



ZASTOSOWANIE INFORMATYKI W GOSPODARCE

Rejestrator szczepień

Autorzy:

Radosław LIS 241385
Mikołaj KAMIŃSKI 241388
Mateusz KRAWCZAK 241318
Krystian KANIA 242014
Kamil KUCHARSKI 235014
Filip SOŁDRZYŃSKI 241385

Prowadzący:

dr inż.
Dariusz CABAN

22 czerwca 2021

Spis treści

1 Opis projektu	3
1.1 Cel i zakres	3
1.2 Wymagania funkcjonalne	3
1.3 Wymagania niefunkcjonalne	4
1.4 Baza danych	5
1.5 Technologie	6
1.6 Rozdział prac	6
2 Implementacja	6
2.1 Aplikacja serwera	6
2.1.1 Rozdział ról	6
2.1.2 Rejestracja obywatela w systemie	7
2.1.3 Rejestracja szpitala	9
2.1.4 Logowanie do systemu	9
2.1.5 Informacja o liczbie zaszczepionych	10
2.1.6 Szczepienia do wyświetlenia obywatelowi	10
2.1.7 Szczepienia do wyświetlenia szpitalowi	10
2.1.8 Zapis na szczepienie z poziomu obywatela	11
2.1.9 Zapis na szczepienie z poziomu szpitala	11
2.1.10 Wprowadzanie szczepionek do systemu	11
2.2 Aplikacja klienta	12
2.2.1 Ekran powitalny	12
2.2.2 Rejestracja pacjenta	12
2.2.3 Wybór placówki	13
2.2.4 Rejestracja pacjenta w szpitalu	14
2.2.5 Szczepienia	14
2.2.6 Szczepionki	15
2.2.7 Panel admina	16
3 Instrukcja użytkownika	17
3.1 Administrator systemowy	17
3.1.1 Dodawanie szpitala	17
3.2 Dodawanie firmy	18
3.3 Pacjent	19
3.3.1 Główne menu - ekran powitalny	19
3.3.2 Menu po zalogowaniu - zapisane szczepienie	20
3.3.3 Menu po zalogowaniu - wybór placówki	21
3.4 Personel medyczny	22
3.4.1 Zarządzanie szczepionkami	22
3.4.2 Zarządzanie pacjentami	23
3.4.3 Zarządzanie szczepieniami	24
3.4.4 Zarządzanie kadrą lekarzy	25

4 Instrukcja wdrożeniowa	25
4.1 Wymagania	25
4.1.1 Ogólne	25
4.1.2 Frontend	25
4.1.3 Backend	26
4.1.4 Skrypt konfiguracyjny	26
4.1.5 Konfiguracja bazy danych	26
4.2 Uruchomienie serwisów	27

1 Opis projektu

1.1 Cel i zakres

Celem jest utworzenie pełnoprawnego systemu szczepień mającego na celu zastąpić obecny rządowy system. W skład prac zaliczać się będzie utworzenie kilku ról (m.in. szpitala oraz obywatela). W skład funkcjonalności obywatela będzie wchodzić między innymi zapis na szczepienia oraz możliwość ich odwołania, a ze strony szpitala możliwość wprowadzania szczepionek do systemu i zarządzanie szczepieniami.

1.2 Wymagania funkcjonalne

Tabela 1 przedstawia najważniejsze wymagania funkcjonalne aplikacji. Wymagania zostały przedstawione wraz z uzasadnieniem oraz kryterium spełnienia.

Tablica 1: Wymagania funkcjonalne

Nr	Opis	Przesłanka	Kryterium spełnienia
1	Bezpieczny przepływ oraz składowanie danych	Aplikacja obsługuje wrażliwe dane takie jak pesele, hasła i inne dane osobowe. Ze względu na to, aplikacja musi być dobrze zabezpieczona	Zapewnienie walidacji danych wejściowych, szyfrowanie, zabezpieczenie przed typowymi atakami, wykorzystywanie tokenów poprzez aplikację
2	Rejestracja na szczepienia przez pacjenta	Głównym celem aplikacji jest zapewnienie użytkownikowi możliwości zapisu na szczepienie	Interfejs użytkownika zapewnia możliwość zapisu na szczepienia wraz z widocznym panelem wizyt
3	Rejestracja na szczepienia przez szpital	Ponieważ nie każda osoba posiada urządzenie zdolne korzystać z aplikacji rejestracji pacjentów powinien dokonywać szpital do którego pacjent jest przydzielony	Interfejs użytkownika zapewnia możliwość zapisu na szczepienia pacjentów poprzez konto należące do szpitala, ponadto w przypadku zalogowania się tych pacjentów na swoje konta, umówione wizyty są widoczne w odpowiednim panelu
4	System powiadomień dla pacjenta	Celem aplikacji jest zadbanie aby klient był na bieżąco informowany o wydarzeniach związanych z aplikacją takich jak termin wizyty, przypomnienie hasła czy odebranie kodu weryfikacyjnego	Użytkownik otrzymuje wiadomości na temat aktualnego stanu spraw poprzez SMS lub e-mail

