Projekt "GASSTATION"

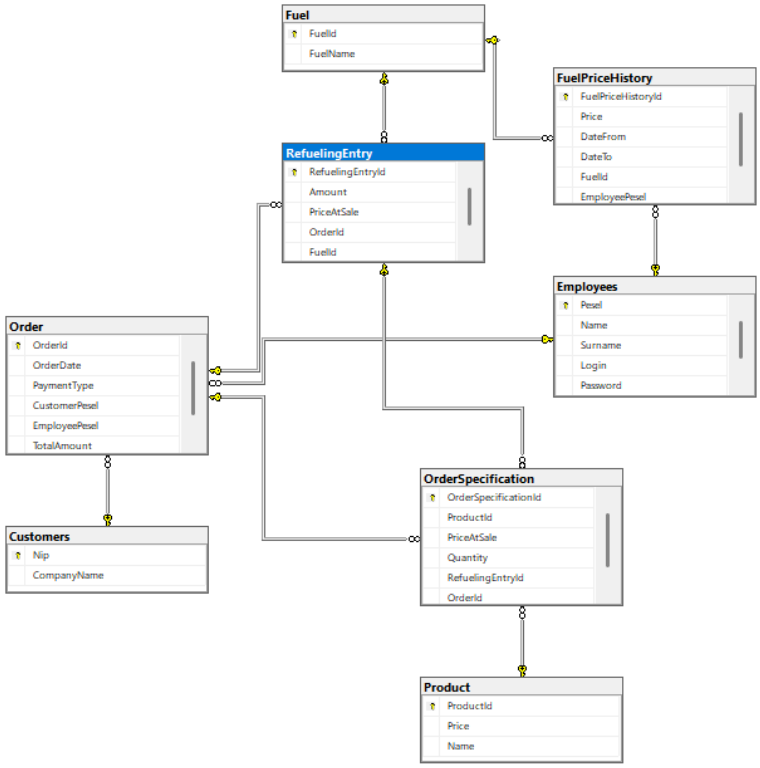
Kacper Łapot,
Jakub Kaźmierczyk
Rafał Łubkowski

Cel projektu:

Wykonanie aplikacji ASP.NET, która będzie symulowała aplikację używaną przez kasjerów stacji benzynowej do zmieniania cen i dodawania artykułów. Aplikacja po udoskonaleniach mogłaby zostać wykorzystana jako narzędzie dla stacji benzynowej.

Model Bazy Danych

Model bazy danych został zaprojektowany do śledzenia sprzedaży paliw i innych produktów, rejestrowania historii cen paliw, powiązania transakcji z konkretnymi zamówieniami, klientami i pracownikami oraz przechowywania szczegółów dotyczących poszczególnych pozycji w zamówieniu.



Tabele:

- **Fuel:** Przechowuje informacje o rodzajach paliw.
    - **FuelId:** Klucz główny, identyfikator paliwa.
    - **FuelName:** Nazwa paliwa.

- **FuelPriceHistory:** Przechowuje historię cen paliw.
    - **FuelPriceHistoryId:** Klucz główny.
    - **Price:** Cena paliwa.
    - **DateFrom:** Data obowiązywania ceny od.
    - **DateTo:** Data obowiązywania ceny do.
    - **FuelId:** Klucz obcy do tabeli Fuel, wskazujący, którego paliwa dotyczy cena.
- **RefuelingEntry:** Reprezentuje pojedyncze tankowanie.
    - **RefuelingEntryId:** Klucz główny, identyfikator tankowania.
    - **Amount:** Ilość zatankowanego paliwa.
    - **PriceAtSale:** Cena paliwa w momencie sprzedaży.
    - **OrderId:** Klucz obcy do tabeli Order, łączący tankowanie z zamówieniem (paragonem).
    - **FuelId:** Klucz obcy do tabeli Fuel, wskazujący, które paliwo zostało zatankowane.
- **Employees:** Przechowuje dane pracowników.
    - **Pesel:** Klucz główny, numer PESEL pracownika.
    - **Name:** Imię pracownika.
    - **Surname:** Nazwisko pracownika.
    - **Login:** Login pracownika do systemu.
    - **Password:** Hasło pracownika do systemu.
- **Order:** Reprezentuje pojedyncze zamówienie (paragon).
    - **OrderId:** Klucz główny, identyfikator zamówienia.
    - **OrderDate:** Data i czas zamówienia.
    - **PaymentType:** Typ płatności.
    - **CustomerId:** Klucz obcy do tabeli Customers, wskazujący klienta (jeśli dotyczy).
    - **EmployeePesel:** Klucz obcy do tabeli Employees, wskazujący pracownika obsługującego zamówienie.
    - **TotalAmount:** Całkowita kwota zamówienia.
- **Customers:** Przechowuje dane klientów.

