

Dziabadaj

Projekt z Baz Danych 1

Informatyka Stosowana FiIS AGH

Mateusz Górczany

Luty 2021

1 Projekt koncepcji, założenia

1.1 Zdefiniowanie tematu projektu

Projekt zakłada wykonanie aplikacji, pozwalającej zalogowanemu użytkownikowi zarejestrować się na szczepienie. Wykorzystany zostanie panel rejestracyjny wykonany na projekt 2 z Technik Internetowych.

1.2 Analiza wymagań użytkownika

Użytkownik miałby możliwość rejestracji w systemie oraz wyboru daty szczepienia, podglądu swoich danych, a także podglądu daty i miejsca szczepienia.

1.3 Zaprojektowanie funkcji

1. dodawanie:

- (a) pacjenta - wraz z adresem, informacjami osobistymi i kontaktem
- (b) personelu - wraz z adresem, informacjami osobistymi i kontaktem
- (c) placówki wraz z adresem i kontaktem
- (d) przychodni
- (e) szczepienia
- (f) producenta
- (g) szczepionki

2. pobieranie:

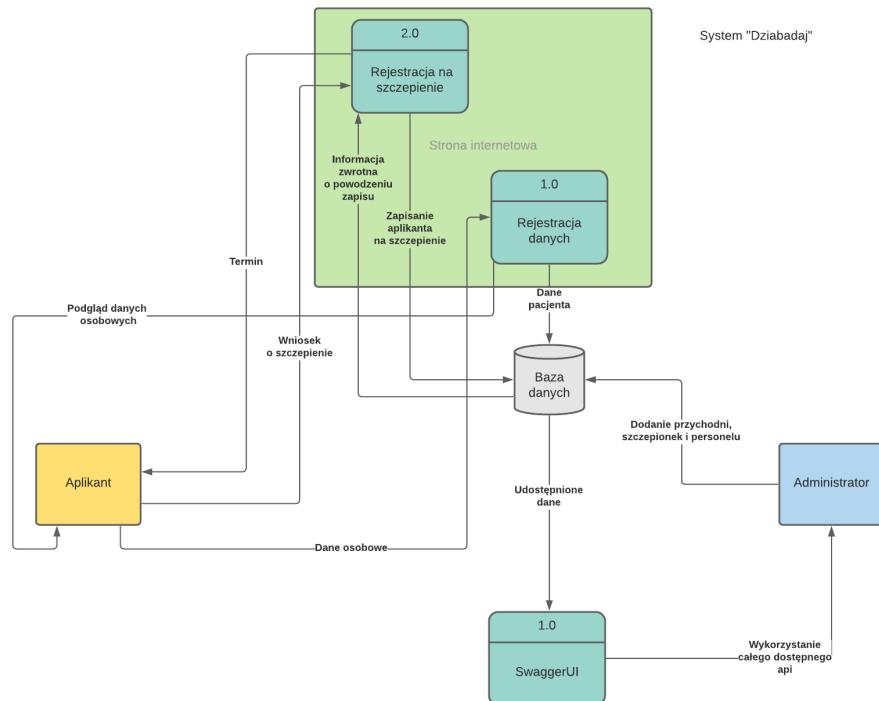
- (a) pełnych danych pacjenta
- (b) pełnych danych placówki
- (c) pełnych danych personelu
- (d) pełnych danych szczepienia

3. monitorowanie ilości chętnych na dany dzień dla placówek
4. zwracanie statystyk, co określających liczbę chętnych do szczepienia ze względu na miejsce szczepienia oraz miejsce pochodzenia

Administrator: - dostęp do pełnych funkcjonalności api za pomocą SwaggerUI - operowanie na obiektach JSON

2 Projekt diagramów

2.1 Budowa i analiza diagramu przepływu danych (DFD)



2.2 Zdefiniowanie encji (obiektów) oraz ich atrybutów

pole	typ	klucz	opcje	opis
id	int	primary	not null, automatycznie generowany	
kraj	varchar		default 'Polska', walidacja regex	
miasto	varchar		not null, walidacja regex	
kod_pocztowy	varchar		not null, walidacja regex jeśli kraj to Polska	
ulica	varchar		not null, walidacja regex	
numer	varchar		not null	numer domu

