

# Zaawansowane Systemy Baz Danych – Case Study „MediPlace”

Michał Ankiersztajn 311171

MediPlace to innowacyjna placówka medyczna, której głównym celem jest usprawnienie procesu umawiania wizyt i dostępu do historii medycznej pacjentów.

## ***Kluczowe funkcjonalności:***

1. **Umawianie się na wizyty względem specjalizacji lekarskich**, dzięki czemu pacjent jest w stanie szybko i skutecznie umówić się na upragnioną wizytę.
2. **Historia wizyt**
  - a. Od strony **lekarza** – potrafi szybciej wystawić diagnozę, sporządzić receptę, jak i poradzić w jaki sposób pacjent powinien się leczyć, dzięki dostępowi do historii medyczną swojego pacjenta, tego jakie leki zażywa i na co się leczy.
  - b. Od strony **pacjenta** – potrafi sprawdzić swoje wizyty, pozostawiony przez lekarza komentarz, wystawioną receptę, jak i leki, które ma zażywać.
3. **Powiadomienia o zbliżającej się wizycie** – pacjenci często umawiają się miesiące wprzód, a potem zapominają o konkretnej dacie. MediPlace przypomina swoim pacjentom o wizycie i daje im możliwość anulowania lub przełożenia danej wizyty. Dzięki temu pacjenci przychodzą na swoje wizyty lub je przekładają, co zmniejsza straty związane z pustymi wizytami.
4. **Badanie skuteczności lekarzy** – kadra menedżerska jest w stanie zidentyfikować lekarzy, którzy wykonują najwięcej wizyt i na tej podstawie stwierdzić komu należy się podwyżka, a kogo trzeba zwolnić.

## ***Możliwości rozwoju:***

1. **Pokoje, ich lokalizacja jak i wyposażenie**, które mogłoby ograniczać wizyty względem specjalizacji doktora, przykładowo stomatolog potrzebuje odpowiedniego wyposażenia do badania zębów.
2. **Pensje lekarzy**, które mogą się zmieniać w zależności od ilości i wyników zleconych im wizyt lekarskich.
3. **Płatności za wizyty**, które mogą się odbywać na raty, bo dużo osób potrzebuje pilnego leczenia, jednak w danym momencie nie ma na to środków finansowych.

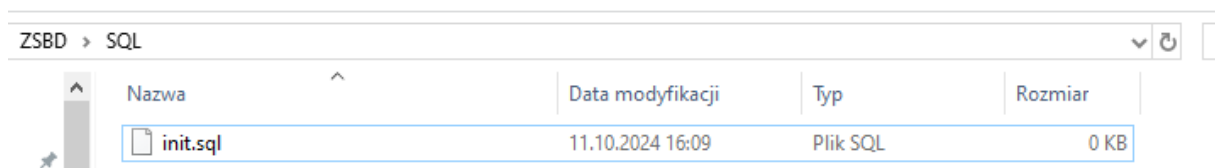
## **Instalacja i konfiguracja PostgreSQL:**

Bazę będę konfigurował i stawiał za pomocą pliku docker-compose.yml:

```
version: '3.4'
```

```
services:
  postgresql_database:
    image: postgres:latest
    environment:
      - POSTGRES_USER=admin
      - POSTGRES_PASSWORD=admin
      - POSTGRES_DB=MediPlaceDatabase
    ports:
      - "5432:5432"
    restart: unless-stopped
    volumes:
      - ./SQL:/docker-entrypoint-initdb.d
```

W projekcie tworzę również plik 'init.sql' w folderze 'SQL', na razie zostawiam pusty w przyszłości znajdą się tu instrukcje inicjalizujące tabele oraz dane w bazie danych:



ZSBD > SQL				
	Nazwa	Data modyfikacji	Typ	Rozmiar
	init.sql	11.10.2024 16:09	Plik SQL	0 KB

Następnie tworzę bazę danych w terminalu PS:

```
PS C:\Users\Michał\Desktop\ZSBD> docker compose up
```

Teraz jestem w stanie się do niej dostać z kolejnego terminala:

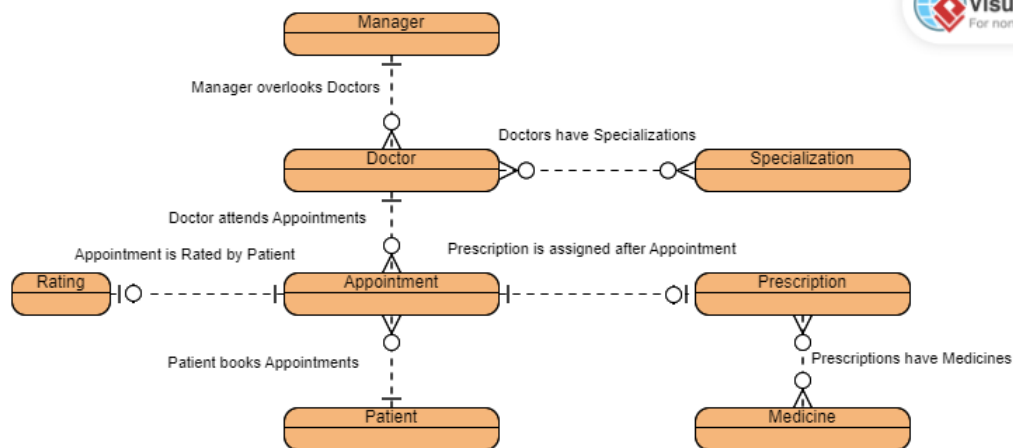
```
PS C:\Users\Michał> docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS                               NAMES
34c9118a0df3   postgres:late "docker-entrypoint.s..." About a minute Up About a minute    0.0.0.0:5432->5432/tcp    zsbdb-postgresql_database-1
PS C:\Users\Michał> docker exec -it zsbdb-postgresql_database-1 psql MediPlaceDatabase -U admin
psql (17.0 (Debian 17.0-1.pgdg120+1))
Type "help" for help.

MediPlaceDatabase=#
```