5	Tworzenie szpitala poprzez administratora	Ze względu na to, że nowe placówki mogą wyrazić zgodę na dołączenie do aplikacji musi być możliwość tworzenia w aplikacji nowego szpitala	Administrator jest w stanie utworzyć szpital ze wszystkimi niezbędnymi właściwościami. Żaden inny typ użytkownika nie może mieć praw do tej operacji
6	Dodawanie szczepionek poprzez placówkę	Ilość dostępnych szczepionek jest zależna od tego, jaką liczbę preparatów posiada szpital	Szpital ma możliwość wprowadzania nowych szczepionek do systemu
7	Odwołanie szczepienia przez użytkownika	W celu nie marnowania wizyt, na których nie zjawi się pacjent system musi zadbać o możliwość odwołania wizyty w nagłych wypadkach	Po umówieniu wizyty pacjent ma możliwość odwołania wizyty samodzielnie bądź za pośrednictwem szpitala otrzymując dodatkowo powiadomienie poprzez wybrane medium
8	Tworzenie wolnych slotów na wizyty poprzez algorytm	Aby zautomatyzować sposób tworzenia wizyt, aplikacja musi posiadać algorytm zdolny kojarzyć terminy z lekarzami, którzy w danym terminie są dostępni	Algorytm samoistnie układają nowe wolne sloty na wizyty reagując również na takie zdarzenia jak zajętość lekarzy, czy odwoływanie wizyt
9	Użytkownik wybiera placówkę korzystając z interaktywnej mapy	W celu ułatwienia użytkownikowi wybrania najodpowiedniejszej dla siebie placówki powinna być udostępniana interaktywna mapa z oznaczonymi placówkami	Użytkownik może wybrać placówkę wykorzystując interaktywną mapę

1.3 Wymagania niefunkcjonalne

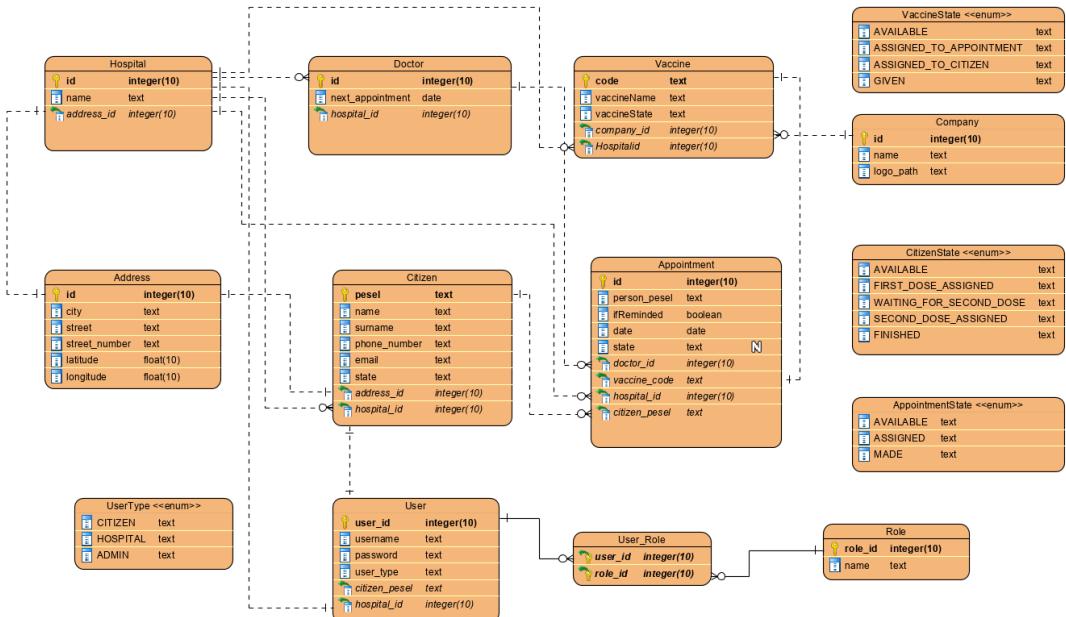
Tabela 2 przedstawia wymagania niefunkcjonalne aplikacji. Podobnie jak w przypadku wymagań funkcjonalnych zostały one przedstawione z uzasadnieniem i kryterium spełnienia.

Tablica 2: Wymagania niefunkcjonalne

Nr	Opis	Przesłanka	Kryterium spełnienia
1	Intuicyjny interfejs użytkownika	Wygląd interfejsu powinien jasno wskazywać jak poruszać się po aplikacji.	Użytkownik z łatwością korzysta z funkcjonalności aplikacji.

2	Responsywność aplikacji.	Aplikacja musi wyglądać dobrze niezależnie od wymiarów ekranu na jakim ją uruchomiono.	Uruchomienie aplikacji na różnych urządzeniach nie powoduje negatywnych zniekształceń komponentów.
3	Interfejs użytkownika zawiera wyłącznie słowa w języku polskim.	Aplikacja jest przeznaczona dla użytkowników używających języka polskiego.	W aplikacji znajdują się jedynie słowa w języku polskim.
4	Krótki czas oczekiwania na wczytanie się kolejnych części systemu	Szybkie uruchamianie poszczególnych składowych systemu ma kluczowe znaczenie w odbiorze aplikacji.	Czas oczekiwania na uruchomienie kolejnej aktywności nie może wynosić więcej niż dwie sekundy.
7	Aplikacja nie zawiera reklam.	Zbyt duża ilość reklam negatywnie wpływa na odbiór aplikacji.	Brak reklam w trakcie działania aplikacji.
8	Dobra kolorystyka.	Pojawiające się kolory powinny sprzyjać dobremu humorowi.	Brak zastrzeżeń użytkowników dotyczących kolorystyki.