Tablica 1: adres

pole	typ	klucz	opcje	opis
id_adres	int	primary, foreign	not null	indeks adresu
id_osoba	int	primary, foreign	not null	indeks osoby, do której się referuje

Tablica 2: adres-osoba

pole	typ	klucz	opcje	opis
id_adres	int	primary, foreign	not null	indeks adresu
id_placowka	int	primary, foreign	not null	indeks placówki, do której się referuje

Tablica 3: adres-placówka

pole	typ	klucz	opcje	opis
id	int	primary	not null, automatycznie generowany	
telefon	varchar			
email	varchar		unique	
info	varchar			dodatkowe informacje

Tablica 4: kontakt

pole	typ	klucz	opcje	opis
id_kontakt	int	primary, foreign	not null	indeks kontaktu
id_osoba	int	primary, foreign	not null	indeks osoby, do której się referuje

Tablica 5: kontakt-osoba

pole	typ	klucz	opcje	opis
id_kontakt	int	primary, foreign	not null	indeks kontaktu
id_placowka	int	primary, foreign	not null	indeks placówki, do której się referuje

Tablica 6: kontakt-placówka

pole	typ	klucz	opcje	opis
id	int	primary	not null, automatycznie generowany	
imie	varchar		not null	
nazwisko	varchar		not null	
data_uro	date		not null, sprawdzenie czy >1900	
pesel	varchar		not null, unique	
haslo	varchar		not null	

Tablica 7: osoba

pole	typ	klucz	opcje	opis
id	int	primary	not null, automatycznie generowany	
id_osoba	int	foreign	not null	indeks osoby, do której referuje
pozycja	pozycja_t		not null	zajmowane stanowisko

Tablica 8: personel

pole	typ	klucz	opcje	opis
id	int	primary, foreign	not null	indeks osoby
info	varchar		not null	informacje o pacjencie

Tablica 9: pacjent

pole	typ	klucz	opcje	opis
id	int	primary	not null, automatycznie generowany	
data	date		not null, \geq CURRENT_DATE	data szczepienia
status	status_t		not null	status szczepienia
id_personel	int	foreign		
id_pacjent	int	foreign	not null	
id_szczepionka	int	foreign		
id_placowka	int	foreign		

Tablica 10: szczepienie

pole	typ	klucz	opcje	opis
id	int	primary	not null, generowany automatycznie	
nazwa	varchar		not null	nazwa placówki
id_przychodnia	int	foreign	not null	
info	varchar			informacje o placówce
dzienny_limit	int			limit chętnych do szczepienia

Tablica 11: placówka

pole	typ	klucz	opcje	opis
id_placowka	int	primary, foreign	not null	
data	date	primary	not null, default CURRENT_DATE	dzień pomiaru
chetni	int		not null	ilość chętnych w danym dniu

Tablica 12: monitor

pole	typ	klucz	opcje	opis
id	int	primary	not null, generowany automatycznie	
nazwa	varchar		not null	nazwa szczepionki
dawki	int		not null	liczba dawek w 1 fiolce
rozmiar	varchar		not null	rozmiar pudełka szczepionek, zapisany w formacie: wys-szer-dlug, należy podać jednostkę
waga	float		not null	waga pudełka w kg
id_producent	int	foreign		
liczba_per_box	int		not null	liczba szczepionek w jednym pudełku

Tablica 13: szczepionka

pole	typ	klucz	opcje	opis
id	int	primary	not null, generowany automatycznie	
nazwa	varchar		not null	nazwa producenta

