**WOJSKOWA AKADEMIA TECHNICZNA**

**im. Jarosława Dąbrowskiego**

****

**WYDZIAŁ CYBERNETYKI**

****

Metody Programowania .NET

Temat: **Wyświetlanie na mapie zdjęć w kontekście rejestrowanej trasy spaceru lub zwiedzania**

Autorzy:

**Kopeć Weronika**

**Michał Harasimiuk**

Grupa:

**WCY18IJ1S1**

Spis treści

[1. Wymagania funkcjonalne 3](#_Toc62336123)

[2. Model fizyczny bazy danych 4](#_Toc62336124)

[3. Specyfikacja techniczna 5](#_Toc62336125)

[3.1. Struktura 5](#_Toc62336126)

[3.2. Mapa 6](#_Toc62336127)

[3.3. Komunikacja JS oraz C# 6](#_Toc62336128)

[4. Przypadki użycia 7](#_Toc62336129)

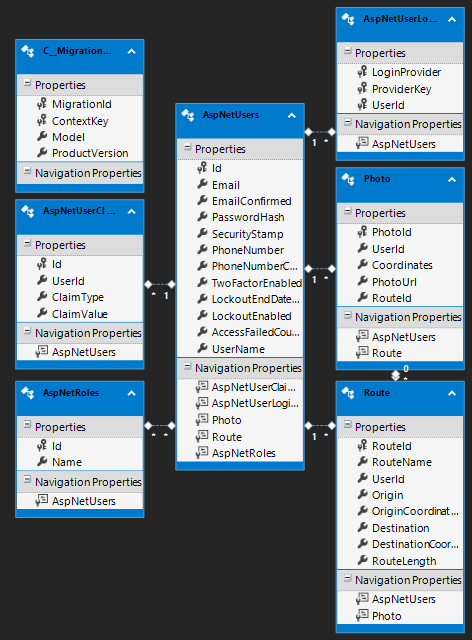
[5. Ekrany użytkownika 8](#_Toc62336130)

# Wymagania funkcjonalne

* 1. System musi uwzględniać role(uprawnienia) użytkowników („Admin”, „User”) .

1. System musi umożliwiać rejestracja nowych użytkowników, logowanie do aplikacji (nowy użytkownik otrzymuje uprawnienia „User”).
2. System musi posiadać jednego użytkownika z uprawnieniami „Admin”.
3. System musi blokować dostęp do części funkcjonalności dla niezalogowanego użytkownika i tego z uprawnieniami „User”.
4. System musi posiadać stronę główna przedstawiająca aplikację TrackerWAT.
5. System powinien posiadać symulację przesyłania opinii użytkownika .
6. System musi posiadać mapę, centrująca początkowo na teren Warszawy, z polem do wyszukiwania miejsc i wstawianie w ich miejscu markerów, opcje usuwania markerów z mapy, wstawianie markera z własnym zdjęciem z adresu URL oraz ich usuwanie, wyświetlanie długości trasy, zapis gotowej trasy ze zdjęciami do bazy.
7. System musi umożliwiać przeglądanie zapisanych tras, z możliwością ich wyświetlenia, edycji i usunięcia.
8. System musi umożliwiać przeglądanie zapisanych zdjęć z możliwością ich usunięcia.
9. System musi umożliwiać dla użytkownika z uprawnieniami „Admin” dostęp do trzech funkcjonalności więcej niż dla użytkownika z uprawnieniami „User”: wyświetlanie istniejących ról (uprawnień) , tworzenie nowych ról (uprawnień) oraz nadawanie ról (uprawnień) użytkownikom.

# Model fizyczny bazy danych



Podstawowy wygenerowany wraz z założeniem projektu model został wzbogacony o dwie tabele:

1. Photo – tabela przechowująca adresy URL zdjęć, z relacją do tabeli AspNetUsers, przechowującą identyfikator użytkownika (zdjęcie przypisane jest pod konkretnego usera) oraz relacją do tabeli Route, w celu kojarzenia zdjęć z daną trasą
2. Route – tabela przechowująca dane trasy (współrzędne miejsca stary i końca, nazwy obiektów) z relacją do tabeli AspNetUsers w celu identyfikacji tras danego użytkownika.

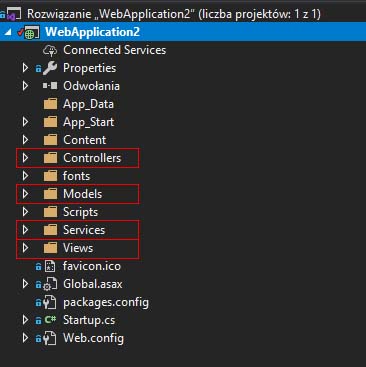
Skrypt dodający nowe tabele znajduje się w pliku **createDB.txt** w folderze głównym.