1.4 Baza danych



Rysunek 1: Diagram encji bazy danych wykorzystywanej w aplikacji.

Rysunek 1 przedstawia diagram encji bazy danych wykorzystywanej w naszej aplikacji. Baza danych składa się z 10 tabel powiązanych relacjami oraz 4 encji będących enumeracją. Baza danych

w końcowej formie spełnia warunki trzeciej postaci normalnej uważanej za wystarczającą dla większości projektów informatycznych. Główne cechy takiej bazy danych to opisywalność przez encję jednego obiektu, atomowość atrybutów oraz żaden atrybut niekluczowy nie jest zależny funkcjonalnie od innych atrybutów niekluczowych.

Bezpieczeństwo wykorzystywanej przez nas bazy danych polega na walidacji danych wejściowych, głównie za pomocą adnotacji frameworka Spring Boot oraz constraintów nałożonych na konkretne atrybuty w bazie. Dostęp do bazy jest oczywiście chroniony silnym hasłem. Dane wrażliwe takie jak hasła są zaszyfrowane funkcją BCrypt.

1.5 Technologie

- **back-end** - Java, Spring Boot,
- **front-end** - Angular,
- **projekt** - Trello,
- **komunikacja** - Discord

1.6 Rozdział prac

- Radosław Lis - back-end, implementacja logiki systemu,
- Mikołaj Kamiński - front-end, implementacja części wizualnej, obsługiwanie zapytań z back-endu,
- Kamil Kucharski - utworzenie rządowego systemu obywateli,
- Mateusz Krawczak - front-end, implementacja części wizualnej, obsługiwanie zapytań z back-endu,
- Filip Sołdrzyński - serwery i CI/CD,
- Krystian Kania - back-end, implementacja logiki systemu.

2 Implementacja

2.1 Aplikacja serwera

2.1.1 Rozdział ról

W systemie zostały przewidziane 3 role z następującymi prawami:

- admin - dodawanie firm ze szczepionkami oraz rejestracja szpitali,
- szpital - dodawanie szczepionek do systemu, rejestracji obywateli w systemie i na szczepienia, odwoływanie szczepień, oznaczanie statusu szczepień
- obywatel - przeglądanie szpitali wraz z dostępnymi wizytami, możliwość zapisu na szczepienie oraz możliwość jego odwołania.

2.1.2 Rejestracja obywatela w systemie

Aby zarejestrować obywatela w systemie należy wysłać żądanu typu POST na punkt końcowy `/api/v1/auth/registration/citizen/notify` z takim ciałem:

```
{  
    "pesel": "123456789",  
    "communication_channel_type": 1  
}
```

Gdzie `communication_channel_type` jest 0 dla weryfikacji SMS-owej i 1 dla weryfikacji mailowej. Jeśli okaże się, że osoba z tym peselem jest już zarejestrowana to serwer odsyła następujące ciało:

```
{  
    "timestamp": "29-03-2021 03:10:23",  
    "exception": "UserAlreadyRegisteredException",  
    "message": "Person with that pesel is already registered"  
}
```

Jeżeli okaże się, że osoba z takim PESEL-em nie istnieje to serwer zwróci taką wiadomość:

```
{  
    "timestamp": "29-03-2021 04:12:56",  
    "exception": "PeselDoesNotExistException",  
    "message": "Person with pesel '123456789' does not exists!"  
}
```

Jeśli zaś okaże się, że istnieje osoba z takim PESEL-em, ale nie obsługuje wybranego kanału komunikacji to serwer zwróci następującą wiadomość:

```
{  
    "timestamp": "29-03-2021 07:52:34",  
    "exception": "NotSupportedCommunicationChannelException",  
    "message": "Person with pesel '123456789' has no phone number  
               assigned!"  
}
```

Jeżeli wszystko pójdzie pomyślnie to serwer zwróci następujące ciało:

```
{  
    "verify_api_path": "/api/v1/auth/registration/citizen/verify?  
                      token=UXkMwFQgKDRsdrk8vBp95VjmceabG0"  
}
```

Następnie należy wysłać żądanie POST na punkt końcowy z uzyskanego ciała z kodem wpisanym przez użytkownika i dostarczonym do niego za pomocą wybranego kanału komunikacji:

```
{  
    "registration_code": "123456"  
}
```

Jeżeli kod jest niepoprawny to serwer zwraca następującą wiadomość:

```
{  
    "timestamp": "29-03-2021 03:06:28",  
    "exception": "IncorrectRegistrationCodeException",  
    "message": "Registration code is incorrect"  
}
```