Tablica 14: producent

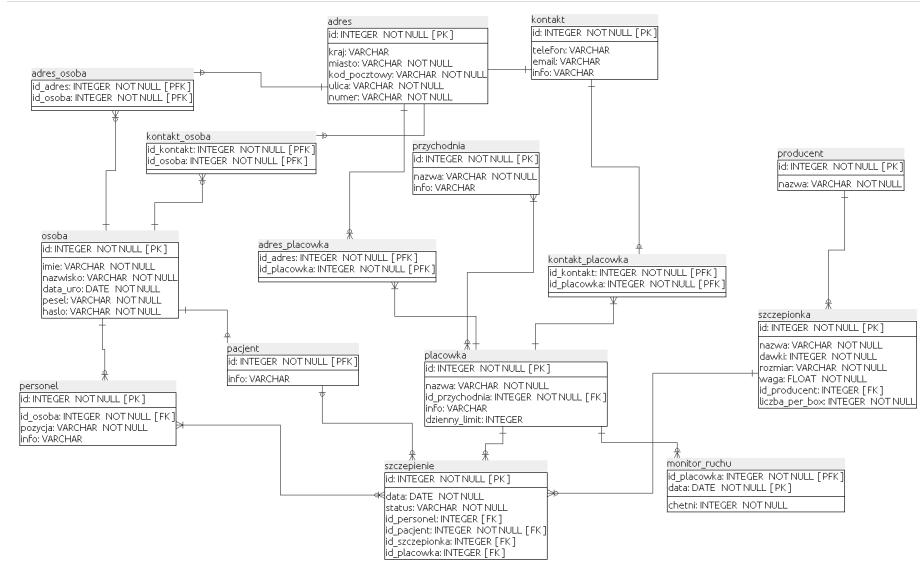
pole	typ	klucz	opcje	opis
id	int	primary	not null, generowany automatycznie	
nazwa	varchar		not null	nazwa przychodni
info	varchar			informacje o przychodni

Tablica 15: przychodnia

2.3 Zaprojektowanie relacji pomiędzy encjami

- adres 0:1 adres placowka
- kontakt 0:1 kontakt placowka
- placowka 1:n kontakt placowka
- placowka 1:n adres placowka
- adres 0:1 adres osoba
- kontakt 0:1 kontakt osoba
- osoba 1:n kontakt osoba
- osoba 1:n adres osoba
- osoba 1:n personel
- osoba 0:1 pacjent
- pacjent 1:n szczepienie
- personel 0:n szczepienie
- placowka 0:n szczepienie
- placowka 1:n monitor ruchu
- producent 0:n szczepionka
- szczepionka 0:n szczepienie

- przychodnia 1:n placowka



3 Projekt logiczny

3.1 Projektowanie tabel, kluczy, indeksów

Spełnione w 2.2.

3.2 Słowniki danych

Spełnione w 2.2.

Dodano także:

Listing 1: Typy stanowisk dla personelu

```
CREATE TYPE pozycja_t
AS ENUM (
    'lekarz',
    'doktor',
    'profesor',
    'pielegniarka'
);
```

Listing 2: Dostępne statusy szczepienia

```
CREATE TYPE status_t
AS ENUM (
```

```
'I_tura , `oczekuje' ,  
'II_tura , `oczekuje' ,  
'zakonczono'  
);
```

3.3 Analiza zależności funkcyjnych i normalizacja tabel (dekompozycja do 3NF ewentualnie BCNF)

Tabele są w postaci 3NF.

3.4 Denormalizacja struktury tabel

Stworzono tabelę monitor ruchu, który rejestruje ilość chętnych w danym dniu dla danej placówki. Dokonano tego z uwagi na fakt, że informacja o szczepieniu znika wraz z usunięciem pacjenta, co zmodifykowałoby historyczne dane oparte na tej tabeli.

Wydzielono oddzielne tabele - adres i kontakt, gdyż typ przetwarzanych danych powtórzyłby się w tabelach osoba oraz placówka.

