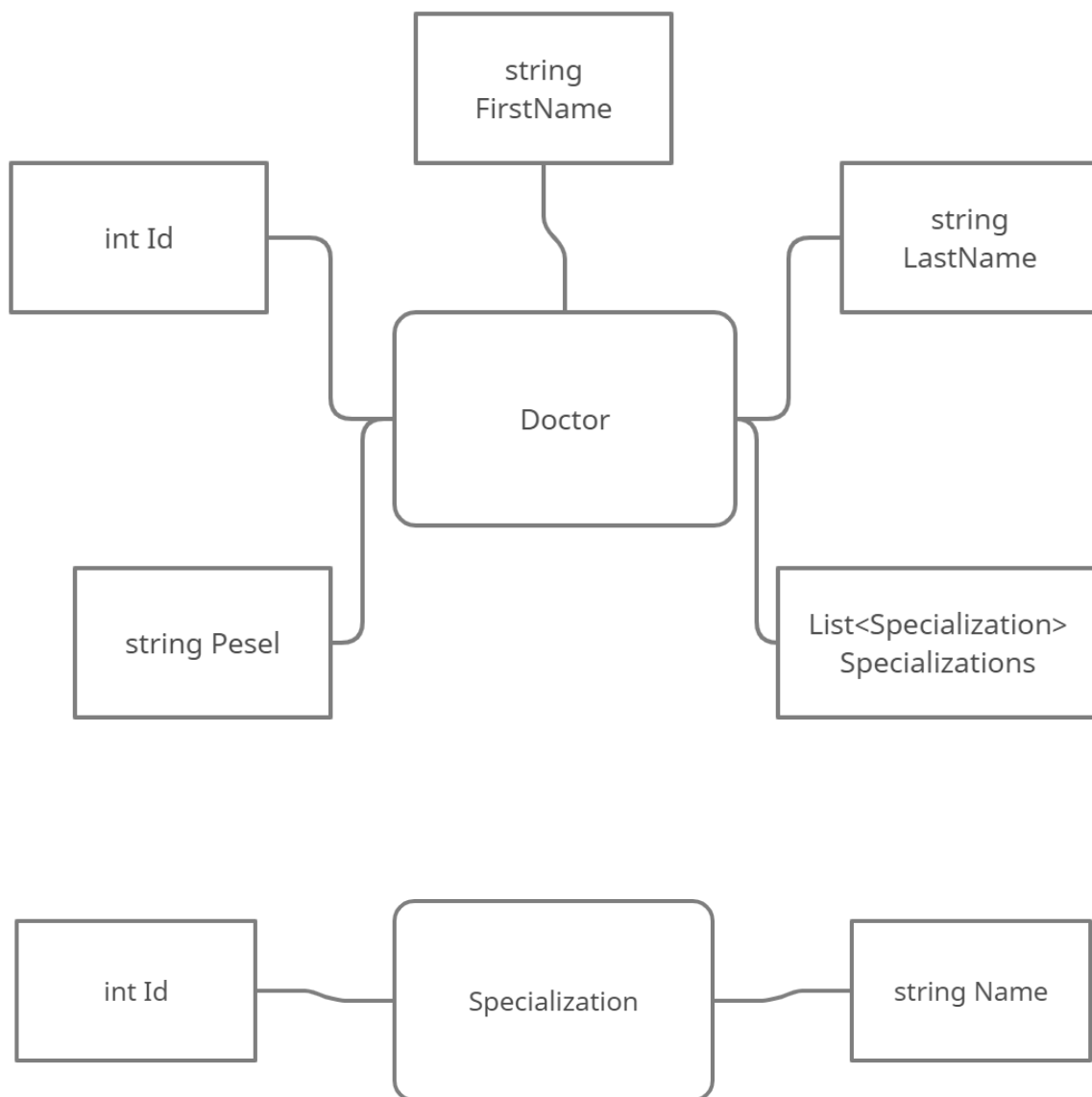


Mikroustuga danych – Lekarze

1. Diagram encji



2. Formalny model danych

Lekarz ma następujące dane:

