

Sprawozdanie

z przedmiotu:

Programowanie - wykład

Temat:

„Aplikacja pozwalająca na spis cen paliw”

Wykonał:

Bartosz Wilk, w59004

Prowadzący:

dr Marek Jaszuk

2019

# Cel projektu

Celem projektu jest wykonanie aplikacji pozwalającej na spis cen paliw na poszczególnych stacjach, co pozwoli użytkownikom na wybranie najbardziej im odpowiadającej opcji.

Do wykonania aplikacji użyłem technologii ASP.NET Core 2.1 (C#), oraz MSSQL.

# Założenia funkcjonalne

1. Dodanie nowej stacji paliw
2. Aktualizacja (zmiana ceny), oraz usunięcie paliwa z danej stacji
3. Przypisanie paliwa do stacji paliw
4. Aktualizacja ceny paliwa na stacji paliw
5. Wypisanie wszystkich stacji paliw
6. Wypisanie informacji o danej stacji paliw

# Założenia niefunkcjonalne

1. Możliwość wykorzystania aplikacji bez potrzeby instalacji

# Opis projektu

Projekt podzieliłem na dwa podprojekty: Gasoline.Api, oraz Gasoline.Data. Pierwszy służy do wykonywania logiki związanej z wykonywaniem zapytań oraz komunikacją z serwisem. Znajdują się w nim dwa kontrolery pozwalające na rozdzielenie funkcjonalności aplikacji. Zaimplementowałem w nim również interfejs użytkownika, tworząc kontroller *HomeController.cs.*

Gasoline.Data służy natomiast do komunikacji z bazą danych, oraz zarządzania nią. Umieszczone zostały tam modele przechowywane w bazie danych, serwisy służące do zarządzania bazą, oraz kontekst bazy, który służy do konfiguracji serwera, ustawienia kluczy głównych, oraz wprowadzenia podstawowych typów paliw.

Opis poszczególnych klas:

* GasolineContext.cs – konfiguracja bazy danych, tutaj należy podać serwer SQL.  
  Pole *public IQueryable<TEntity> Table<Tentity>()* pozwala na wykorzystanie w bazie dowolnego obiektu bez konieczności tworzenia nowych zmiennych w opisywanej klasie. W metodzie *OnModelCreating* dodałem ustawianie kluczy głównych, oraz dodanie podstawowych typów paliw – zostanie ona wywołana przy tworzeniu migracji bazy.
* Modele FuelType.cs, GasStation.cs, GasStationFuel.cs – są one wykorzystane do określenia jakie pola posiadają dane obiekty, oraz do utworzenia tabel z bazie danych.
* Serwisy
  + FuelTypeService.cs – służy da zarządzania typami paliw
  + GasStationService.cs – służy do zarządzania bazą stacji paliw, pozwala na wyświetlenie wszystkich stacji, dodanie nowej, dodanie paliwa do stacji, aktualizacja stacji (np. zmiana adresu, lub nazwy), usunięcie stacji, aktualizacja ceny paliwa na danej stacji, oraz usunięcie paliwa ze stacji.

# Opis techniczny aplikacji

Aplikacja działa jako serwis API, dzięki czemu dane z niej mogą zostać wykorzystane na większości urządzeń z możliwością komunikacji z siecią. Pozwala ona wykonanie zapytań:

"fuelTypes/"

GET ""- *wszystkie możliwe paliwa*

GET "name/{name} - *wyszukuje paliwa o podanej nazwie np  
Benzyna=>{Benzyna 95, Benzyna 98}*

POST ""

body:

{fuelName}

- *dodanie nowego paliwa*

DELETE "{id}" - *usunięcie paliwa*

"GasStation/"

GET "" - *zwraca wszystkie stacje paliw wraz w oferowanymi paliwami*

GET "gasStationsFuels" - *zwraca wszystkie paliwa na stacjach*

GET "gasStations/{id}" - *zwraca stację o podanym id*

POST ""

body:

{

Name

City

Street

PostalCode

}

- *dodanie nowej stacji*

PUT "{id}"

body:

{

Name

City

Street

PostalCode

}

- *aktualizacja stacji o podanym ID*

DELETE "{id}" - *usunięcie stacji z danym ID*

POST "{id}/fuels"

body:

{

fuelTypeId

Price

}

- *dodanie paliwa {fuelTypeId} do stacji {id}*

PUT "{stationId}/fuels/{fueldId}"

body:

{

Price

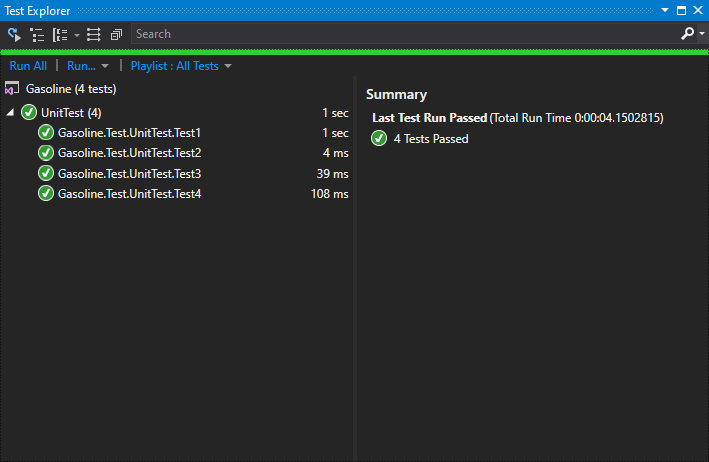
}

- *aktualizacja paliwa {fueldId} na stacji {stationId*}

DELETE "{stationId}/fuels/{fuelId}"

- *usunięcie paliwa {fuelId} ze stacji {stationId}*

# Testy



Utworzone cztery testy pozwoliły na sprawdzenie:

* + Test1 – dodanie nowego typu paliwa
  + Test2 – pobranie wszystkich typów paliwa
  + Test3 – pobranie tylko typów paliwa, które w nazwie posiadają “Benzyna”
  + Test4 – usunięcie paliwa o Guid = “74d5aaf9-f5f1-45fe-bc79-1ab94eddb806”

# Interfejs użytkownika

Ze względu, że utworzona aplikacja utworzona została w środowisku ASP.NET Core, jej interfejs obsługiwany jest z poziomu przeglądarki internetowej.

#### Widok startowy

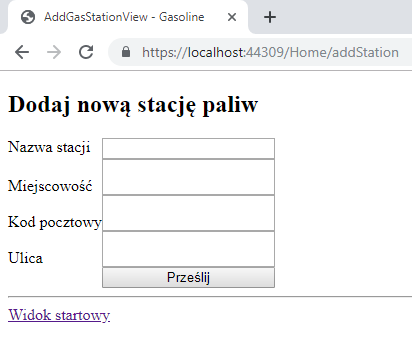
Widoczne są trzy odnośniki

* Dodanie nowej stacji paliw
* Wyświetlenie już dodanych stacji paliw
* Wyświetlenie typów paliw



#### Dodanie nowej stacji

Formularz pozwalający na wprowadzenie danych na temat dodawanej stacji paliw.

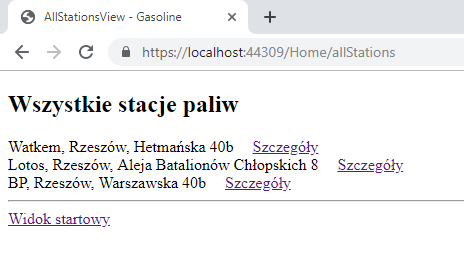


Po dodaniu paliwa zostajemy przekierowani do widoku szczegółów stacji, gdzie możemy usunąć stację, oraz dodawać i usuwać rodzaje paliw, oraz aktualizować ich ceny.



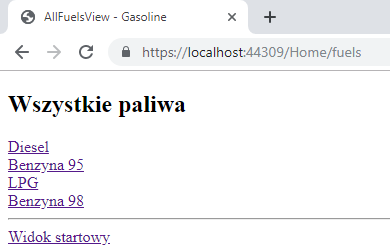
#### Wyświetlenie wszystkich stacji

Pozwala na wyświetlenie wszystkich stacji paliw, oraz przejście do szczegółów każdej z nich.



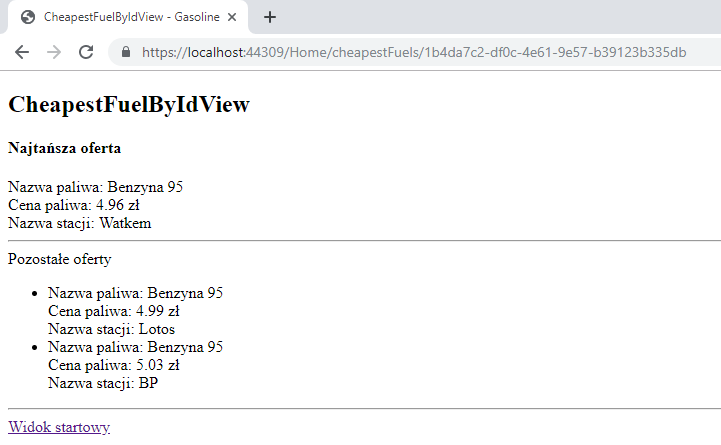
#### Wyświetlenie paliw

Wyświetla wszystkie możliwe paliwa.