3.5 Zaprojektowanie operacji na danych

Listing 3: 1(a)

```
dodaj_pacjenta(  
    imie varchar ,  
    nazwisko varchar ,  
    data_uro date ,  
    pesel varchar ,  
    haslo varchar ,  
    miasto varchar ,  
    kod_pocztowy varchar ,  
    ulica varchar ,  
    numer varchar ,  
    telefon varchar ,  
    email varchar ,  
    kraj varchar default 'Polska' ,  
    kontakt_info varchar default null ,  
    pacjent_info varchar default null  
)
```

Listing 4: 1(b)

```
dodaj_personel(  
    pozycja pozycja_t ,  
    imie varchar ,  
    nazwisko varchar ,
```

```
    data_uro date ,  
    pesel varchar ,  
    haslo varchar ,  
    miasto varchar ,  
    kod_pocztowy varchar ,  
    ulica varchar ,  
    numer varchar ,  
    telefon varchar ,  
    email varchar ,  
    kraj varchar default 'Polska' ,  
    kontakt_info varchar default null ,  
    personel_info varchar default null  
)
```

Listing 5: 1(c)

```
dodaj_placowke(  
    nazwa varchar ,  
    dzienny_limit integer ,  
    id_przychodnia integer ,  
    miasto varchar ,  
    kod_pocztowy varchar ,  
    ulica varchar ,  
    numer varchar ,  
    telefon varchar ,  
    email varchar default null ,  
    kontakt_info varchar default null ,  
    placowka_info varchar default null ,  
    kraj varchar default 'Polska'  
)
```

Listing 6: 2(a)

```
view informacje_pacjentow
```

Listing 7: 2(b)

```
view placowka_informacje
```

Listing 8: 2(c)

```
view personel_detailed
```

Listing 9: 2(d)

```
view vaccinations_detailed
```

Listing 10: 3

```
trigger obslugiwacz_szczepien  
trigger obslugiwacz_monitora
```

Listing 11: 4

```
view applicants_per_city_by_day  
view applicants_per_origin
```

Listing 12: Walidacja dodawanych osób

```
trigger validator_osoby
```

Listing 13: Obsługa usuwania rekordów:

```
trigger manager_usuwania_pacjentow  
trigger manager_usuwania_personelu
```

Ponadto użyto słów kluczowych **CHECK** oraz **UNIQUE** do ograniczenia wprowadzania niepoprawnych danych, np. złego kodu pocztowego dla adresu z Polski:

```
check(( kraj='Polska ' and kod_pocztowy similar to '\d{2}-\d{3} ') or kraj<>'Polska'
```

Listing 14: validator nazw - błędne są takie które posiadają niepoprawne znaki np liczby

```
waliduj_nazwe(nazwa varchar)
```

Pozostałe ograniczenia można zobaczyć w załączonym pliku SQL.

Reszta funkcjonalności została wykonana poleceniami SQL z poziomu serwera. Wersje angielskie pojawiły się na potrzeby api.

4 Projekt funkcjonalny

4.1 Interfejsy do prezentacji, edycji i obsługi danych

Wprowadzanie:

1. Formularz rejestracyjny: <http://localhost:1999/>, pobierane dane:

- imię
- nazwisko
- datę urodzenia
- pesel
- email
- telefon
- miejscowości

- ulice
- numer domu
- kod pocztowy
- hasło

2. Formularz logowania: <http://localhost:1999/login>, pobierane dane:

- email
- hasło

3. Formularz rejestracji na szczepienie: <http://localhost:1999/details>, pobierane dane:

- data szczepienia
- id placówki

Prezentacja:

1. Po zalogowaniu dane użytkownika pojawiają się automatycznie <http://localhost:1999/details>
2. Po rejestracji na szczepienie pojawia się automatycznie, <http://localhost:1999/vaccinations>

4.2 Wizualizacja danych

- przegląd danych użytkownika po zalogowaniu: <http://localhost:1999>
- przegląd dostępnych placówek: <http://localhost:1999/vaccinations>
- podgląd informacji nt. miejsca szczepienia - po rejestracji wyboru placówki oraz logowaniu: <http://localhost:1999/vaccinations>

4.3 Makropolecenia

Do pomocy obsługi formularzy posłużyły funkcje zdefiniowane w punkcie 3.5.

