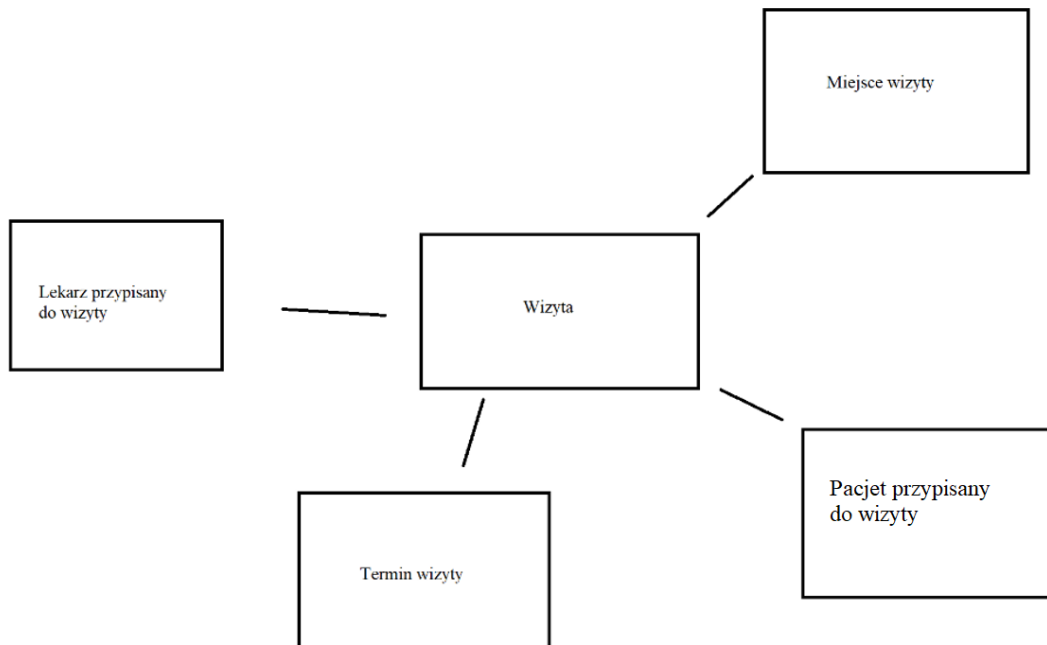


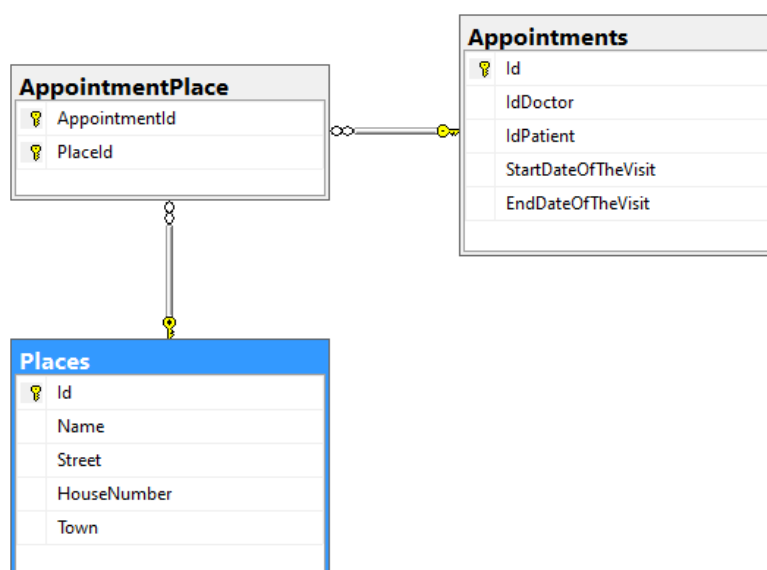
Michał Prądziński oraz Piotr Jabłoński

## Dokumentacja usługi danych (usługa danych wizyt):

### Diagram encji ideowy:




### Diagram encji z bazy danych:




## Formalny model danych w bazie danych:



Wizyta:

	Column Name	Data Type	Allow Nulls
	Id	int	<input type="checkbox"/>
	IdDoctor	int	<input type="checkbox"/>
	IdPatient	int	<input type="checkbox"/>
	StartDateOfTheVisit	datetime2(7)	<input type="checkbox"/>
	EndDateOfTheVisit	datetime2(7)	<input type="checkbox"/>
			<input type="checkbox"/>

Miejsca:

