



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

KOLEGIUM INFORMATYKI STOSOWANEJ

Kierunek: INFORMATYKA

Specjalność: Programowanie

System Zarządzania Pojazdami

Jan Gorazd

Nr albumu studenta w69972

Prowadzący: mgr inż. Ewa Żesławska

Praca projektowa programowanie obiektowe C#

Rzeszów 2025

Spis treści

| | | |
|----------|--|----------|
| 1 | Wstęp | 2 |
| 2 | Opis założeń projektu | 2 |
| 2.1 | Cele projektu | 2 |
| 2.2 | Wymagania funkcjonalne | 2 |
| 2.3 | Wymagania niefunkcjonalne | 2 |
| 3 | Opis struktury projektu | 3 |
| 3.1 | Technologie i narzędzia | 3 |
| 3.2 | Struktura bazy danych | 3 |
| 3.3 | Struktura programowania | 4 |
| 4 | Harmonogram realizacji projektu | 4 |
| 4.1 | Informacje o repozytorium i systemie kontroli wersji | 4 |
| 5 | Prezentacja warstwy użytkowej projektu | 5 |
| 5.1 | Funkcjonalności | 5 |
| 5.2 | Interakcja użytkownika z systemem | 5 |
| 6 | Podsumowanie | 6 |

1 Wstęp

Celem projektu jest stworzenie aplikacji umożliwiającej zarządzanie danymi klientów oraz pojazdów w firmie zajmującej się flotą pojazdów. System ten ma na celu zautomatyzowanie wielu procesów administracyjnych, które obecnie mogą być wykonywane manualnie lub w sposób niespójny. Dzięki implementacji rozwiązania informatycznego, firma zyskuje narzędzie pozwalające na przejrzyste przechowywanie danych, szybką aktualizację informacji oraz sprawne wyszukiwanie potrzebnych rekordów.

Projekt został zrealizowany w oparciu o język programowania C#, wykorzystując nowoczesne podejścia do zarządzania danymi przy użyciu relacyjnej bazy danych SQL Server. Aplikacja konsolowa jest intuicyjna w obsłudze i pozwala na prostą integrację z istniejącymi systemami firmy, co czyni ją wszechstronnym narzędziem dla różnego rodzaju użytkowników – od pracowników administracyjnych po kierowników zarządzających flotą pojazdów.

Dzięki elastyczności systemu możliwe jest również jego dalsze rozwijanie, co pozwala na dostosowanie aplikacji do zmieniających się potrzeb przedsiębiorstwa. System zaprojektowano z myślą o skalowalności, co oznacza, że może on obsługiwać zarówno małe firmy, jak i większe organizacje posiadające rozbudowane floty pojazdów.

2 Opis założeń projektu

2.1 Cele projektu

- Zarządzanie klientami: Umożliwienie dodawania, edycji i usuwania informacji o klientach.
- Zarządzanie pojazdami: Obsługa dodawania, edycji i usuwania pojazdów oraz przypisywanie ich do klientów.
- Integracja z bazą danych: Przechowywanie i zarządzanie danymi w SQL Server.

2.2 Wymagania funkcjonalne

- Dodawanie, edycja i usuwanie danych klientów.
- Dodawanie, edycja i usuwanie pojazdów przypisanych do klientów.
- Wyświetlanie listy klientów i pojazdów z pełnymi informacjami.
- Bezpieczne operacje na bazie danych z użyciem parametrów.

2.3 Wymagania niefunkcjonalne

- Intuicyjny interfejs konsoli.
- Wysoka wydajność operacji na bazie danych.
- Skalowalność pozwalająca na przechowywanie dużej liczby danych klientów i pojazdów.
- Bezpieczeństwo operacji na danych (np. użycie parametrów w zapytaniach SQL).