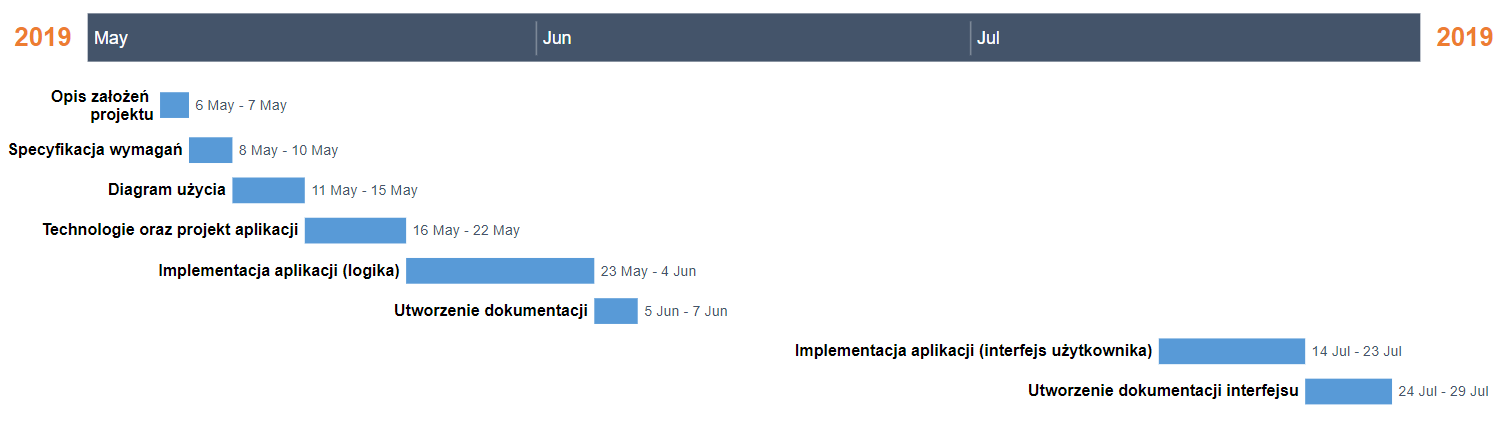


#### Widok paliwa

Pozwala na sprawdzenie wszystkich stacji paliw z dostępnym danym paliwem, posortowane według ceny.



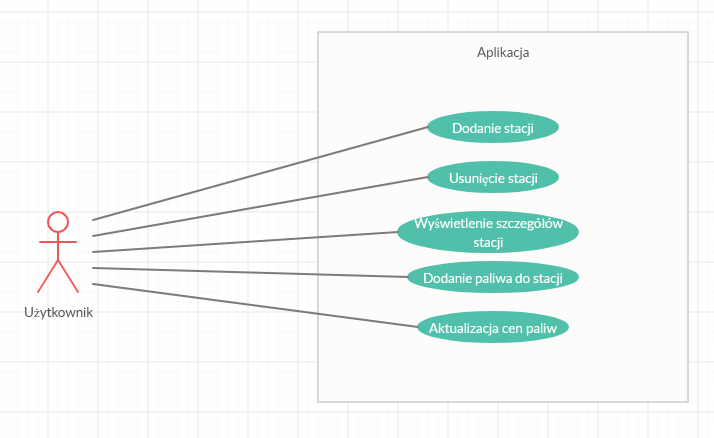
# Harmonogram realizacji projektu (diagram Gantta)



Tabela

|  |  |  |  |
| --- | --- | --- | --- |
| Nazwa zadania | Początek | Koniec | Ilość dni |
| Opis założeń projektu | 2019-05-06 | 2019-05-07 | 2 |
| Specyfikacja wymagań | 2019-05-08 | 2019-05-10 | 3 |
| Diagram użycia | 2019-05-11 | 2019-05-15 | 3 |
| Technologie oraz projekt aplikacji | 2019-05-16 | 2019-05-22 | 5 |
| Implementacja aplikacji (logika) | 2019-05-23 | 2019-06-04 | 9 |
| Utworzenie dokumentacji | 2019-06-05 | 2019-06-07 | 3 |
| Implementacja aplikacji (interfejs użytkownika) | 2019-07-14 | 2019-07-23 | 7 |
| Utworzenie dokumentacji interfejsu | 2019-07-24 | 2019-07-29 | 4 |

# Diagram użycia



# Wnioski

Stworzone API pozwala na wykonanie wszystkich wymaganych w założeniach zadań.

Do obsługi aplikacji utworzyłem interfejs w architekturze MVC.

W projekcie wykorzystałem narzędzie EntityFramework.Core, które umożliwia pracę z bazami danych. Utworzyłem dwa serwisy (FuelTypeService – do zarządzania rodzajami paliw, oraz GasStationService – do zarządzania stacjami paliw) pozwalające na wygodne zarządzanie danymi – funkcje pozwalające na wykonywanie różnych zadań znajdują się w danych dwóch serwisach.

Repozytorium projektu na GitHub: <https://github.com/wilczek013/Gasoline.git>

# Materiały źródłowe

Tworząc aplikację korzystałem z dokumentacji Microsoft, oraz starałem się nie korzystać z innych materiałów.

<https://docs.microsoft.com/pl-pl/aspnet/core/?view=aspnetcore-2.1>

<http://www.doxygen.nl/>

<https://www.getpostman.com/>

<https://www.youtube.com/watch?v=t5n07Ybz7yI> – poradnik odnośnie używania programu Postman