	Column Name	Data Type	Allow Nulls
	Id	int	<input type="checkbox"/>
	Name	nvarchar(MAX)	<input type="checkbox"/>
	Street	nvarchar(MAX)	<input type="checkbox"/>
	HouseNumber	nvarchar(MAX)	<input type="checkbox"/>
	Town	nvarchar(MAX)	<input type="checkbox"/>
			<input type="checkbox"/>

Tablica łącząca:

	Column Name	Data Type	Allow Nulls
	AppointmentId	int	<input type="checkbox"/>
	PlaceId	int	<input type="checkbox"/>
			<input type="checkbox"/>

Formalny model danych, który przesyłamy w formie json:

```
AppointmentDto {
  id integer($int32)
  idDoctor integer($int32)
  idPatient integer($int32)
  startDateOfTheVisit string($date-time)
  endDateOfTheVisit string($date-time)
  place PlaceDto {
    name string nullable: true
    street string nullable: true
    houseNumber string nullable: true
    town string nullable: true
  }
}
```

Przykładowe dane:

Miejsca:

	Id	Name	Street	HouseNumber	Town
1	1	Pomorski Urząd Zdrowia	Wolkowyska	23/25	Gdansk
2	2	Wojewodzki Ośrodek Ochrony Zdrowia	Mysliwska	18	Warszawa
3	3	Samodzielny Publiczny Zakład Opieki Zdrowotnej	Nowogrodzka	80	Debki
4	4	Przychodnia Lekarska	Traugutta	15A	Katowice
5	5	Gospody-Med	Gospody	7	Krakow

Tablica łącząca:

	AppointmentId	PlaceId
1	1	1
2	2	2
3	3	3
4	4	4
5	5	5
6	6	1
7	7	2
8	8	3
9	9	4
10	10	5
11	11	1
12	12	2
13	13	3
14	14	4
15	15	5

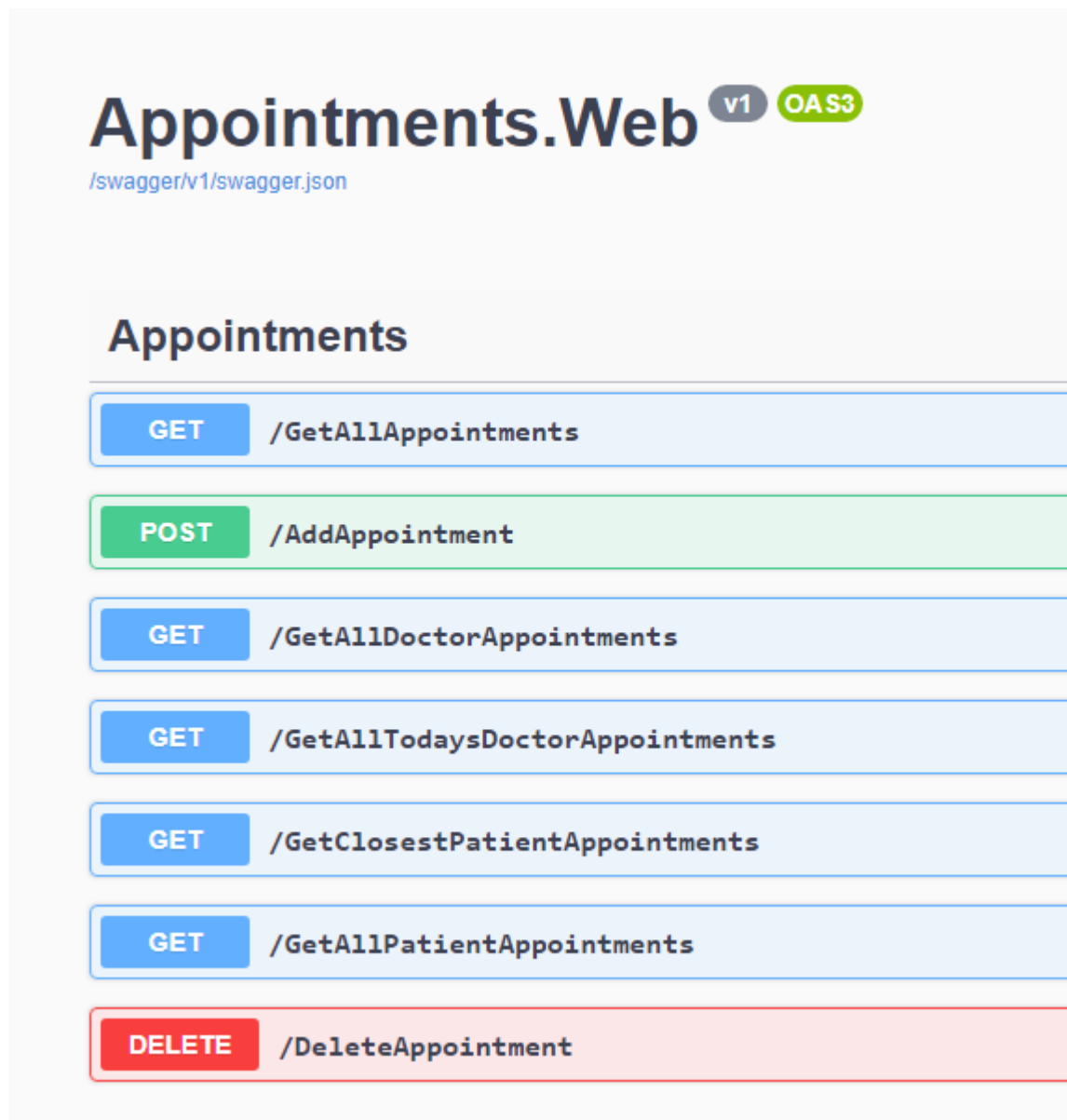
## Wizyty:

	Id	IdDoctor	IdPatient	StartDateOfTheVisit	EndDateOfTheVisit
1	1	1	1	2021-04-17 14:47:18.3600000	2021-04-17 14:47:18.3600000
2	2	1	2	2021-04-17 14:47:18.3600000	2021-04-17 14:47:18.3600000
3	3	1	3	2021-04-17 14:47:18.3600000	2021-04-17 14:47:18.3600000
4	4	2	3	2021-04-17 14:47:18.3600000	2021-04-17 14:47:18.3600000
5	5	2	2	2021-04-17 14:47:18.3600000	2021-04-17 14:47:18.3600000
6	6	3	1	2021-04-17 14:47:18.3600000	2021-04-17 14:47:18.3600000
7	7	1	8	2021-04-17 14:47:18.3600000	2021-04-17 14:47:18.3600000
8	8	3	5	2021-04-17 14:47:18.3600000	2021-04-17 14:47:18.3600000
9	9	4	3	2021-04-17 14:47:18.3600000	2021-04-17 14:47:18.3600000
10	10	4	7	2021-04-17 14:47:18.3600000	2021-04-17 14:47:18.3600000
11	11	3	6	2021-04-17 14:47:18.3633333	2021-04-17 14:47:18.3633333
12	12	6	11	2021-04-17 14:47:18.3633333	2021-04-17 14:47:18.3633333
13	13	5	9	2021-04-17 14:47:18.3633333	2021-04-17 14:47:18.3633333
14	14	5	7	2021-04-17 14:47:18.3633333	2021-04-17 14:47:18.3633333
15	15	5	11	2021-04-17 14:47:18.3633333	2021-04-17 14:47:18.3633333

Pliki initialMigration.sql oraz insert zawierają kod sql, który zawiera założenie tabel w bazie (initialMigration) oraz wstrzyknięcie danych (insert).

Pliki znajdują się w folderze razem ze sprawozdaniem.

