

# System zarządzający stacją paliw

---

Zespół:

1. Jakub Mroczkowski
2. Paweł Bąk
3. Kamil Mierzwiński

## Spis treści

1.	Wstęp .....	3
2.	Analiza wymagań systemu .....	3
2.1	Wymagania funkcjonalne .....	3
2.2	Wymagania niefunkcjonalne .....	3
2.3	Diagram przypadków użycia .....	4
3.	Wykorzystane technologie .....	4
4.	Projekt aplikacji .....	4
4.1	Architektura aplikacji .....	4
4.2	Projekt koncepcyjny bazy danych .....	4
4.3	Projekt schematu relacyjnego.....	5
4.4	Mapowanie klas na tabele bazodanowe .....	7
5.	Funkcjonalność aplikacji.....	7
6.	Interfejs użytkownika .....	7
7.	Podsumowanie.....	7
	Dodatek A: Skrypty tworzące obiekty baz danych .....	7

## **1. Wstęp**

Celem budowy systemu jest zautomatyzowanie stacji paliw. Aplikacja ma umożliwić: zarządzanie transakcjami oraz kartami promocyjnymi, wprowadzanie dostaw produktów oraz generowanie raportów.

## **2. Analiza wymagań systemu**

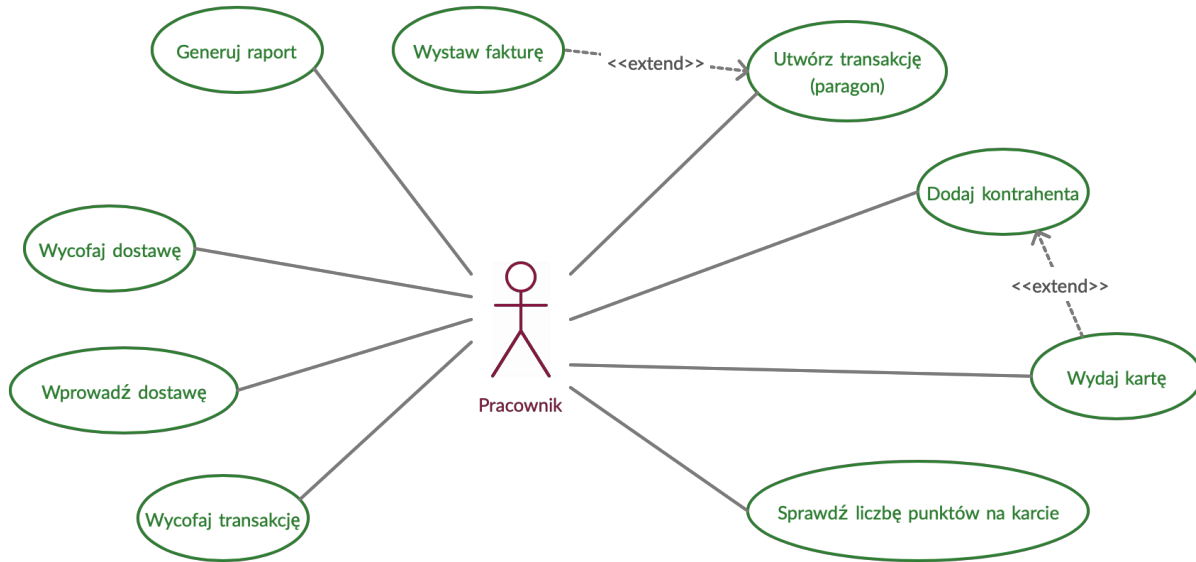
### **2.1 Wymagania funkcjonalne**

- Generowanie raportów
- Utworzenie transakcji
- Wycofanie transakcji
- Wystawianie faktury
- Dodanie kontrahenta
- Wydanie karty
- Sprawdzenie liczby punktów na danej karcie
- Wprowadzenie dostawy
- Wycofanie dostawy

### **2.2 Wymagania нефunkcjonalne**

- Czas odpowiedzi systemu dla generowania raportów:
  - Poniżej 10 sekund
- Czas odpowiedzi systemu w przypadku pozostałych aktywności:
  - Poniżej 5 sekund
- Czas na zasygnalizowanie przez system przyjęcia akcji użytkownika:
  - Poniżej 100 ms
- Program pracuje na systemie Windows 10
- Program posiada graficzny interfejs
- Do pełnej pracy systemu potrzebne są zewnętrzne czujniki (napełnienie zbiornika, praca dystrybutora)