5 Dokumentacja

5.1 Wprowadzanie danych

logowanie : <http://localhost:1999/login> SwaggerUI = <http://localhost:1999/api/v1>

1. pacjent + osoba + kontakt + adres : formularz <http://localhost:1999> lub SwaggerUI lub ręcznie
2. szczepienie : formularz <http://localhost:1999/vaccinations> SwaggerUI lub ręcznie

3. personel + osoba + kontakt + adres : SwaggerUI
4. placówka + adres + kontakt : SwaggerUI lub ręcznie
5. przychodnia : SwaggerUI lub ręcznie
6. producent : SwaggerUI lub ręcznie
7. szczepionka : SwaggerUI lub ręcznie
8. monitor ruchu : automatycznie

5.2 Dokumentacja użytkownika

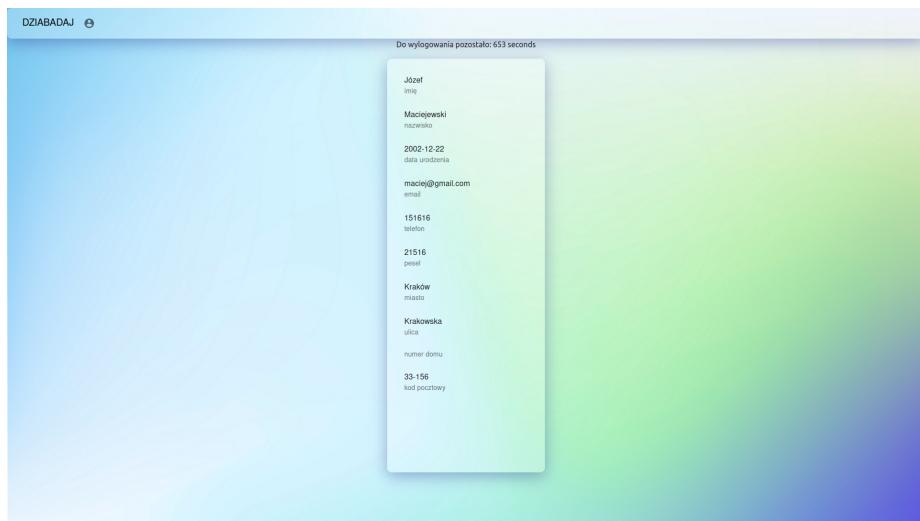
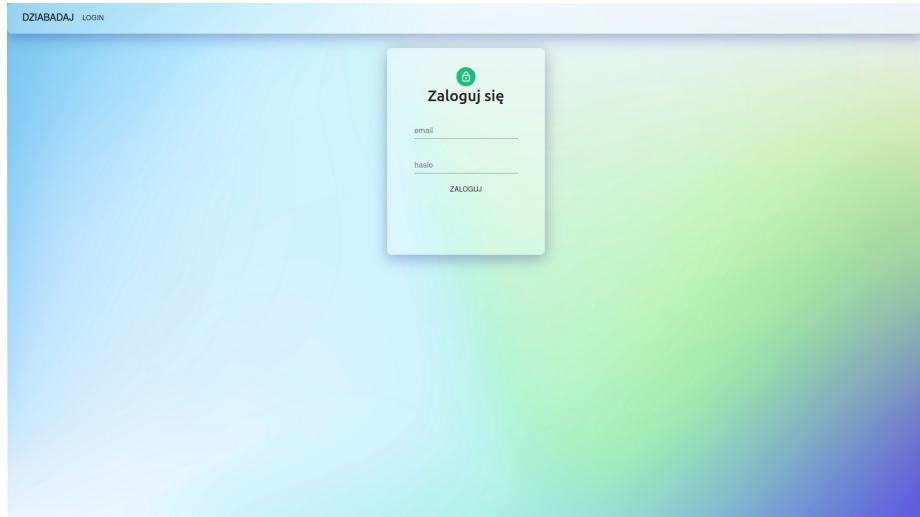
Baza danych PostgreSQL jest usługą świadczoną przez stronę elephantdb.com, wobec czego do działania aplikacji niezbędne jest połączenie internetowe.

