL (z zachowaniem walidacji).


```

--- MENU ---
1. Dodaj nowy pojazd
2. Usun pojazd
3. Edytuj rejestracje pojazdu
4. Wyszukaj pojazd po rejestracji
5. Wyszwietl wszystkie pojazdy
6. Wyszwietl pojazdy danego typu
7. Pokaz statystyki pojazdow
8. Eksportuj wszystkie pojazdy z pliku do SQL
0. Wyjście
Wybierz opcje: 8
Eksportowanie 30 pojazdow do SQL...
Pojazd WAW1234 juz istnieje w bazie SQL. Pominiecie.
Pojazd KRK5678 juz istnieje w bazie SQL. Pominiecie.
Pojazd POZ8765 juz istnieje w bazie SQL. Pominiecie.
Pojazd GDA3456 juz istnieje w bazie SQL. Pominiecie.
Pojazd SOP1234 juz istnieje w bazie SQL. Pominiecie.
Pojazd WRO6543 juz istnieje w bazie SQL. Pominiecie.
Pojazd LU12345 juz istnieje w bazie SQL. Pominiecie.
Pojazd PO32145 juz istnieje w bazie SQL. Pominiecie.
Pojazd DUPA123 juz istnieje w bazie SQL. Pominiecie.
Pojazd LUB4567 juz istnieje w bazie SQL. Pominiecie.
Pojazd KAT7654 juz istnieje w bazie SQL. Pominiecie.
Pojazd WAR7890 juz istnieje w bazie SQL. Pominiecie.
Pojazd GDA1237 juz istnieje w bazie SQL. Pominiecie.
Pojazd WRO4568 juz istnieje w bazie SQL. Pominiecie.
Pojazd TES1234 juz istnieje w bazie SQL. Pominiecie.
Pojazd NIS5678 juz istnieje w bazie SQL. Pominiecie.
Pojazd BMW7891 juz istnieje w bazie SQL. Pominiecie.
Pojazd AUD4567 juz istnieje w bazie SQL. Pominiecie.
Pojazd DAF7890 juz istnieje w bazie SQL. Pominiecie.
Pojazd SPR6543 juz istnieje w bazie SQL. Pominiecie.
Pojazd IVE6789 juz istnieje w bazie SQL. Pominiecie.
Pojazd POR9876 juz istnieje w bazie SQL. Pominiecie.
Pojazd DUC5432 juz istnieje w bazie SQL. Pominiecie.
Pojazd NEO3214 juz istnieje w bazie SQL. Pominiecie.
Pojazd DUC7896 juz istnieje w bazie SQL. Pominiecie.
Pojazd IVE5678 juz istnieje w bazie SQL. Pominiecie.
Pojazd SET4567 juz istnieje w bazie SQL. Pominiecie.
Pojazd PLK100 juz istnieje w bazie SQL. Pominiecie.
Pojazd HYU1234 juz istnieje w bazie SQL. Pominiecie.
Pojazd KAW9876 juz istnieje w bazie SQL. Pominiecie.
Eksport zakonczony! Dodano: 0, Pominiecie: 30, Bledne dane: 0

```

Rysunek 4.9: Eksport danych do SQL

4.1.1 Sposób działania

Cała komunikacja z użytkownikiem odbywa się w pętli **while**, która wyświetla menu i w zależności od wybranej opcji wywołuje konkretne metody warstwy dostępowej. Dzięki temu warstwa użytkowa jest odpowiedzialna wyłącznie za:

1. Pobieranie danych wejściowych od użytkownika,
2. Wywoływanie metod z warstwy logiki/baz danych (np. `DodajPojazd`, `UsunPojazd`),
3. Prezentację wyników w konsoli (np. listy pojazdów, komunikaty o błędach).

4.1.2 Zalety podejścia

- **Oddzielenie logiki biznesowej od interfejsu** – kod obsługujący dane i bazę jest w odrębnych klasach, co ułatwia modyfikacje i testy.
- **Łatwa rozbudowa** – w przyszłości można zastąpić interfejs konsolowy np. interfejsem graficznym, bez konieczności modyfikowania metod logiki lub bazy danych.
- **Przejrzystość** – menu konsolowe jest intuicyjne w obsłudze, a kolejne opcje (dodawanie, usuwanie, wyszukiwanie) są jasno wyodrębnione.

4.2 Podsumowanie

Warstwa użytkowa w postaci aplikacji konsolowej stanowi prosty, ale skuteczny sposób na interakcję z systemem **ZarządzaniePojazdami**. Dzięki czytelnej strukturze menu oraz wywoływaniu metod z warstwy logiki i dostępu do danych, użytkownik może w łatwy sposób dodawać, usuwać, modyfikować i przeglądać pojazdy w bazie.

Rozdział 5

Podsumowanie

W niniejszym rozdziale przedstawiono zrealizowane prace związane z projektem oraz kierunki dalszego rozwoju systemu.

5.1 Zrealizowane prace

W ramach projektu udało się zrealizować następujące elementy:

- Stworzenie wielowarstwowej architektury aplikacji, obejmującej warstwę dostępu do danych oraz interfejs użytkownika.
- Implementacja systemu zarządzania pojazdami, w tym obsługa dodawania, usuwania, edycji rejestracji, wyszukiwania oraz wyświetlania pojazdów.
- Integracja danych z dwóch źródeł – bazy plikowej i bazy SQL – umożliwiająca eksport danych oraz synchronizację między nimi.
- Opracowanie konsolowego interfejsu użytkownika, który zapewnia intuicyjną obsługę systemu poprzez menu oraz wizualizację operacji (poprzez dołączone zrzuty ekranu).
- Zabezpieczenie operacji na danych poprzez implementację odpowiednich walidacji oraz obsługi wyjątków.

5.2 Planowane dalsze prace rozwojowe

W kolejnych etapach rozwoju projektu planowane są następujące działania:

- Rozszerzenie funkcjonalności interfejsu użytkownika poprzez wdrożenie graficznego interfejsu (GUI), co umożliwi bardziej przyjazną interakcję z systemem.
- Ulepszenie mechanizmu walidacji danych i obsługi błędów, aby zwiększyć niezawodność aplikacji w środowisku produkcyjnym.
- Integracja systemu z nowoczesnymi rozwiązaniami bazodanowymi oraz API, co pozwoli na lepszą skalowalność i integrację z innymi systemami.
- Przeprowadzenie testów wydajnościowych oraz optymalizacja operacji na dużych zbiorach danych, aby zapewnić szybki dostęp do informacji.
- Rozwój modułu raportowania i analizy statystyk, co umożliwi generowanie szczegółowych raportów oraz lepsze podejmowanie decyzji na podstawie zgromadzonych danych.

Spis rysunków

2.1	Diagram klas	10
3.1	Diagram Gauntta	11
4.1	Menu główne aplikacji	12
4.2	Dodawanie pojazdu	13
4.3	Usuwanie pojazdu	14
4.4	Edycja rejestracji	14
4.5	Wyszukiwanie pojazdu po numerze rejestracyjnym	15
4.6	Wyświetlanie wszystkich pojazdów	15
4.7	Wyświetlanie danego typu pojazdów	15
4.8	Statystyki pojazdów	16
4.9	Eksport danych do SQL	17