Jeżeli zaś kod jest poprawny, ale upłynął czas na jego podanie (domyślnie 60 sekund) to serwer zwraca następującą wiadomość:

```
{  
    "timestamp": "29-03-2021 03:07:22",  
    "exception": "RegistrationCodeExpiredException",  
    "message": "Token expired, cause of 191s > 60s"  
}
```

Jeżeli kod jest zgodny z wygenerowanym przez serwer, to odsyłane są w odpowiedzi dane obywatela oraz kolejny punkt końcowy:

```
{  
    "person": {  
        "name": "Jan",  
        "surname": "Kowalski",  
        "pesel": "123456789",  
        "email": "janek@gmail.com",  
        "phone_number": null  
    },  
    "register_api_path": "/api/v1/auth/registration/citizen/  
                        register?token=SS9sbDZR7otkblw88fo0q0QmmIdkXr"  
}
```

Następnie należy wysłać żądanie POST na uzyskany punkt końcowy z następującym ciałem, składającym się z danych wpisanych przez obywatela:

```
{  
    "password": "123456",  
    "username": "testuser",  
    "city": "Krakow",  
    "street": "Wroclawska",  
    "street_number": "19"  
}
```

Jeżeli którykolwiek z tokenów okaże się niepoprawny to serwer odeśle następującą wiadomość:

```
{  
    "timestamp": "29-03-2021 03:08:24",  
    "exception": "TokenDoesNotExistsException",  
    "message": "There is no such a token"  
}
```

Jeśli zaś nazwa użytkownika okaże się zajęta to serwer odeśle następującą wiadomość:

```
{  
    "timestamp": "29-03-2021 03:15:20",  
    "exception": "UsernameNotAvailableException",  
    "message": "Username 'admin' is in use!"  
}
```

Jeżeli zaś wszystko przebiegnie poprawnie to serwer odeśle jedynie status z kodem 200 bez żadnego ciała.

2.1.3 Rejestracja szpitala

Aby zarejestrować placówkę w systemie należy z poziomu administratora wywołać żądanie POST na punkt końcowy `/api/v1/auth/registration/hospital/register` o następującym ciele:

```
{  
    "password": "password",  
    "username": "szpital_wro",  
    "hospital_name": "szpital we wrocławiu",  
    "city": "Wrocław",  
    "street": "Grunwaldzka",  
    "street_number": "12c"  
}
```

2.1.4 Logowanie do systemu

Logowanie do systemu jest takie samo dla dowolnego typu użytkownika i polega na wysłaniu żądania POST na punkt końcowy `/api/v1/auth/login` z ciałem zawierającym login oraz hasło:

```
{  
    "password": "adminpassword",  
    "username": "admin"  
}
```

Jeżeli dane logowania są niepoprawne to serwer zwraca następującą wiadomość:

```
{  
    "timestamp": "29-03-2021 03:08:24",  
    "exception": "AuthenticationException",  
    "message": "Incorrect credentials"  
}
```

Jeżeli zaś dane logowania są poprawne to serwer odsyła ciało zawierające token przeznaczony do wysyłania kolejnych żądań w zależności od roli:

```
{  
    "access_token": "eyJhbGciOiJIUzUxMiJ9.eyJzdWIiOiJhZG1pbisImlhd  
    CI6MTYxNzAyMzY0OCwiZXhwIjoxNjE3MTEwMDQ4fQ.un27  
    rc4DzILY9Dd-X-xYrJUiJIKXszxA6kMXGennKt96PwcyuNi  
    oYXLn46KGkqu9HIIxo1NIT1DFAKxBKqkeA",  
    "type": "Bearer",  
    "username": "admin",  
    "authorities": [  
        {  
            "authority": "ROLE_HOSPITAL"  
        },  
        {  
            "authority": "ROLE_ADMIN"  
        },  
        {  
            "authority": "ROLE_CITIZEN"  
        }  
    ]  
}
```

2.1.5 Informacja o liczbie zaszczepionych

Aby uzyskać liczbę zaszczepionych obywateli należy wysłać żądanie typu GET na `/api/v1/citizens/vaccinated` i serwer zwraca następującą wiadomość:

```
{  
    "count": "2424"  
}
```

2.1.6 Szczepienia do wyświetlenia obywatelowi

Aby uzyskać wszystkie wolne terminy szczepień to należy wysłać żądanie typu GET na punkt końcowy `/api/v1/hospitals/{id}/appointments?page=0&size=2` odpowiednio ustawiając ID szpitala oraz parametry paginacji, tak żeby dostosować liczbę wyświetlanych wizyt do rozmiaru strony.

2.1.7 Szczepienia do wyświetlenia szpitalowi

Aby uzyskać wybrane szczepienia to należy wysłać żądanie typu GET na punkt końcowy `/api/v1/users/self/appointments`, podając opcjonalnie parametry paginacji, a szpital jest wyciągany przez system z kontekstu.

Można również zdefiniować o jakim statusie szczepienia mają być zwrócone wysyłając żądanie na `/api/v1/users/self/appointments?available=false&made=true&assigned=true`. Domyślnie wszystkie trzy parametry są na `true` i nie trzeba ich wszystkich wymieniać.

2.1.8 Zapis na szczepienie z poziomu obywatela

Obywatel może zostać zapisany na szczepienie po wysłaniu żądania typu PATCH na punkt końcowy `/api/v1/appointments/{id}/actions/enroll` z wybranym ID szczepienia.