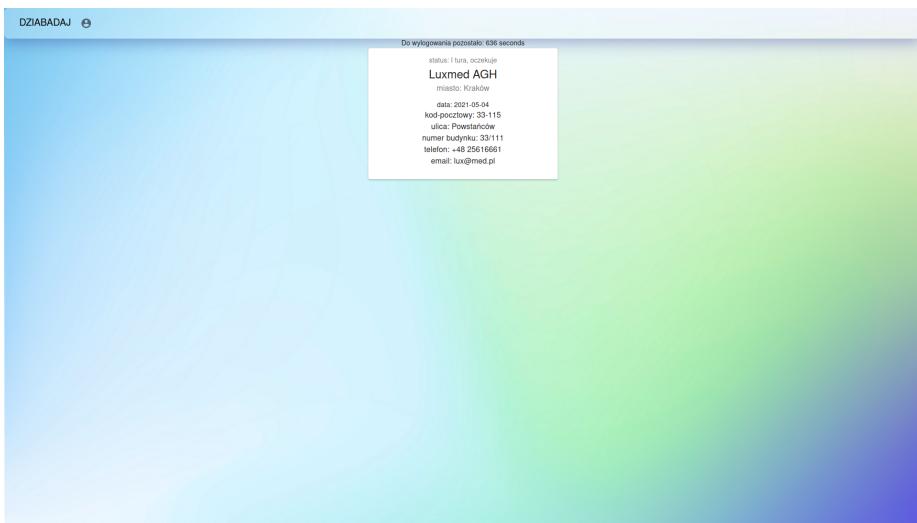
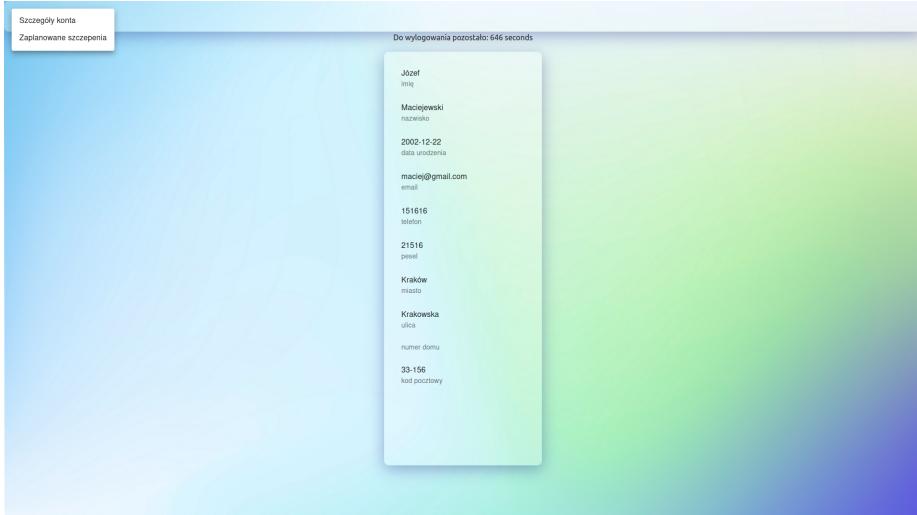
DZIABADAJ LOGIN