# Specyfikacja techniczna

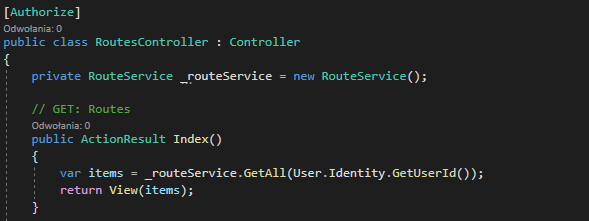
## Struktura

Aplikacja została stworzona w technologii ASP.NET Framework, ze względu na dostępny z jego poziomu Entity Framework. Informacje są zapisywane w lokalnej bazie danych SQL Server.

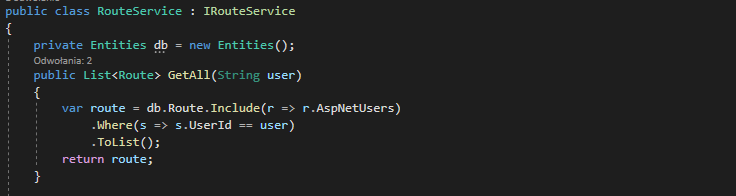
Zastosowana architektura to **MVC** (Model – View – Controller).



W obecnym modelu kontrolery są odpowiedzialne jedynie za routing. Za operacje na bazie danych odpowiadają *serwisy*.



Powyżej zrzut ekranu z kontrolera obsługującego klasę Route i odwołującego się do serwisu RouteService, implementującego funkcję wypisującą wszystkie drogi przypisane do zalogowanego użytkownika. Poniżej wspomniany serwis.



## Mapa

Główna funkcjonalność aplikacja jest osadzona w *Views/Home/Index.cshtml*. Umieszczenie mapy wymagało uruchomienia API udostępnionego od Google, a następnie wstawienia na stronę odwołania. W linku znajduje się unikalny klucz.

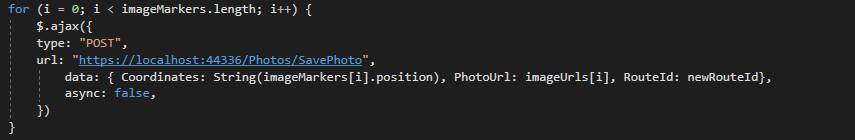


Tutaj może coś napisać? Nie musi być jakoś szczegółowo..

## Komunikacja JS oraz C#

Wysyłanie oraz odbieranie danych pomiędzy mapą (JavaScript), a kontrolerami (c#) odbywa się za pomocą technologii ajax. Dzięki temu możliwe jest ściąganie i wysyłanie danych bez potrzeby przeładowania całej strony.

W zapisie do bazy danych za pomocą tej technologii odwołujemy się do konkretnego adresu obsługiwanego przez kontrolery, przekazując odpowiednie parametry. Takie odwołania są wykonywane asynchronicznie, co oznacza, że jest wysyłane żądanie, ale nie oczekuje się jego powrotu, a wykonuje dalej kod JavaScript. W przypadku zaimplementowanego modelu bazy, gdzie przy utworzeniu nowego zdjęcia wymagane jest wprowadzenie ID trasy, wymagane było zastosowanie opcji synchronicznej.