2.1.9 Zapis na szczepienie z poziomu szpitala

Obywatel może zostać zapisany na szczepienie przez szpital po wysłaniu żądania typu POST na punkt końcowy `/api/v1/appointments/{id}/hospital/actions/enroll` z wybranym ID szczepienia i następującym ciałem zawierającym PESEL obywatela:

```
{  
    "pesel": "56111245968"  
}
```

2.1.10 Wprowadzanie szczepionek do systemu

Szczepionki mogą zostać wprowadzone do systemu poprzez wysłanie żądania typu POST na punkt końcowy `/api/v1/users/self/vaccines` będąc zalogowanym na szpital i o następującym ciele, które ma strukturę listy obiektów JSON:

```
[  
    {  
        "code": "3hj563f6a5",  
        "company_name": "AstraZeneca"  
    },  
    ...  
    {  
        "code": "7f4547jko",  
        "company_name": "Johnson&Johnson"  
    }  
]
```

Reszta dostępnych punktów końcowych możliwa jest do uzyskania po uruchomieniu serwisu backend'owego i wejściu na porcie 8080 na ścieżkę `/swagger-ui/`. Są tam pokazane wszystkie punkty końcowe wraz z odpowiednią dokumentacją.

2.2 Aplikacja klienta

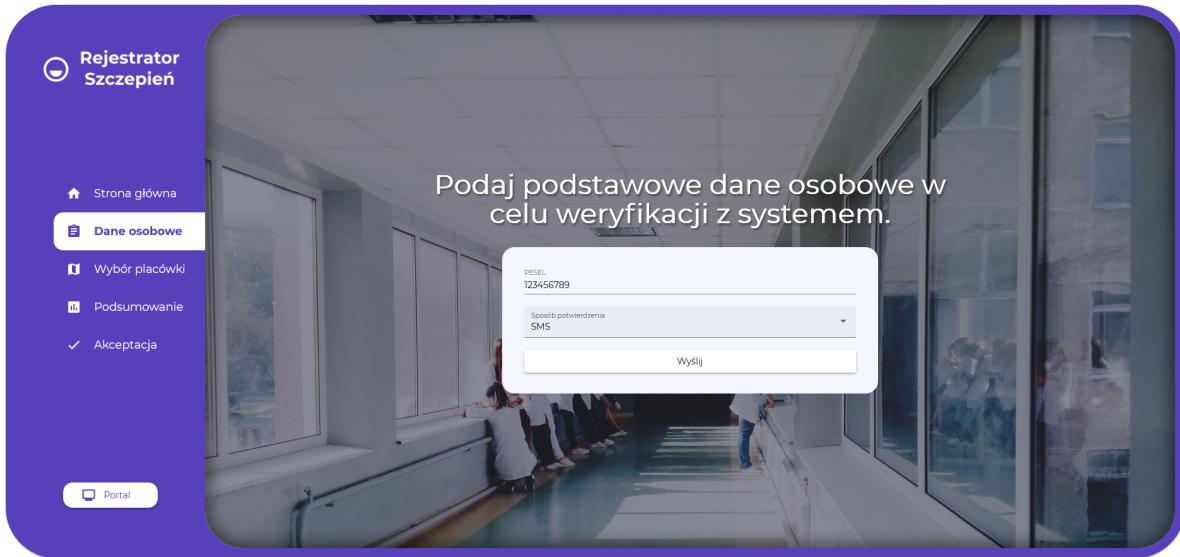
2.2.1 Ekran powitalny

Ekran powitalny zawiera aktualności dotyczące COVID-19, aktualna liczbę zachorowań, liczbę zarejestrowanych pacjentów oraz szpitali w aplikacji. Dodatkowo umieszczona została krótka instrukcja obsługi systemu.



2.2.2 Rejestracja pacjenta

Użytkownik rejestruje się w systemie poprzez podanie numeru PESEL oraz wybraniu opcji potwierdzenia tożsamości. Po potwierdzeniu użytkownik może zalogować się do systemu.



2.2.3 Wybór placówki

Użytkownik może wybrać szpital, w którym chce się zaszczepić. Po wybraniu placówki użytkownikowi ukazuje się lista z dostępnymi terminami szczepień.



2.2.4 Rejestracja pacjenta w szpitalu

Szpital może rejestrować nowych użytkowników w swoim szpitalu. Dodatkowo może podglądać szczepienia danego pacjenta oraz przypisać nowe.