    - **Nip:** Klucz główny, numer NIP klienta (sugeruje klientów biznesowych).
    - **CompanyName:** Nazwa firmy klienta.
- **OrderSpecification:** Szczegóły zamówienia, w tym sprzedane produkty i tankowania.
    - **OrderSpecificationId:** Klucz główny.
    - **ProductId:** Klucz obcy do tabeli Product, wskazujący sprzedany produkt (jeśli dotyczy).
    - **PriceAtSale:** Cena produktu w momencie sprzedaży.
    - **Quantity:** Ilość sprzedanego produktu.
    - **RefuelingEntryId:** Klucz obcy do tabeli RefuelingEntry, łączący szczegół zamówienia z konkretnym tankowaniem (jeśli dotyczy).
    - **OrderId:** Klucz obcy do tabeli Order, do którego należy ten szczegół.
- **Product:** Przechowuje informacje o produktach innych niż paliwo.
    - **ProductId:** Klucz główny, identyfikator produktu.
    - **Price:** Cena produktu.
    - **Name:** Nazwa produktu.

Relacje:

Diagram przedstawia następujące relacje między tabelami:

- **Fuel** 1-do-wielu **FuelPriceHistory:** Jedno paliwo może mieć wiele historycznych cen.
- **Fuel** 1-do-wielu **RefuelingEntry:** Jedno paliwo może być przedmiotem wielu tankowań.
- **RefuelingEntry** 1-do-jednego **OrderSpecification:** Każde tankowanie jest jednym elementem specyfikacji zamówienia.
- **Order** 1-do-wielu **RefuelingEntry:** Jedno zamówienie może obejmować wiele tankowań.
- **Order** 1-do-wielu **OrderSpecification:** Jedno zamówienie może zawierać wiele pozycji szczegółowych (produktów lub tankowań).
- **Customers** 1-do-wielu **Order:** Jeden klient może złożyć wiele zamówień.
- **Employees** 1-do-wielu **Order:** Jeden pracownik może obsłużyć wiele zamówień.

- **Product** 1-do-wielu **OrderSpecification:** Jeden produkt może pojawić się w wielu pozycjach szczegółowych zamówień.

Kontrolery

- **HomeController:** Służy do uruchomienia aplikacji oraz obsługi logowania.
- **CashierController:** Odpowiada za widok kasjera i logikę związaną z obsługą transakcji przez kasjera.
- **EmployeeController:** Odpowiada za zarządzanie danymi pracowników. Zawiera akcje do logowania, pobierania listy pracowników oraz szczegółów pojedynczych pracowników.
- **FuelController:** Zarządza informacjami o paliwach oraz ich cenach. Umożliwia pobieranie aktualnych cen, całej historii cen, a także dodawanie i usuwanie wpisów paliw i historii cen.
- **OrderController:** Zajmuje się zarządzaniem zamówieniami (transakcjami). Służy do wyświetlania listy zamówień, przeglądania ich szczegółów oraz tworzenia i usuwania nowych zamówień.
- **ProductController:** Jest odpowiedzialny za obsługę produktów innych niż paliwa. Udostępnia funkcje do pobierania listy produktów, przeglądania ich szczegółów, a także dodawania, edytowania i usuwania produktów.
- **CustomerController:** Służy do zarządzania danymi klientów. Pozwala na pobieranie listy klientów, wyświetlanie szczegółów konkretnego klienta oraz dodawanie i usuwanie klientów z systemu.