## 2.3 Diagram przypadków użycia



## 3. Wykorzystane technologie

Framework ASP.NET Core wersja 3.0.0 + Entity Framework Core wersja 3.0.0

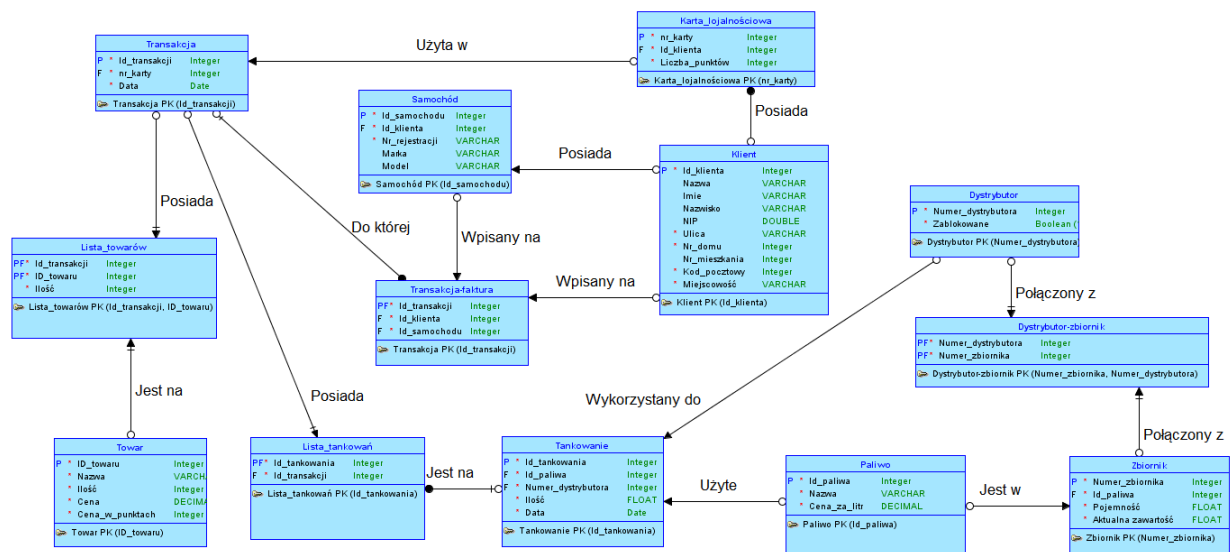
Baza danych:

System zarządzania: MS SQL, Hosting: GearHost

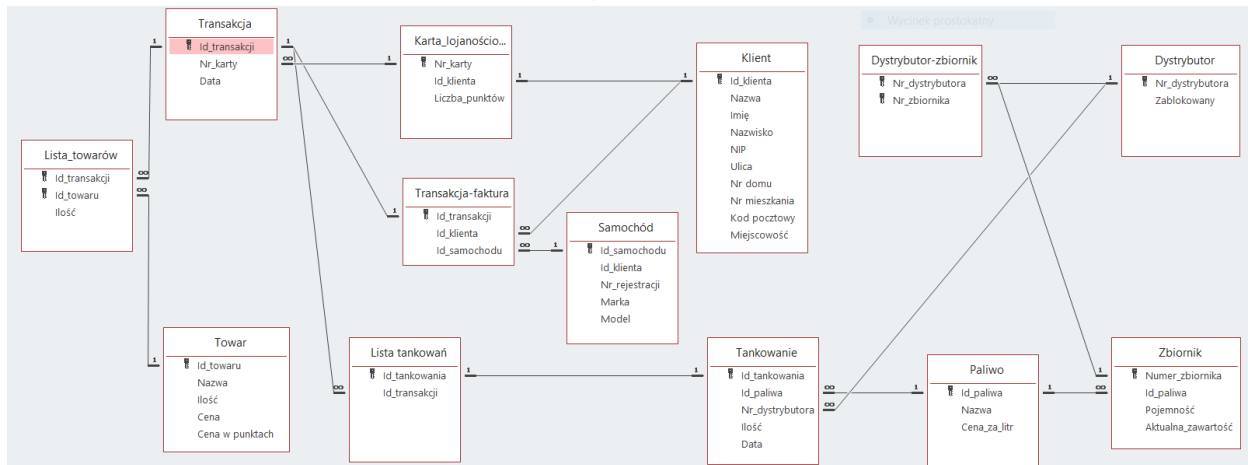
## 4. Projekt aplikacji

### 4.1 Architektura aplikacji

### 4.2 Projekt koncepcyjny bazy danych



### 4.3 Projekt schematu relacyjnego



#### Opis tabel:

1. **Towar** – zawiera informacje na temat danego produktu sprzedawanego na stacji paliw
  - a. **Id\_towaru** – unikatowy identyfikator
  - b. **Nazwa** – tytuł produktu
  - c. **Ilość na stanie** – aktualna liczebność towaru w magazynie
  - d. **Cena** – opłata za sztukę produktu wyrażona w PLN
  - e. **Cena w punktach** – opłata za sztukę produktu wyrażona w punktach znajdujących się na karcie lojalnościowej
2. **Lista\_towarów** – zawiera informacje o zakupionych towarach w transakcji
  - a. **Id\_transaksi** – identyfikator określający transakcję
  - b. **Id\_towaru** – identyfikator określający towar
  - c. **Ilość** – liczebność wybranego towaru w danej transakcji
3. **Transakcja** – zawiera informacje o transakcji
  - a. **Id\_transaksi** – unikatowy identyfikator
  - b. **Id\_karty** – identyfikator określający kartę użytą podczas wykonywania transakcji
  - c. **Data** – data transakcji
4. **Lista-tankowań** – łączy informacje o danym tankowaniu z konkretną transakcją
  - a. **Id\_tankowania** – unikatowy identyfikator
  - b. **Id\_transaksi** – identyfikator określający transakcję
5. **Tankowanie** - tabela zawierająca informacje o pojedynczych tankowaniach (jedno tankowanie z jednego dystrybutora):
  - a. **Id\_tankowania** - pole uzupełniane automatycznie, identyfikuje pojedynczy rekord
  - b. **Ilość** - pole zawierające ilość paliwa składającą się na pojedyncze tankowanie w ustalonej jednostce
  - c. **Data** - data tankowania
  - d. **Id\_paliwa** - pole wskazujące na tabelę Paliwo i określające jakie paliwo zatankowano
  - e. **Nr dystrybutora** - pole wskazujące na tabelę Dystrybutor i określające, z którego dystrybutora tankowano

6. Paliwo - tabela zawierająca informacje o różnych rodzajach paliwa:
  - a. Id\_paliwa - pole uzupełniane automatycznie, identyfikuje pojedynczy rekord
  - b. Nazwa - pole zawierające nazwę rodzaju paliwa, którą posługują się pracownicy stacji
  - c. Cena za litr - pole zawierające cenę paliwa za litr w złotych polskich
7. Dystrybutor - tabela zawierająca informacje o poszczególnych dystrybutorach na obsługiwanej stacji:
  - a. Nr\_dystrybutora - pole jednoznacznie identyfikujące dystrybutor na stacji
  - b. Zablokowany - pole zawierające informację o tym czy dystrybutor jest aktualnie dostępny
8. Dystrybutor-zbiornik - tabela łącząca dystrybutory ze zbiornikami z paliwem, do których są podłączone:
  - a. Nr\_dystrybutora - pole wskazujące na tabelę Dystrybutor i przechowujące numer jednego dystrybutora
  - b. Nr\_zbiornika - pole wskazujące na tabelę Zbiornik i przechowujące numer jednego zbiornika
9. Zbiornik - tabela zawierająca informacje o zbiornikach z paliwem:
  - a. Nr\_zbiornika - pole jednoznacznie identyfikujące zbiornik z paliwem na stacji
  - b. Pojemność - pojemność zbiornika w ustalonej jednostce
  - c. Aktualna zawartość - aktualne zapełnienie zbiornika w ustalonej jednostce
  - d. Id\_paliwa - pole wskazujące na tabelę Paliwo i określające jakie paliwo znajduje się w zbiorniku
10. Klient- Przechowuje informacje o klientach stacji paliw, którzy mogą posiadać kartę lojalnościową lub biorą zakupy na fakturę:
  - a. Id\_klienta - klucz główny, unikalny identyfikator klienta (typ: liczba całkowita)
  - b. Nazwa - nazwa firmy, jeśli klientem jest osoba prywatna, to pole jest null (typ: napis)
  - c. Imie - Imie klienta, wypełnione jeśli klient jest osobą prywatną (typ: napis)
  - d. Nazwisko - nazwisko klienta, wypełnione jeśli klient jest osobą prywatną (typ: napis)
  - e. NIP - numer NIP klienta, jeśli klient nie posiada- NULL, typ (liczba)
  - f. Ulica, nr domu, nr mieszkania, kod pocztowy, miejscowość - dane adresowe klienta (typ: napis)
11. Transakcja-faktura - przechowuje transakcje do których została wystawiona faktura:
  - a. id\_transakcji - klucz główny, identyfikator transakcji (typ: liczba całkowita)
  - b. id\_klienta - identyfikator klienta, dla którego została wystawiona faktura.(typ: liczba całkowita)
  - c. Data - przechowuje datę transakcji (typ:date)
  - d. id\_samochodu - jeśli na fakturze było paliwo, w tym miejscu może znaleźć się id pojazdu który był tankowany
12. Samochód - przechowuje samochody, które są przypisane do klientów:
  - a. Id\_samochodu - klucz główny, identyfikator samochodu (typ: liczba całkowita)
  - b. id\_klienta - identyfikator klienta do którego należy samochód (typ: liczba całkowita)
  - c. nr\_rejestracji - numer rejestracyjny samochodu (typ: napis)
  - d. marka - marka samochodu (typ: napis)
  - e. model - model samochodu (typ: napis)
13. Karta lojalnościowa - przechowuje karty lojalnościowe przypisane do klientów:
  - a. Nr\_karty - klucz główny, identyfikator karty (typ: liczba całkowita)