Baza pacjentów

Pesel pacjenta
9910323923

Rejestracja nowego użytkownika Szukaj

Dane pacjenta

Imię: Jan
Nazwisko: Kowalski
Email: janek@gmail.com
PESEL: 9910323923
Nr telefonu: 789456123
Miejsce: Kraków
Ulica: Kazimierza Wielkiego

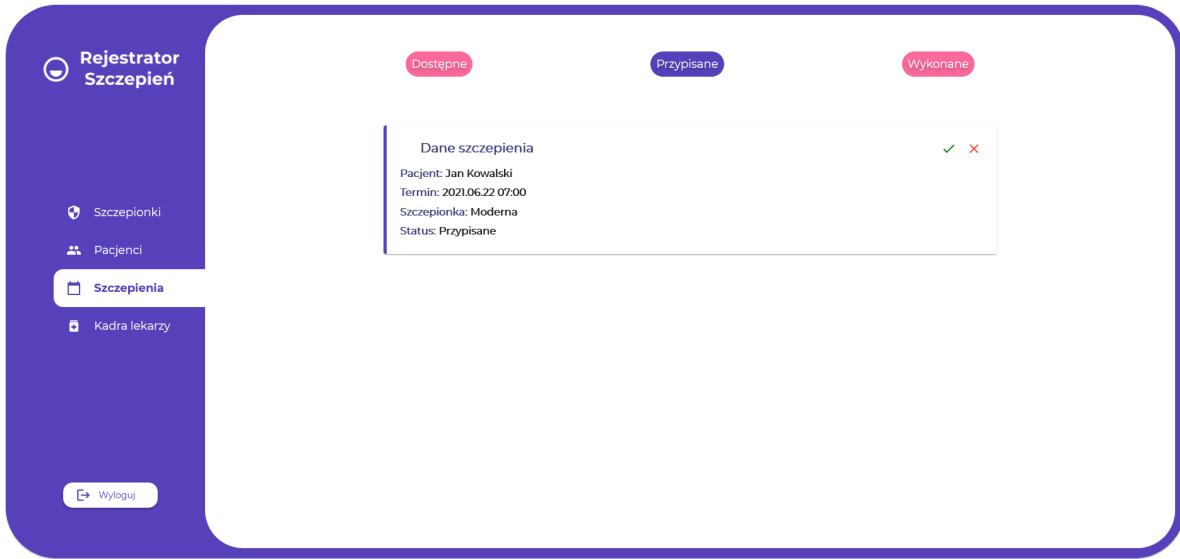
Terminy szczepień pacjenta

Dane szczepienia

Pacjent: Jan Kowalski
Termin: 2021.06.22 07:00
Szczepionka: Moderna
Status: Przypisane

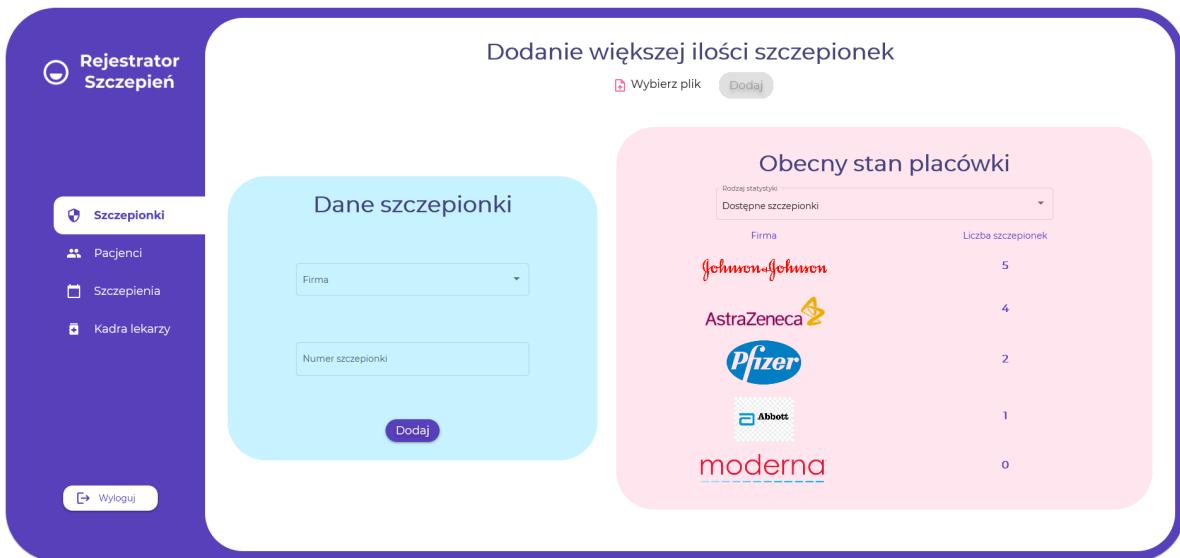
2.2.5 Szczepienia

Szpital ma możliwość sprawdzenia ilości dostępnych szczepień, przypisanych szczepień oraz wykonanych szczepień w swojej placówce.