Widoki

- **Index:** Strona logowania dla systemu, zbudowana przy użyciu technologii ASP.NET MVC (Razor). Wyświetla formularz umożliwiający użytkownikowi (pracownikowi) wprowadzenie loginu i hasła. Używa modelu EmployeeLoginDTO do wiązania danych i walidacji. Formularz jest wysyłany metodą POST do akcji Login w kontrolerze Home.
- **Cashier_View:** Reprezentuje główny interfejs kasjera systemu stacji benzynowej, służący do obsługi transakcji. Wykorzystuje model CreateOrderDTO oraz dane przekazane przez ViewBag (lista tankowań, aktualne ceny paliw, dane zalogowanego pracownika).

Widoki Częściowe (Partial Views)

- **Modal dodawania klienta (addCustomerModal):** Wyskakujące okno (modal Bootstrapa) służące do dodawania nowego klienta do systemu. Zawiera formularz renderowany z widoku częściowego _CreateCustomerForm.cshtml.

- **Modal dodawania produktu (addProductModal):** Wyskakujące okno (modal Bootstrapa) przeznaczone do dodawania nowego produktu. Zawiera formularz renderowany z widoku częściowego _CreateProductForm.cshtml.
- **Modal zmiany ceny paliwa (changeFuelPriceModal):** Modal Bootstrapa umożliwiający kasjerowi zmianę aktualnej ceny wybranego rodzaju paliwa. Składa się z formularza z polem wyboru rodzaju paliwa i polem do wprowadzenia nowej ceny.
- **Modal zamówienia:** Modal (wyskakujące okno) odpowiedzialne za proces finalizacji płatności za zamówienie. Zawiera formularz z polami do wyboru klienta (opcjonalnie), pracownika obsługującego transakcję oraz metody płatności. Wyświetla sumę do zapłaty i przycisk potwierdzający finalizację, obsługiwaną asynchronicznie za pomocą JavaScriptu.

Instrukcja obsługi i Podział Pracy

Instrukcja obsługi

Aby uruchomić i korzystać z aplikacji GASSTATION, wykonaj następujące kroki:

1. **Inicjalizacja bazy danych:**
    - Po pobraniu projektu otwórz Konsolę Menedżera Pakietów (Package Manager Console) w Visual Studio.
    - Wpisz komendę: Update-Database i naciśnij Enter. Spowoduje to wygenerowanie początkowej zawartości bazy danych (seed).
2. **Logowanie:**
    - Po uruchomieniu aplikacji zobaczysz stronę logowania.
    - Zaloguj się przy użyciu jednego z predefiniowanych kont użytkowników:
        - **Login:** admin | **Hasło:** SecurePassword123
        - **Login:** janusz | **Hasło:** Password456
        - **Login:** alicja | **Hasło:** AnotherPassword789
3. **Obsługa zamówienia:**
    - Po zalogowaniu znajdziesz się na widoku kasjera.
    - Możesz obsługiwać zamówienie, dodając różne produkty i/lub instancje tankowania.
4. **Dostępne funkcje:**
    - Program pozwala na **zmianę ceny paliwa**.

    - Możesz **dodać nowy produkt**.
    - Aplikacja umożliwia **finalizację płatności** za zamówienie.
5. **Symulacja tankowania:**
    - Przycisk **"Losuj tankowania"** pozwala na ciągłe losowanie instancji tankowania. Jest to symulacja podjeżdżających i tankujących samochodów.
6. **Wylogowanie:**
    - Przycisk **"Wyloguj"** pozwala użytkownikowi bezpiecznie wylogować się z systemu i wyczyścić ciasteczka sesji.

Podział Pracy

Projekt został zrealizowany przy następującym podziale zadań:

- **Rafał Łubkowski:** Odpowiedzialny za tworzenie modeli oraz schematu bazy danych (Modele, DTO, DbContext).
- **Jakub Kaźmierczyk:** Odpowiedzialny za widoki i warstwę Frontendową aplikacji.
- **Kacper Łapot:** Odpowiedzialny za połączenie backendu z frontendem za pomocą Service'ów i Controllerów.