- b. id\_klienta - identyfikator klienta, do którego należy karta (typ: liczba całkowita)
- c. liczba\_punktów - ilość punktów zebranych na karcie (typ: liczba całkowita)

#### 4.4 Mapowanie klas na tabele bazodanowe

### 5. Funkcjonalność aplikacji

### 6. Interfejs użytkownika

### 7. Podsumowanie

## Dodatek A: Skrypty tworzące obiekty baz danych

```
CREATE TABLE [Client] (  
    [IdClient] int NOT NULL IDENTITY,  
    [Name] nvarchar(max) NULL,  
    [FirstName] nvarchar(max) NULL,  
    [Surname] nvarchar(max) NULL,  
    [NIP] nvarchar(max) NULL,  
    [Street] nvarchar(max) NOT NULL,  
    [HouseNumber] nvarchar NOT NULL,  
    [ApartmentNumber] int NULL,  
    [Postcode] nvarchar(max) NOT NULL,  
    [Locality] nvarchar(max) NOT NULL,  
    CONSTRAINT [PK_Client] PRIMARY KEY ([IdClient])  
);  
  
CREATE TABLE [Fuel] (  
    [IdFuel] int NOT NULL IDENTITY,  
    [Name] nvarchar(max) NOT NULL,  
    [PriceForLiter] decimal(18,2) NOT NULL,  
    CONSTRAINT [PK_Fuel] PRIMARY KEY ([IdFuel])  
);  
  
CREATE TABLE [GasPump] (  
    [IdGasPump] int NOT NULL IDENTITY,  
    [Locked] bit NOT NULL,  
    CONSTRAINT [PK_GasPump] PRIMARY KEY ([IdGasPump])  
);
```

```

CREATE TABLE [Product] (
    [IdProduct] int NOT NULL IDENTITY,
    [Name] nvarchar(max) NOT NULL,
    [QuantityInStorage] int NOT NULL,
    [Price] decimal(18,2) NOT NULL,
    [PriceInPoints] int NOT NULL,
    CONSTRAINT [PK_Product] PRIMARY KEY ([IdProduct])
);

CREATE TABLE [Car] (
    [IdCar] int NOT NULL IDENTITY,
    [IdClient] int NOT NULL,
    [NumberPlate] nvarchar(max) NOT NULL,
    [CarBrand] nvarchar(max) NULL,
    [CarModel] nvarchar(max) NULL,
    CONSTRAINT [PK_Car] PRIMARY KEY ([IdCar]),
    CONSTRAINT [FK_Car_Client_IdClient] FOREIGN KEY ([IdClient]) REFERENCES
[Client] ([IdClient]) ON DELETE CASCADE
);

CREATE TABLE [LoyaltyCard] (
    [IdLoyaltyCard] int NOT NULL IDENTITY,
    [IdClient] int NOT NULL,
    [ActualPoints] int NOT NULL,
    CONSTRAINT [PK_LoyaltyCard] PRIMARY KEY ([IdLoyaltyCard]),
    CONSTRAINT [FK_LoyaltyCard_Client_IdClient] FOREIGN KEY ([IdClient])
REFERENCES [Client] ([IdClient]) ON DELETE CASCADE
);

CREATE TABLE [FuelTank] (
    [IdFuelTank] int NOT NULL IDENTITY,
    [IdFuel] int NOT NULL,
    [Capacity] real NOT NULL,
    [ActualQuantity] real NOT NULL,
    CONSTRAINT [PK_FuelTank] PRIMARY KEY ([IdFuelTank]),
    CONSTRAINT [FK_FuelTank_Fuel_IdFuel] FOREIGN KEY ([IdFuel]) REFERENCES
[Fuel] ([IdFuel]) ON DELETE CASCADE
);

CREATE TABLE [Fueling] (
    [IdFueling] int NOT NULL IDENTITY,
    [IdFuel] int NOT NULL,
    [IdGasPump] int NOT NULL,
    [Quantity] real NOT NULL,
    [Date] datetime2 NOT NULL,
    CONSTRAINT [PK_Fueling] PRIMARY KEY ([IdFueling]),
    CONSTRAINT [FK_Fueling_Fuel_IdFuel] FOREIGN KEY ([IdFuel]) REFERENCES
[Fuel] ([IdFuel]) ON DELETE CASCADE,
    CONSTRAINT [FK_Fueling_GasPump_IdGasPump] FOREIGN KEY ([IdGasPump])
REFERENCES [GasPump] ([IdGasPump]) ON DELETE CASCADE
);