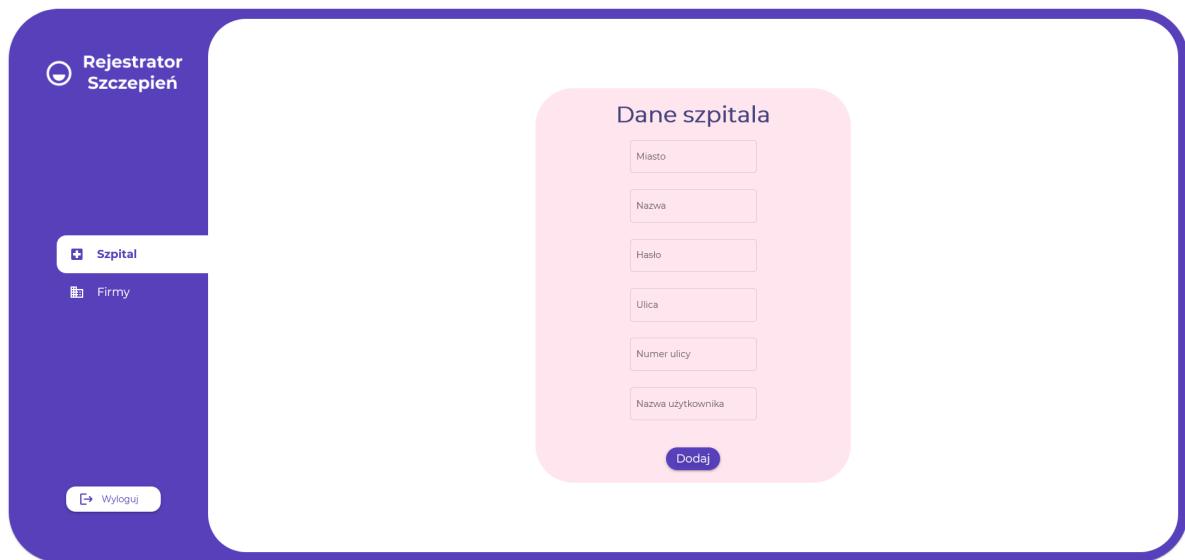
2.2.6 Szczepionki

Szpital może dodawać szczepionki do swojej placówki. Można to zrobić pojedynczo wybierając firmę i wpisując kod szczepionki lub przesyłać plik z większą ilością szczepionek. Dodatkowo można sprawdzić aktualny stan szczepionek w szpitalu.



2.2.7 Panel admina

Admin jest odpowiedzialny za dodawanie szpitali oraz firm do systemu. Na poniższym przykładzie, admin wypełnia formularz, który zawiera dane szpital i dodaje go do systemu.

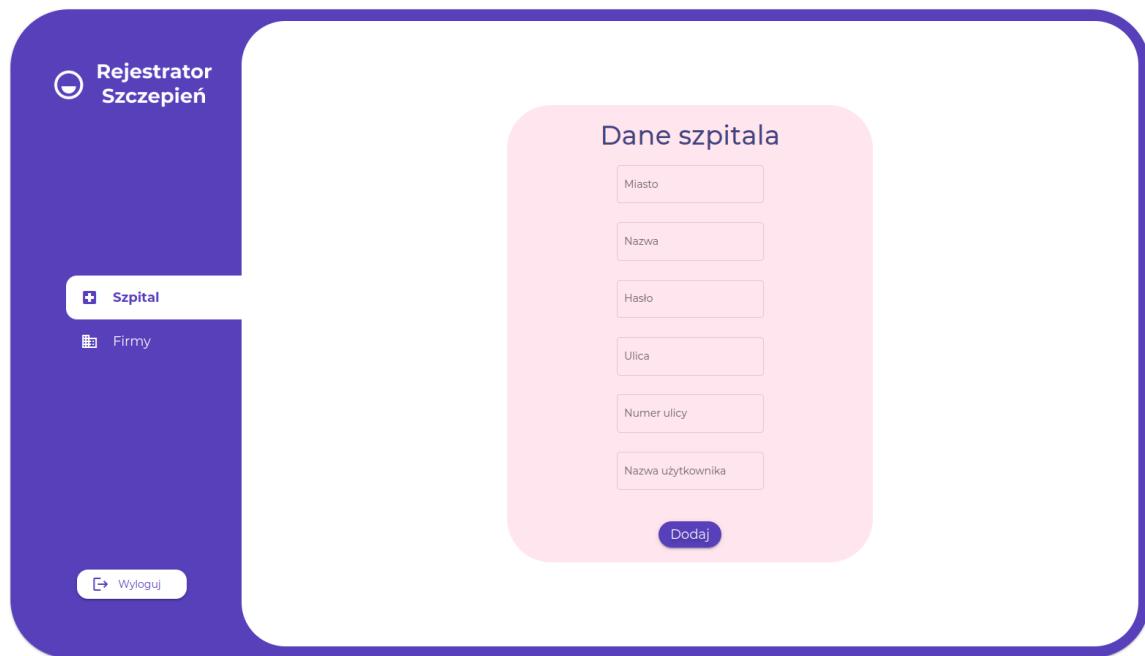


3 Instrukcja użytkownika

3.1 Administrator systemowy

3.1.1 Dodawanie szpitala

Jako administrator, w zakładce „Szpital” na panelu bocznym, po wypełnieniu formularza podając wszystkie wymagane dane mamy możliwość dodania do systemu nowego szpitala.