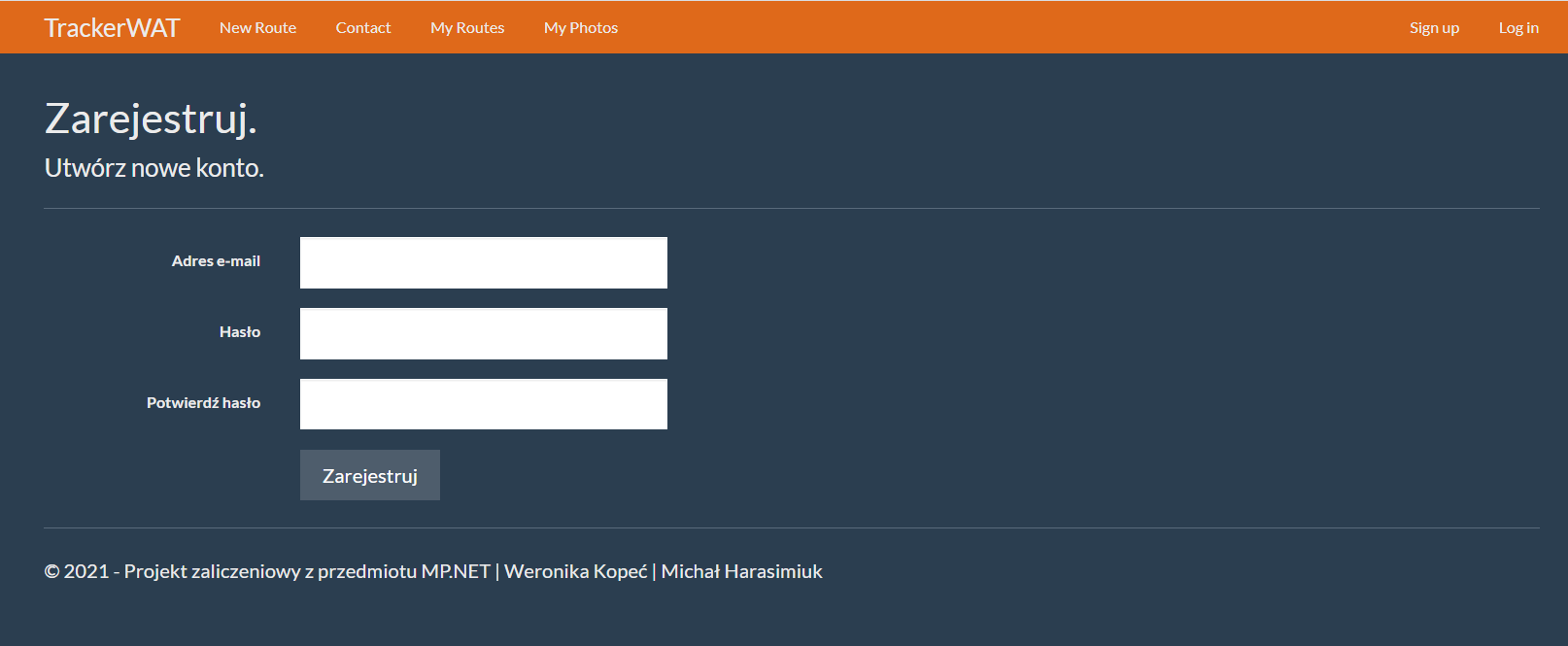
Algorytm jest następujący:

* 1. Pobierz maksymalne RouteId z tabeli Route.
  2. Dodaj nową trasę do tabeli Route, na sztywno przekazując inkrementowaną wartość z pkt. 1 do klucza głównego RouteId.
  3. Dodaj nowe zdjęcia do tabeli Photo, na sztywno podając inkrementowaną wartość z pkt. 1 do klucza obcego RouteId. Async : false dba również o to, aby proces poczekał, aż zostanie nadany nowy maksymalny PhotoId, aby nie uzyskać błędu związanego z constraintami na bazie danych.