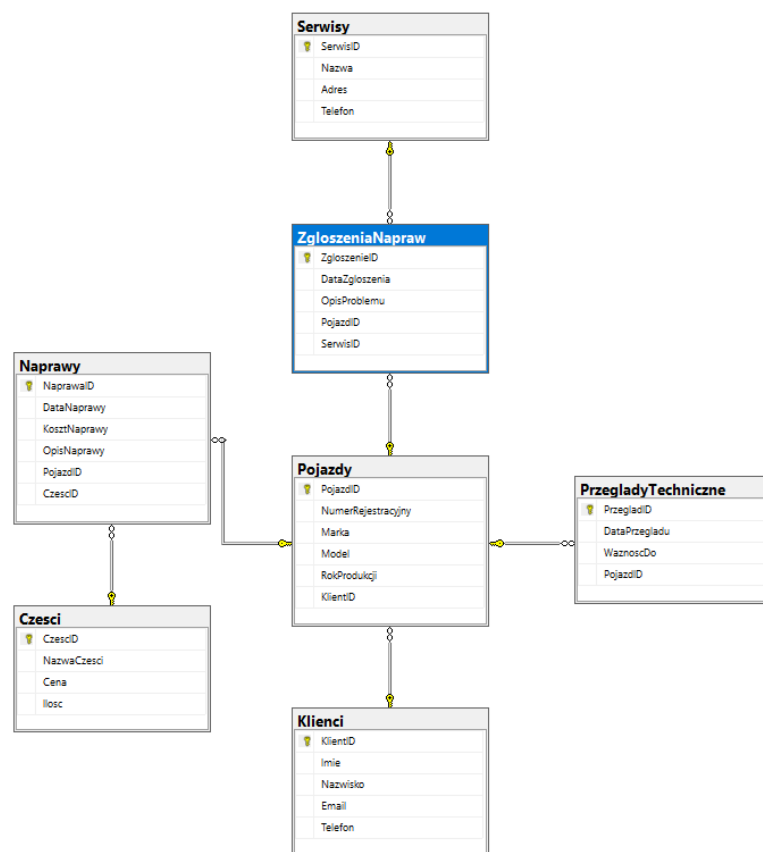
3 Opis struktury projektu

3.1 Technologie i narzędzia

- Język programowania: C#
- Biblioteki: `Microsoft.Data.SqlClient` - obsługa połączeń z bazą danych.
- Baza danych: SQL Server

3.2 Struktura bazy danych

Schemat bazy danych przedstawiony jest na Rysunku. Baza danych zawiera następujące tabele:



Rysunek 1: Schemat bazy danych

- **Klienci**: Przechowuje informacje o klientach, takie jak imię, nazwisko, email i telefon.
- **Pojazdy**: Zawiera dane o pojazdach, takie jak numer rejestracyjny, marka, model, rok produkcji i przypisanie do klienta.
- **Serwisy**: Przechowuje informacje o serwisach, w których realizowane są naprawy.

- **Naprawy:** Przechowuje informacje o przeprowadzonych naprawach, w tym kosztach i użytych częściach.
- **Części:** Zawiera dane o częściach używanych do napraw, takie jak nazwa, cena i ilość.
- **PrzeglądyTechniczne:** Przechowuje dane o przeglądach technicznych, w tym datę przeglądu, ważność oraz identyfikator pojazdu.
- **ZgłoszeniaNapraw:** Przechowuje zgłoszenia napraw, łącząc pojazd, serwis i szczegóły zgłoszenia, takie jak data i opis problemu.

3.3 Struktura programowania

Aplikacja została zaprojektowana w sposób modularny z podziałem na klasy i metody odpowiedzialne za różne funkcje systemu:

- **Klasa DodajKlienta:** Zawiera metody do dodawania nowych klientów do bazy danych.
- **Klasa DodajPojazd:** Odpowiada za dodawanie nowych pojazdów i przypisywanie ich do klientów.
- **Klasa UsunKlienta:** Zawiera funkcje usuwania danych klienta na podstawie jego identyfikatora.
- **Klasa UsunPojazd:** Umożliwia usuwanie pojazdów z bazy danych.
- **Klasa ZmianaDanychKlienta:** Służy do edytowania danych klientów.
- **Klasa WyświetlKlientow:** Wyświetla listę klientów w konsoli.
- **Klasa WyświetlPojazdy:** Wyświetla listę pojazdów w konsoli wraz z ich przypisaniem do klientów.

4 Harmonogram realizacji projektu

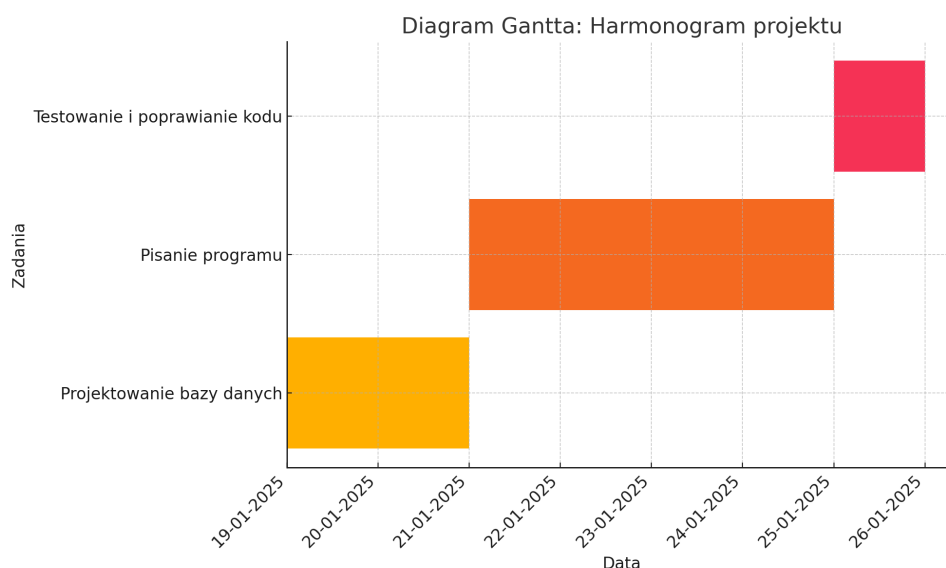
Etapy realizacji:

1. Projektowanie bazy danych: 2 dni (19-20 stycznia 2025).
2. Implementacja modułów dodawania i wyświetlania danych: 4 dni (21-24 stycznia 2025).
3. Testowanie i poprawianie kodu: 1 dzień (25 stycznia 2025).