## Dlaczego PostgreSQL?

Wybrałem ją ponieważ jest to oprogramowanie typu Open Source, które jest rozwijane od ponad 35 lat. Dodatkowo mam w niej już doświadczenie (wykorzystywana w pracy inżynierskiej). Znam ją na dosyć podstawowym poziomie, a chciałbym poznać bardziej skomplikowane funkcje, jak i zarządzania dostępem. Myślę, że jest to dobra wiedza wstępna do zadanego ćwiczenia.

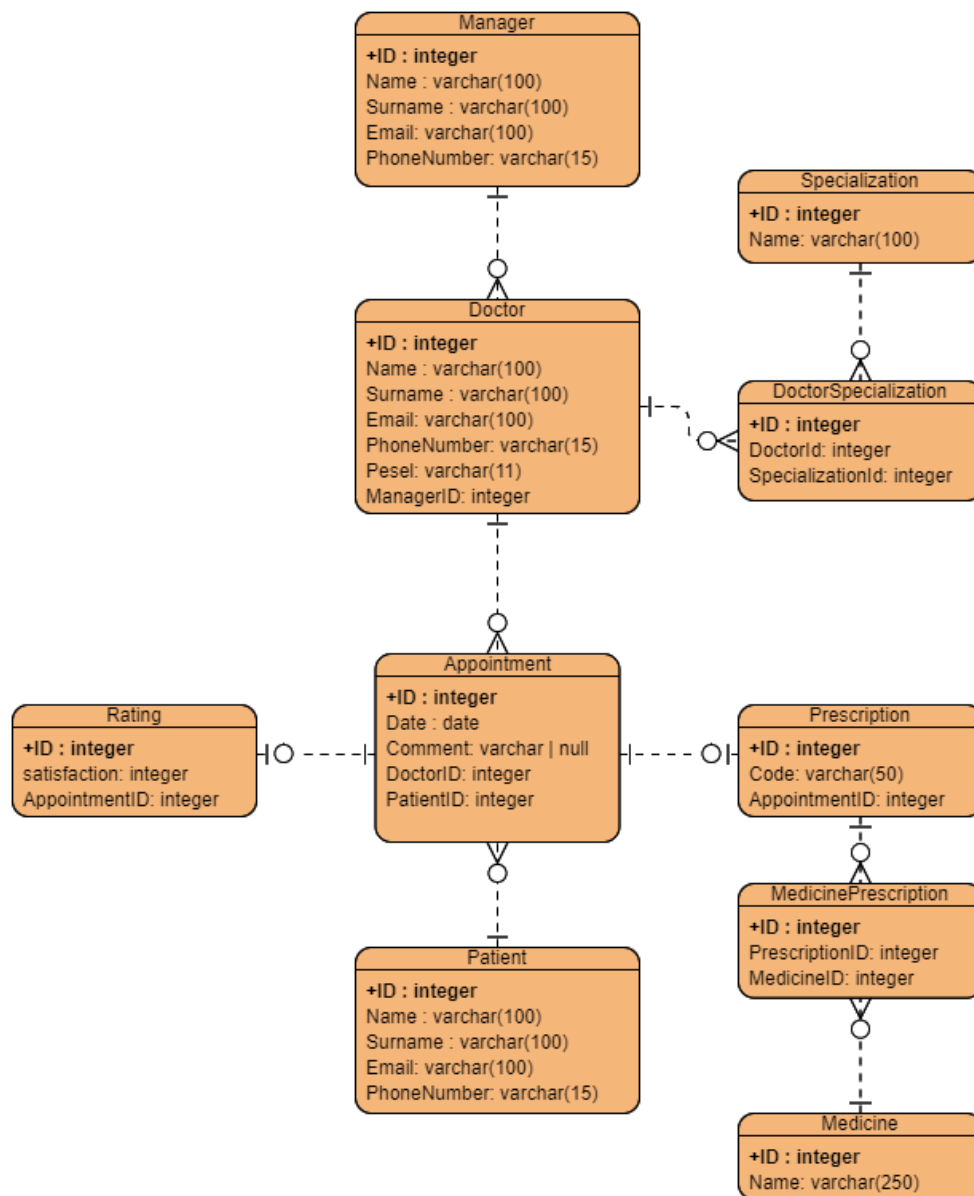
## Baza danych:



### Wstępny Diagram ERD

Użytkownicy, którzy będą korzystać z bazy danych:

- Lekarze:
  - Dostęp do wszystkich wizyt do których są przypisani (zarówno w przeszłości jak i w przyszłości)
  - Aktualizacji notatki w wizycie
  - Dodanie recepty do wizyty
  - Dostęp do danych pacjenta, który jest powiązany z wizytą należącą do danego lekarza i jego wizyt razem z receptami, które zawierają leki
- Menedżerzy
  - Dostęp do danych lekarzy i ilości ich wizyt
  - Dodawanie specjalizacji lekarzom
  - Aktualizacja danych lekarza
- Pacjenci
  - Wyświetlanie swoich przeszłych wizyt razem z receptami
  - Wyświetlanie możliwych wizyt w danym przedziale czasowym (tak, aby zarówno pacjent, jak i lekarz byli w stanie przyjść na daną wizytę)



Schemat bazy danych na podstawie ERD