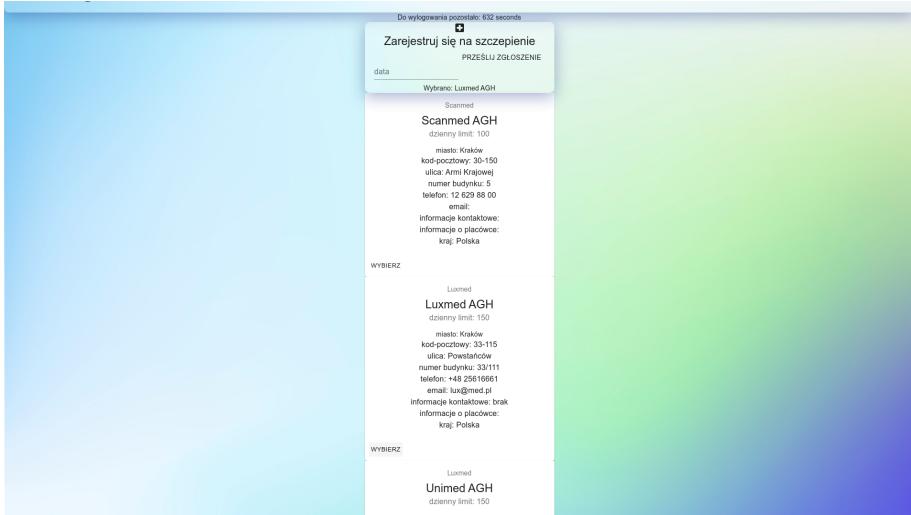
Zarejestruj się na szczepienie

imię _____
nazwisko _____
data urodzenia _____
pesel _____
email _____
telefon _____
miasto/miejscowość _____
ulica _____
numer domu _____
kod pocztowy _____
hasło _____

PRZESŁUJ ZGŁOSZENIE







The screenshot shows a Swagger API documentation page titled "My Title". It includes a "Base URL" field set to "http://localhost:8080/v1/swagger.json". Below the title, there is a "A description" section. The API endpoints are organized into categories:

- resources/patients**: Obsluga pacjenta
- resources/facilities**: Obsługa placówek
- resources/personnel**: Personel szczepiony
- auth**: Authorization and authentication
- resources/vaccines**: Resource
 - POST** /resources/vaccines/create dodaj szczepionkę
 - GET** /resources/vaccines/producer/all Pobierz producenta
 - POST** /resources/vaccines/producer/create dodaj producenta
 - GET** /resources/vaccines/{Id} Pobierz szczepionkę
- resources/clinics**: Resource
 - GET** /resources/clinics/all odbierz przychodnie
 - POST** /resources/clinics/create dodaj przychodnię
- vaccination**: Rejestracja szczepień
 - POST** /vaccination/register Zarejestruj pacjenta

My Title ⓘ
[Base URL: /api/v1]
<http://localhost:1999/api/v1/swagger.json>

A description

resources/patients Obiega pacjenta

GET /resources/patients/all List of patients

Parameters

Name	Description
X-Fields	string(seas) An optional fields mask (header)

Responses

Code	Description	Response content type
200	Success	application/json

Example Value : Model

```
[{"id": "string", "name": "string", "last_name": "string", "date": "2021-02-12", "email": "string", "phone": "string", "country": "string", "city": "string", "street": "string", "zip": "string", "country": "string", "patient_info": "string"}]
```

My Title ⓘ
[Base URL: /api/v1]
<http://localhost:1999/api/v1/swagger.json>

A description

resources/patients Obiega pacjenta

GET /resources/patients/all List of patients

Parameters

Name	Description
X-Fields	string(seas) An optional fields mask (header)

Responses

Code	Description	Response content type
200	Success	application/json

Example Value : Model

```
[{"id": "string", "name": "string", "last_name": "string", "date": "2021-02-12", "email": "string", "phone": "string", "country": "string", "city": "string", "street": "string", "zip": "string", "country": "string", "patient_info": "string"}]
```

5.3 Opracowanie dokumentacji technicznej

Dokumentacja api SwaggerUI, dostępna jest pod adresem: <http://localhost:1999/api/v1>

5.4 Wykaz literatury

1. <https://www.postgresqltutorial.com/>
2. [postgresql.org/](https://www.postgresql.org/)
3. medium.com

4. <https://hackersandslackers.com/>
5. flask-restx.readthedocs.io/
6. <https://www.psycopg.org/docs/>