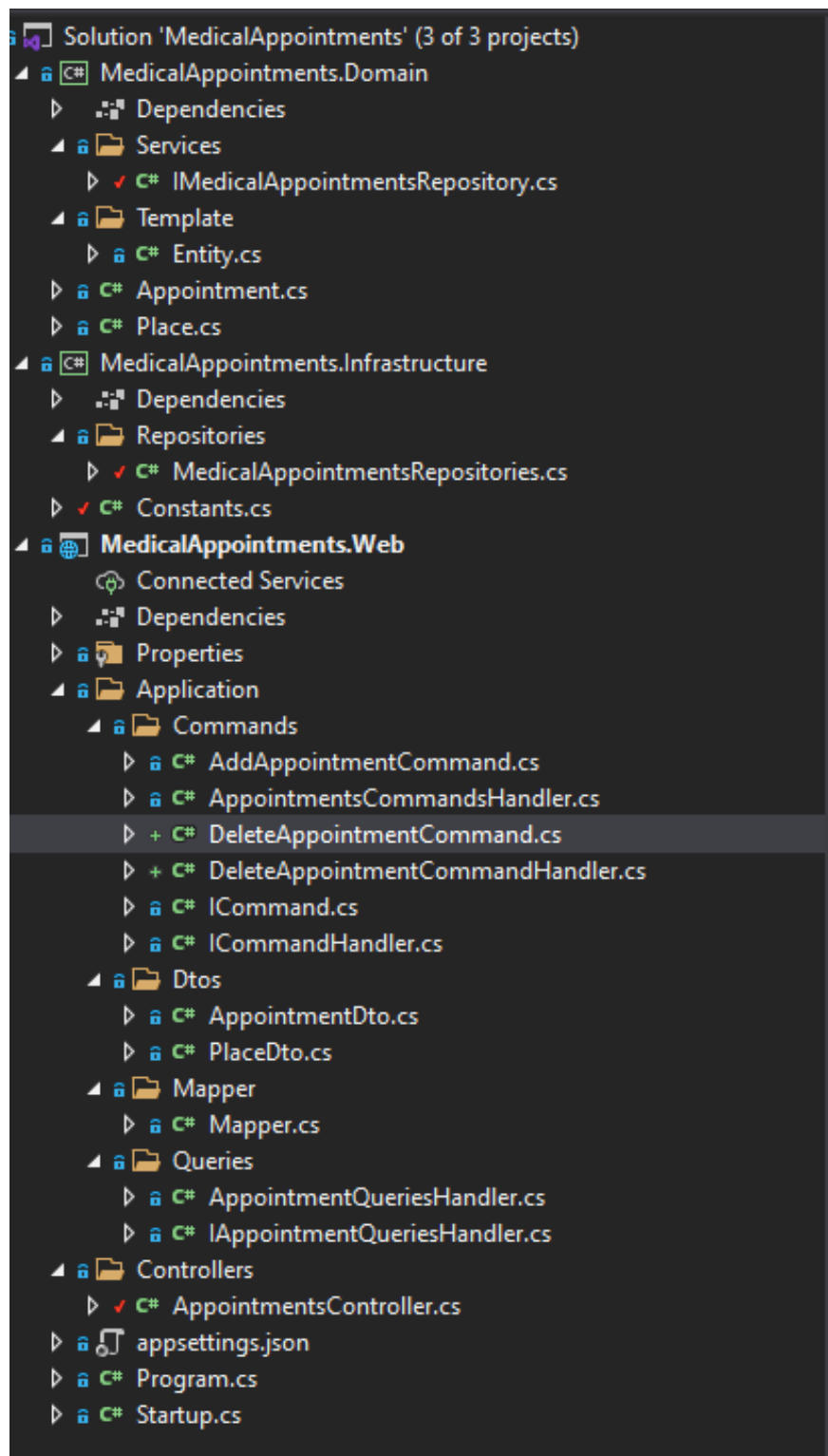
## Dokumentacja interfejsu:



- GetAllAppointments – nie przyjmuje nic; zwraca listę wizyt w formacie json
- AddAppointment – przyjmuje dane wizyty w formacie json; nic nie zwraca
- GetAllDoctorAppointments – przyjmuje parametr idDoctor; zwraca wszystkie wizyty danego lekarza w formie listy obiektów
- GetAllTodaysDoctorAppointments – przyjmuje parametr idDoctor; zwraca wszystkie dzisiejsze wizyty danego lekarza w formie listy obiektów
- GetClosestPatientAppointments – przyjmuje parametr idPatient; zwraca wizyty w formie listy obiektów, które odbędą się w czasie najbliższych dwóch tygodni

- GetAllPatientAppointments – przyjmuje parametr idPatient; zwraca wszystkie wizyty danego pacjenta w formie listy obiektów
- deleteAppointment – przyjmuje parametr id wizyty; nic nie zwraca

### Struktura rozwiązania:



Usługę danych podzieliliśmy na 3 projekty:

- MedicalAppointments.Domain – jest tutaj opisany model danych oraz interfejs repozytorium
- MedicalAppointments.Infrastructure – występuje tutaj repozytorium w folderze o tej samej nazwie; w tym repozytorium znajdują się metody komunikacji z bazą danych;
- Dodatkowo jest plik Constants – nazwa wskazuje na stałe używane np. do definicji z jaką bazą danych będzie się łączyć
- MedicalAppointments.Web – aplikacja odpowiedzialna za obsługę styków rest.

Podzielona jest na trzy części:

- dwa pliki program.cs oraz startup.cs – opisują funkcję main oraz parametry uruchomienia
- Controllers – obsługuje żądania http (również wywołuje funkcje z części projektu Application)
- Application – jest po to, aby obsłużyć obiekty data transfer object. Jest podzielony na cztery foldery:
  - Commands – aby móc dodawać i usuwać obiekty
  - Dtos – obiekty data transfer object
  - Mapper – mapowanie obiektów z repozytorium na obiekty data transfer object
  - Queries – zapytania typu get