```



```

CREATE TABLE [Transaction] (
    [IdTransaction] int NOT NULL IDENTITY,
    [IdLoyaltyCard] int NULL,
    [Date] datetime2 NOT NULL,
    [Discriminator] nvarchar(max) NOT NULL,
    [IdClient] int NULL,
    [IdCar] int NULL,
    CONSTRAINT [PK_Transaction] PRIMARY KEY ([IdTransaction]),
    CONSTRAINT [FK_Transaction_LoyaltyCard_IdLoyaltyCard] FOREIGN KEY
([IdLoyaltyCard]) REFERENCES [LoyaltyCard] ([IdLoyaltyCard]) ON DELETE
NO ACTION,
    CONSTRAINT [FK_Transaction_Car_IdCar] FOREIGN KEY ([IdCar]) REFERENCES
[Car] ([IdCar]) ON DELETE NO ACTION,
    CONSTRAINT [FK_Transaction_Client_IdClient] FOREIGN KEY ([IdClient])
REFERENCES [Client] ([IdClient]) ON DELETE NO ACTION
);

```

```

CREATE TABLE [PumpTank] (
    [IdGasPump] int NOT NULL,
    [IdFuelTank] int NOT NULL,
    CONSTRAINT [PK_PumpTank] PRIMARY KEY ([IdGasPump], [IdFuelTank]),
    CONSTRAINT [FK_PumpTank_FuelTank_IdFuelTank] FOREIGN KEY
([IdFuelTank]) REFERENCES [FuelTank] ([IdFuelTank]) ON DELETE CASCADE,
    CONSTRAINT [FK_PumpTank_GasPump_IdGasPump] FOREIGN KEY ([IdGasPump])
REFERENCES [GasPump] ([IdGasPump]) ON DELETE CASCADE
);

```

```

CREATE TABLE [FuelingList] (
    [IdFueling] int NOT NULL,
    [IdTransaction] int NOT NULL,
    CONSTRAINT [PK_FuelingList] PRIMARY KEY ([IdFueling]),
    CONSTRAINT [FK_FuelingList_Fueling_IdFueling] FOREIGN KEY
([IdFueling]) REFERENCES [Fueling] ([IdFueling]) ON DELETE CASCADE,
    CONSTRAINT [FK_FuelingList_Transaction_IdTransaction] FOREIGN KEY
([IdTransaction]) REFERENCES [Transaction] ([IdTransaction]) ON DELETE
CASCADE
);

```

```

CREATE TABLE [ProductList] (
    [IdTransaction] int NOT NULL,
    [IdProduct] int NOT NULL,
    [Quantity] int NOT NULL,
    CONSTRAINT [PK_ProductList] PRIMARY KEY ([IdProduct],
[IdTransaction]),
    CONSTRAINT [FK_ProductList_Product_IdProduct] FOREIGN KEY
([IdProduct]) REFERENCES [Product] ([IdProduct]) ON DELETE CASCADE,
    CONSTRAINT [FK_ProductList_Transaction_IdTransaction] FOREIGN KEY
([IdTransaction]) REFERENCES [Transaction] ([IdTransaction]) ON DELETE
CASCADE
);

```