- Id – identyfikator, typ int
- FirstName – imię, typ string
- LastName – nazwisko, typ string
- Pesel – numer pesel, typ string
- Specializations – specjalizacje, typ List<Specialization>

Specjalizacja ma następujące dane:

- Id – identyfikator, typ int
- Name – nazwa specjalizacji, typ string

3. Przykładowe dane

```
<net:Doctor id="0" firstName="Jan" lastName="Kowalski" pesel="00258869233">
  <net:Specializations>
    <net:Specialization id="0" name="Neurologia"/>
    <net:Specialization id="1" name="Okulistyka"/>
  </net:Specializations>
</net:Doctor>

<net:Doctor id="1" firstName="Paweł" lastName="Rutkowski" pesel="00348779244">
  <net:Specializations>
    <net:Specialization id="0" name="Neurologia"/>
    <net:Specialization id="2" name="Kardiologia"/>
    <net:Specialization id="3" name="Onkologia kliniczna"/>
  </net:Specializations>
</net:Doctor>

<net:Doctor id="2" firstName="Przemysław" lastName="Kucharski" pesel="01277783355">
  <net:Specializations>
    <net:Specialization id="4" name="Ortopedia"/>
  </net:Specializations>
</net:Doctor>

<net:Doctor id="3" firstName="Rafał" lastName="Kucharski" pesel="01258783332">
  <net:Specializations>
    <net:Specialization id="5" name="Pediatria"/>
  </net:Specializations>
</net:Doctor>

<net:Doctor id="4" firstName="Jaromir" lastName="Sokołowski" pesel="01252213311">
  <net:Specializations>
    <net:Specialization id="2" name="Kardiologia"/>
    <net:Specialization id="6" name="Urologia"/>
  </net:Specializations>
</net:Doctor>

<net:Doctor id="5" firstName="Przemek" lastName="Maciejewski" pesel="01348573398">
  <net:Specializations>
    <net:Specialization id="7" name="Angiologia"/>
    <net:Specialization id="3" name="Onkologia kliniczna"/>
    <net:Specialization id="8" name="Alergologia"/>
  </net:Specializations>
</net:Doctor>
```

4. Struktura rozwiązania

1. Projekt - Doctor.Logic:
 - *Doctors* – klasa odpowiadająca za logikę mikrousługi, korzysta z czytacza (Klasa *Reader*) do odczytania danych
 - *Reader* – klasa odpowiadająca za czytanie danych z pliku xml
2. Projekt - Doctors.Model
 - *IDoctors* – interfejs zawierający wzór metod dla klasy Doctors
 - *Doctor* – klasa definiująca model lekarza
 - *Specialization* – klasa definiująca model specjalizacji
3. Projekt - Doctors.Rest
 - *Controller* – klasa tworząca kontroler serwisu Rest
4. Projekt - Doctors.Rest.Model
 - *DataConverter* – klasa konwertująca dane odczytane z pliku xml do danych, które będą przesyłane w sieci
 - *DoctorData* – klasa definiująca model danych (Doktor) przesyłanych w sieci
 - *DoctorSimpleData* – klasa definiująca prostszy model danych (Doktor) przesyłanych w sieci (tzn. wyświetlane będą tylko wybrane dane: imię, nazwisko, specjalizacje)
 - *SpecializationData* – klasa definiująca model danych (Specjalizacja) przesyłanych w sieci
 - *IDoctorService* - interfejs zawierający wzór metod dla kontrolera Rest

5. Interfejs

1. Operacje usługowe:

1. Pobranie wszystkich lekarzy z bazy danych - GetAllDoctors():
 - Dane wejściowe: brak
 - Dane wyjściowe: tablica z danymi wszystkich doktorów
2. Pobranie wybranego lekarza z bazy danych dla widoku pacjenta - GetDoctor(string lastName):
 - Dane wejściowe: nazwisko szukanego lekarza
 - Dane wyjściowe: tablica z danymi (imię, nazwisko, specjalizacja) szukanых lekarzy o podanym nazwisku