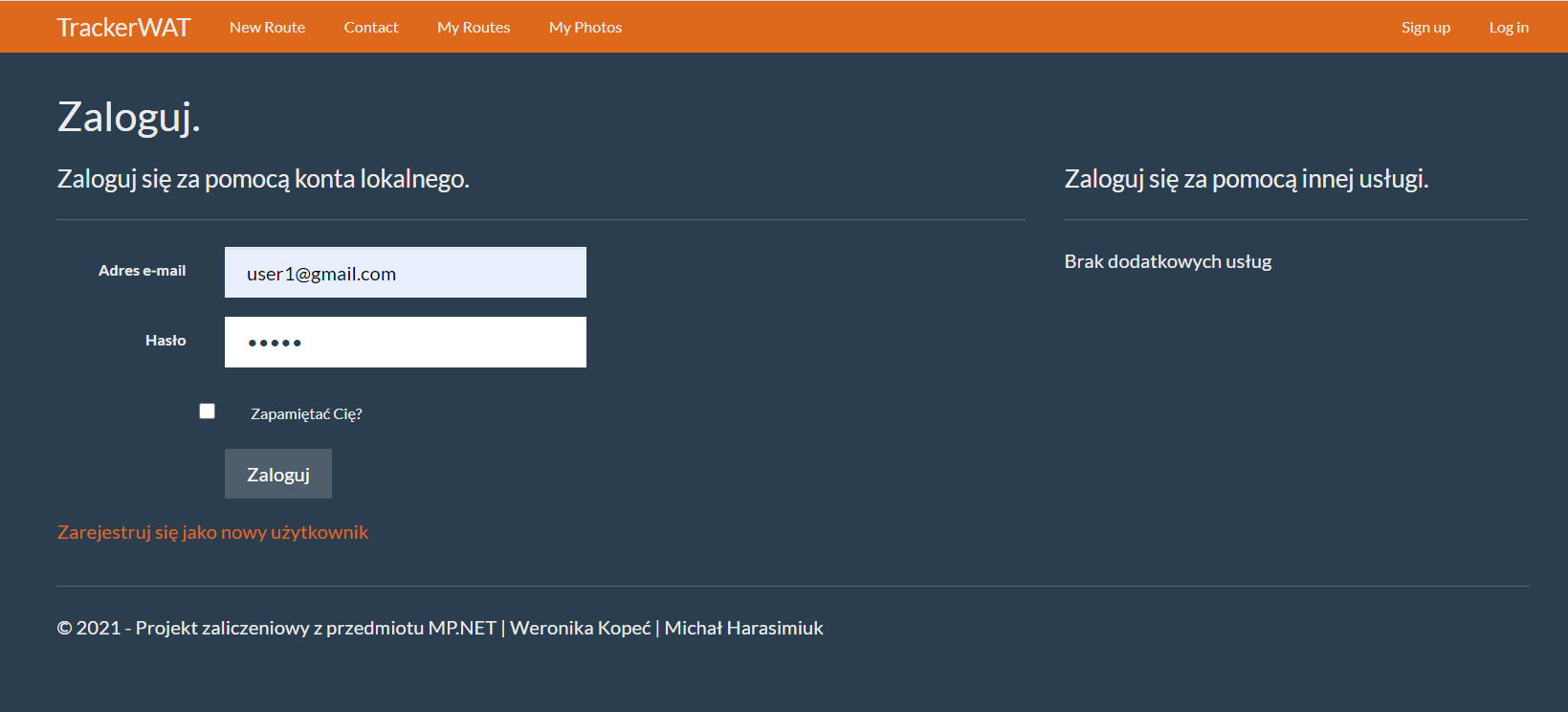
# Przypadki użycia

# Ekrany użytkownika

* 1. Rejestracja



* 1. Logowanie

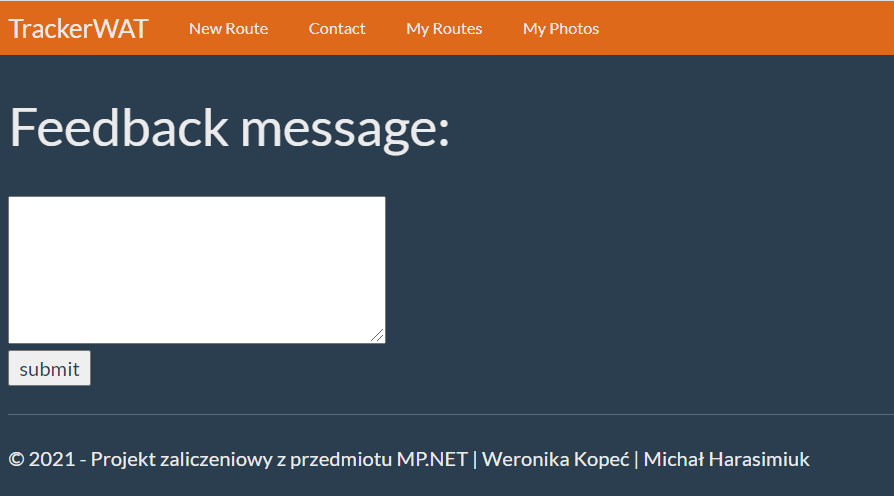


Kliknięcie zakładek *My Routes*, *My Photos* wymaga bycia zalogowanym użytkownikiem. W przypadku braku logowania i kliknięcia w te zakładki, użytkownik zostanie od razu przekierowany na stronę logowania.