4.1 Informacje o repozytorium i systemie kontroli wersji

Do zarządzania kodem źródłowym projektu wykorzystano system kontroli wersji **Git** oraz platformę **GitHub** jako zdalne repozytorium.

- **Adres repozytorium:** https://github.com/JGorazd/Projekt_Programowanie



Rysunek 2: Diagram Gantta: Harmonogram projektu

5 Prezentacja warstwy użytkowej projektu

5.1 Funkcjonalności

- Menu główne dostępne w konsoli z opcjami: dodanie klienta, pojazdu, wyświetlanie danych, modyfikacje i usuwanie.
- Każda opcja w menu wywołuje odpowiednią metodę, która prowadzi użytkownika przez kolejne kroki, np. wprowadzanie danych klienta.

5.2 Interakcja użytkownika z systemem

Interfejs użytkownika jest w pełni oparty na konsoli i zaprojektowany w sposób intuicyjny. Po uruchomieniu aplikacji użytkownik jest witany przez menu główne, w którym może wybrać jedną z dostępnych opcji za pomocą wprowadzenia odpowiedniej liczby. Każda operacja w systemie, np. dodanie klienta, wymaga podania konkretnych danych, takich jak imię, nazwisko, adres email i numer telefonu.

Przykładowe użycie programu:

1. Po uruchomieniu aplikacji użytkownik widzi menu główne z listą opcji.
2. Użytkownik wybiera opcję 1. **Dodaj klienta.**
3. System wyświetla komunikaty, prosząc o wprowadzenie kolejnych danych, takich jak imię, nazwisko, email i telefon.
4. Po zakończeniu wprowadzania danych system potwierdza zapisanie klienta w bazie danych.
5. Użytkownik wraca do menu głównego i może wybrać kolejną opcję.

```
C:\Users\user\source\repos\Z  X + v
1. Dodaj klienta
2. Wyświetl klientów
3. Usuń klienta
4. Dodaj pojazd
5. Wyświetl pojazdy
6. Usuń pojazd
7. Zmień dane klienta
8. Wyjście
Wybierz opcję:
```

Rysunek 3: Menu

```
Wybierz opcję: 1
Imię: Kacper
Nazwisko: Kmiec
Email: KKmiec@o2.pl
Telefon: 971710651
1 row(s) inserted.
```

Rysunek 4: Przykład użycia

```
1 row(s) inserted.
1. Dodaj klienta
2. Wyświetl klientów
3. Usuń klienta
4. Dodaj pojazd
5. Wyświetl pojazdy
6. Usuń pojazd
7. Zmień dane klienta
8. Wyjście
Wybierz opcję: |
```

Rysunek 5: Powrót

6 Podsumowanie

Projekt został pomyślnie zrealizowany, a jego wynikiem jest w pełni funkcjonalna aplikacja konsolowa, umożliwiająca skuteczne zarządzanie klientami oraz pojazdami. System spełnia wszystkie założone wymagania funkcjonalne i нефункционалне, co czyni go wydajnym i intuicyjnym narzędziem do obsługi flot pojazdów.

Możliwe dalsze kierunki rozwoju projektu obejmują:

- Dodanie graficznego interfejsu użytkownika, co pozwoli na łatwiejszą obsługę dla użytkowników bez doświadczenia z aplikacjami konsolowymi.
- Rozszerzenie o zaawansowane funkcje raportowania, które mogą pomóc w analizie danych klientów i pojazdów.

- Integracja z zewnętrznymi systemami ERP, co pozwoli na lepsze zarządzanie procesami biznesowymi.
- Dodanie funkcji powiadomień o zbliżających się terminach przeglądów technicznych lub innych ważnych zdarzeń.
- Wprowadzenie mechanizmów backupu i przywracania danych, aby zwiększyć bezpieczeństwo systemu.

Literatura

- [1] <https://learn.microsoft.com/pl-pl/visualstudio/data-tools/save-data-from-an-object-to-a-database?view=vs-2022&tabs=csharp>, z dnia 26.01.2025
- [2] <https://learn.microsoft.com/pl-pl/dotnet/api/system.data.sqlclient.sqldatareader.read?view=net-8.0-pp>, z dnia 26.01.2025

Spis rysunków

| | |
|--|---|
| 1. Schemat bazy danych | 3 |
| 2. Diagram Gantt: Harmonogram projektu | 5 |
| 3. Menu główne aplikacji | 6 |
| 4. Przykład komunikatów aplikacji | 6 |
| 5. Powrót do menu głównego | 6 |