* 1. Strona informacyjna

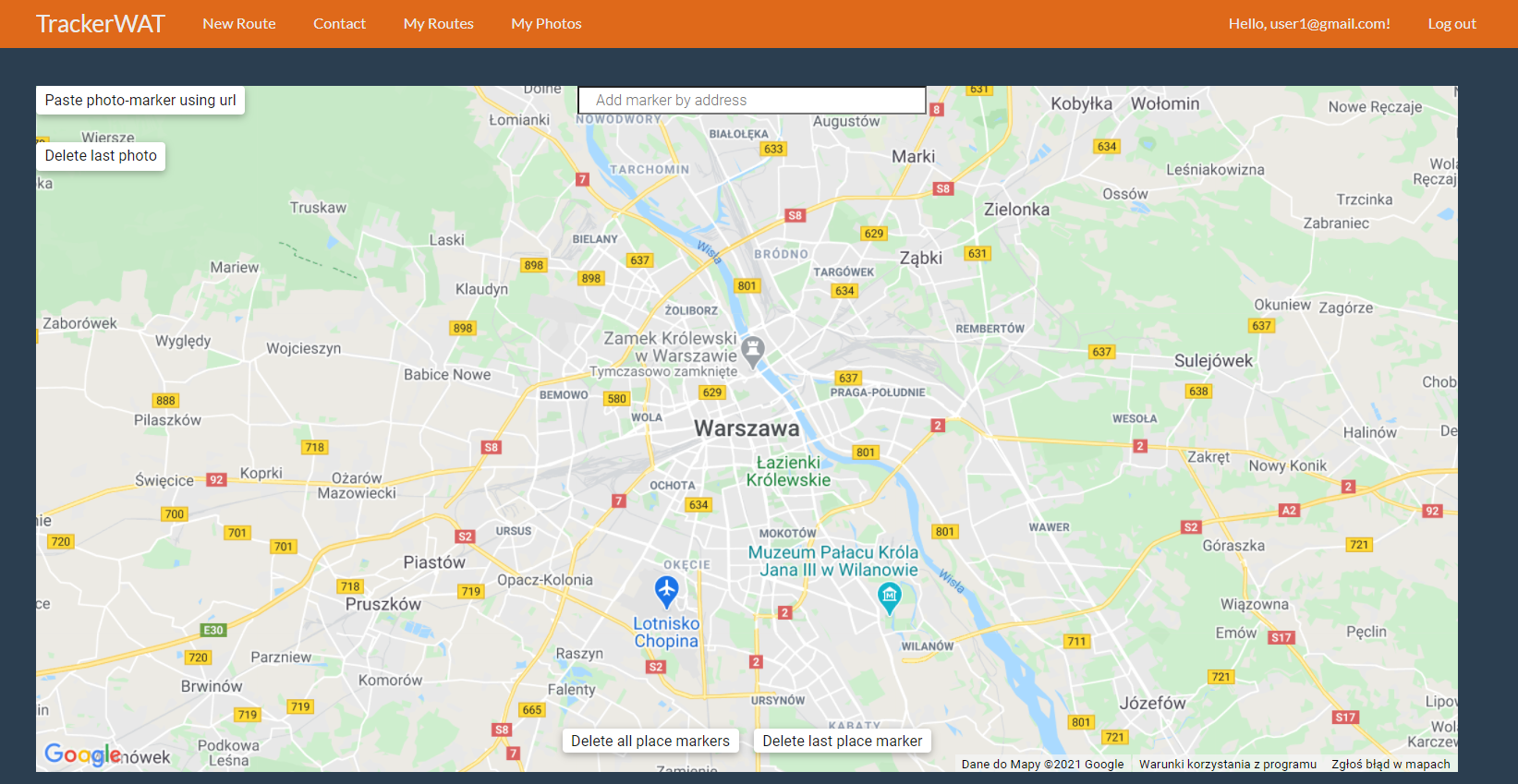


* 1. Symulacja wysyłania opinii

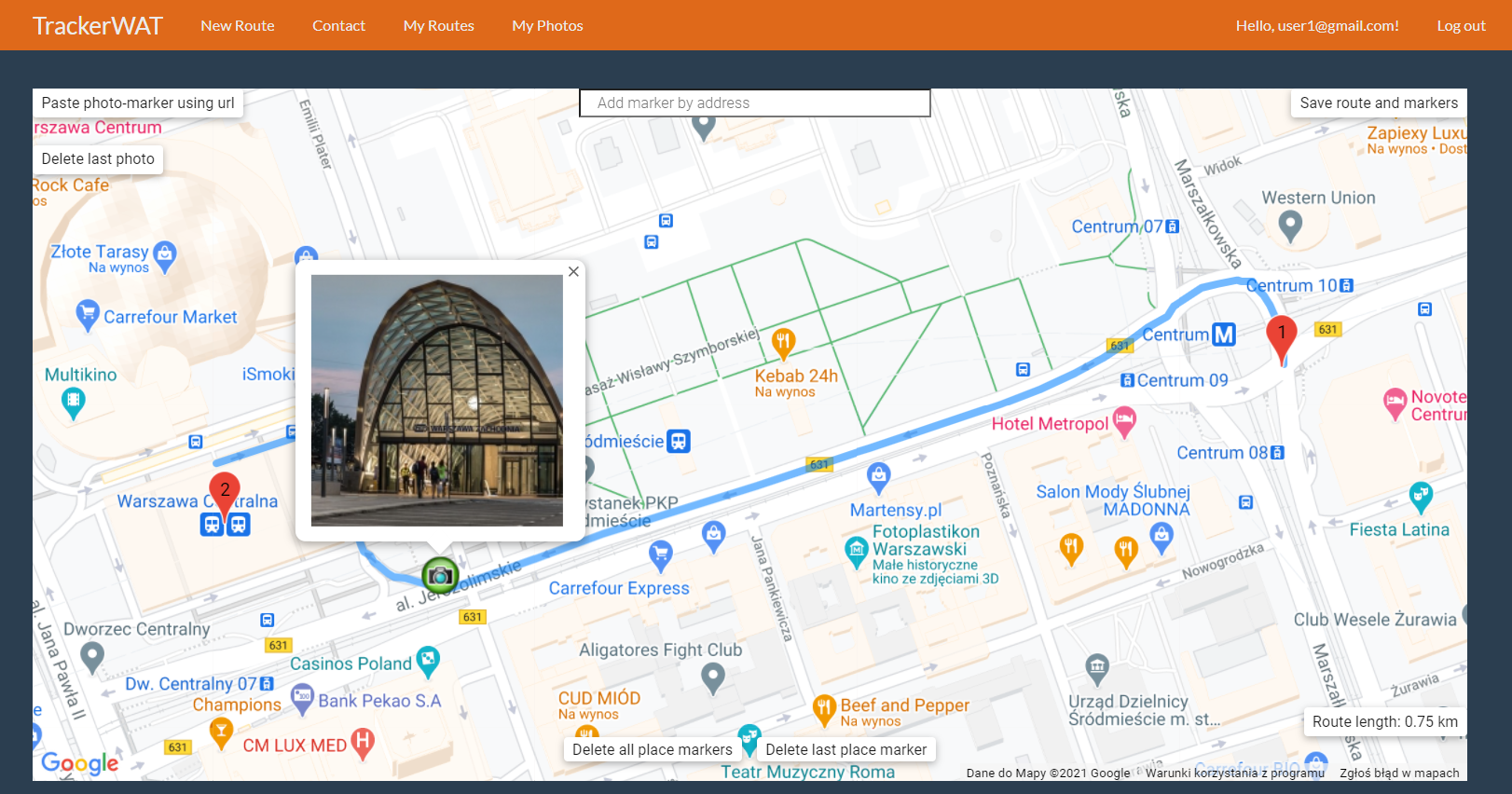




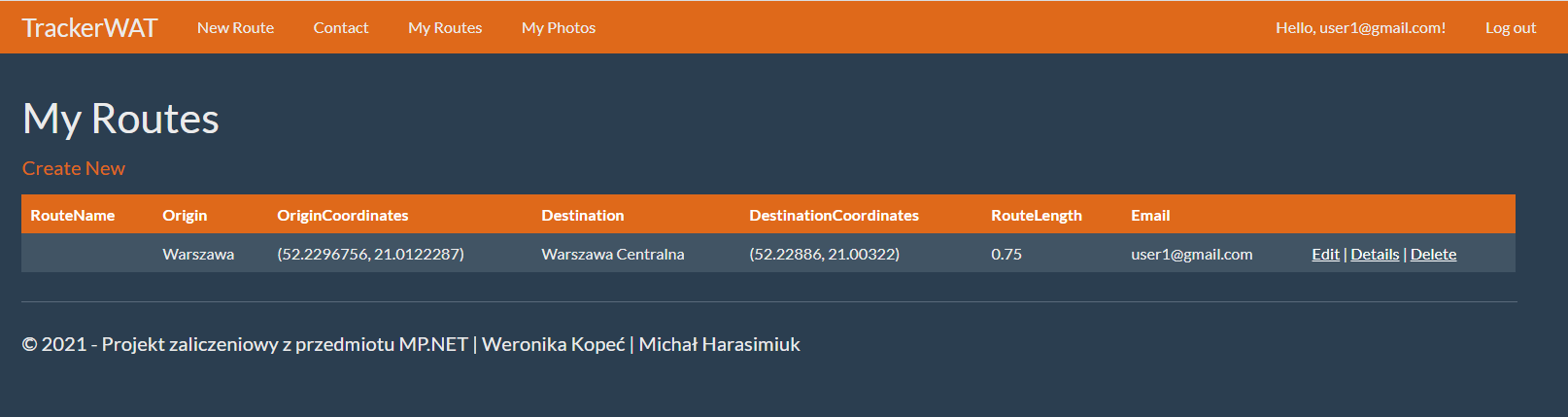
* 1. Tworzenie nowej trasy



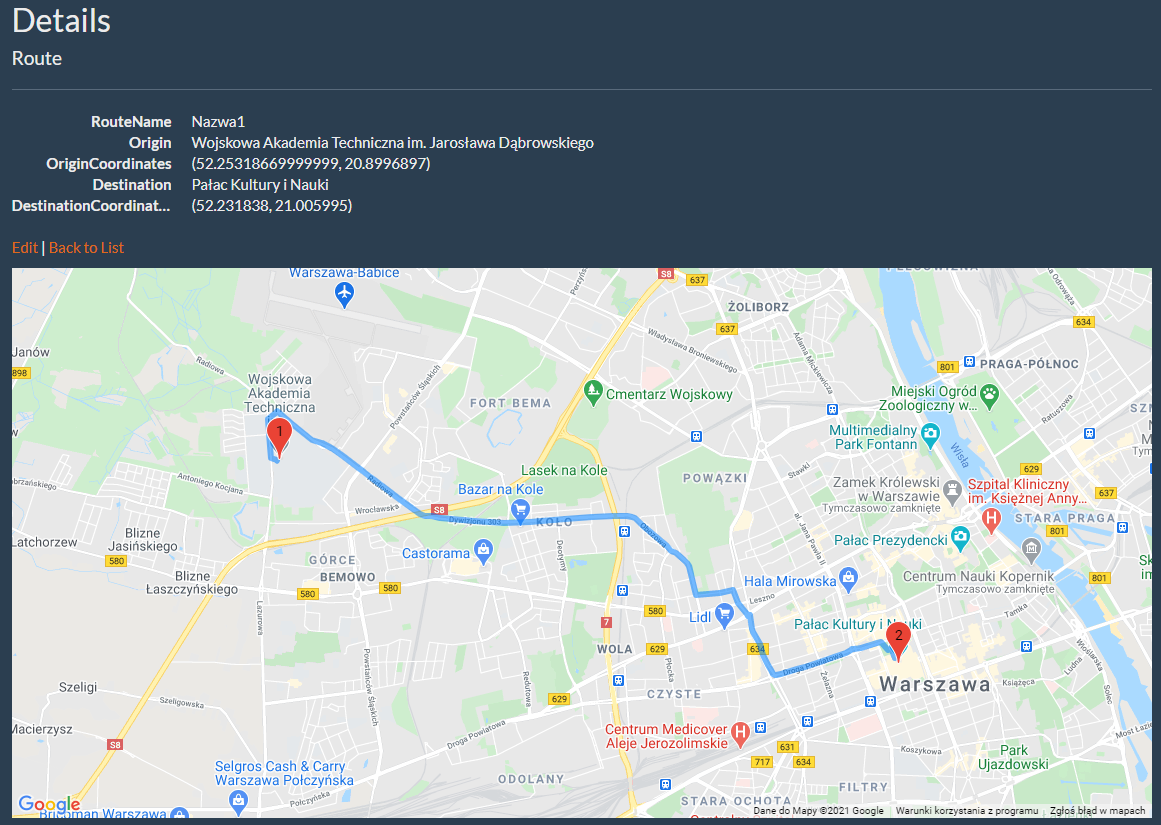
Tworzenie nowej trasy odbywa się poprzez wyszukanie obiektu w polu ***Add marker by address*** na górze mapy. Po umieszczeniu dwóch markerów (początku i końca trasy), w prawym górnym rogu pojawia się przycisk umożliwiający nam zapisanie trasy. Istnieje możliwość dodania markera ze zdjęciem. W tym celu należy kliknąć przycisk znajdujący się w górnym lewym rogu mapy, następnie wkleić link do zdjęcia. Marker pojawi się domyślnie na środku mapy, można jednak najechać na niego kursorem i przeciągnąć w dowolne miejsce na mapie. Aplikacja oferuje możliwość dodania do trasy nieskończenie wiele markerów ze zdjęciem.



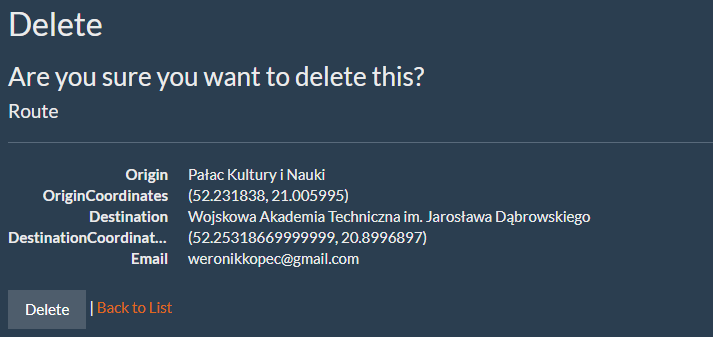
* 1. Lista tras użytkownika



* 1. Strona informacji wybranej trasy (Details)

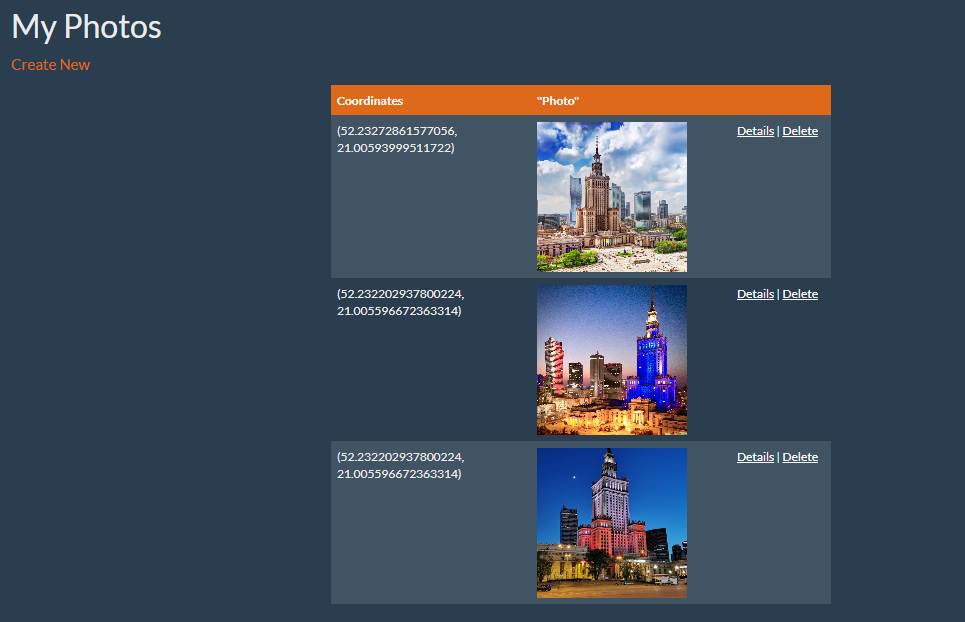


* 1. Strona usuwania trasy



Usunięcie trasy nie spowoduje usunięcia wszystkich zdjęć dodanych na tej trasie. Zamiast tego te zdjęcia otrzymają w kluczu obcym Photo.RouteId wartość NULL.

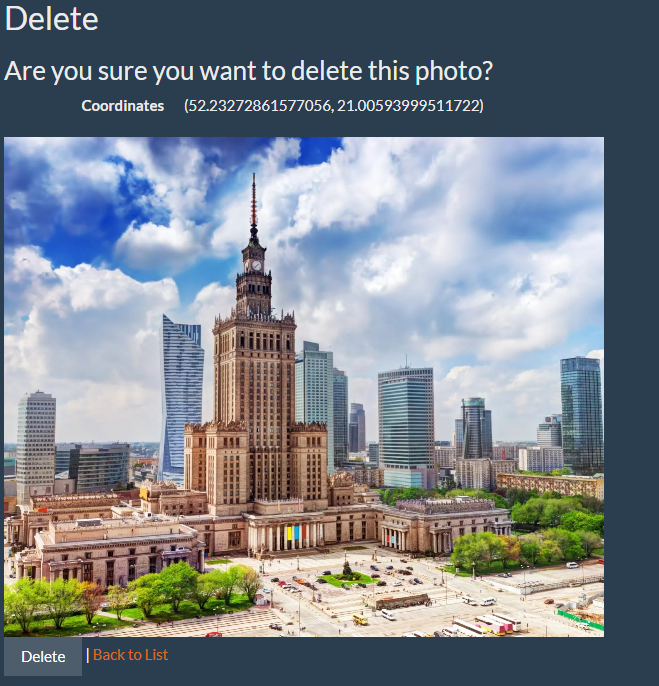
* 1. Lista zdjęć użytkownika



* 1. Strona informacji wybranego zdjęcia (Details)



* 1. Strona usuwania zdjęcia



* 1. Funkcje administratora