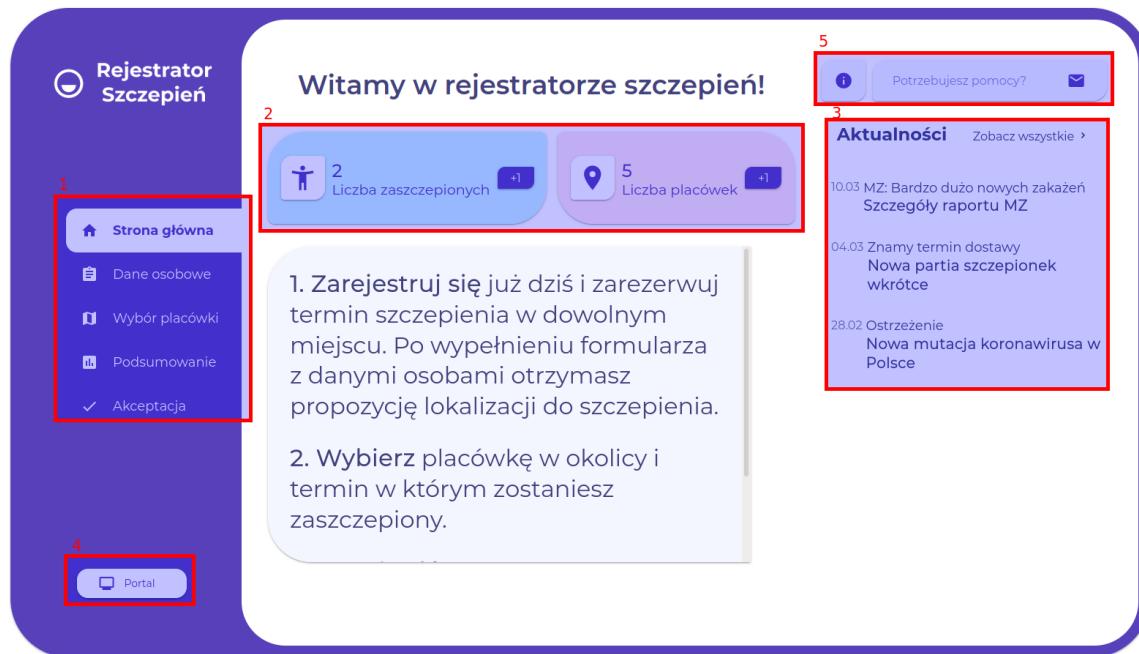
3.2 Dodawanie firmy

Jako administrator, w zakładce „Firmy” na panelu bocznym, po wypełnieniu formularza podając wszystkie wymagane dane (tj. nazwy firmy oraz jej loga) mamy możliwość dodania do systemu nowej firmy, jako dostawcy szczepionek.



3.3 Pacjent

3.3.1 Główne menu - ekran powitalny

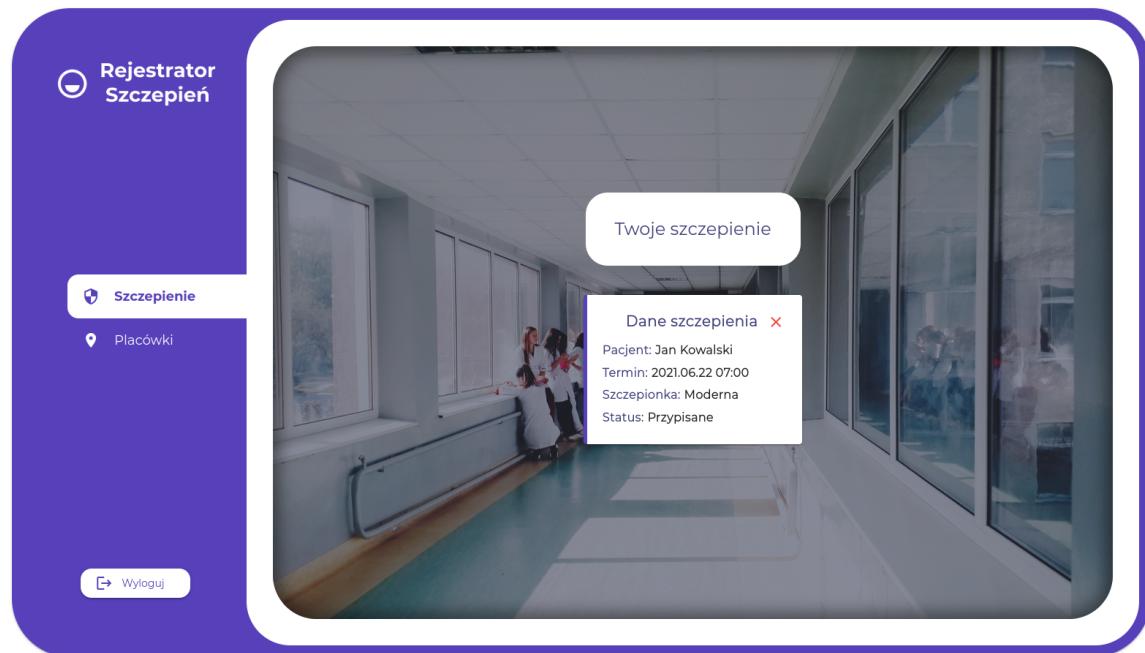


Na ekranie powitalnym użytkownik ma dostęp do następujących modułów:

- 1 - Panel boczny - ma panelu bocznym znajduje się nawigacja po systemie dla niezalogowanego użytkownika.
- 2 - Informacje statystyczne o liczbie zaszczepionych już pacjentów oraz liczbie placówek umożliwiających szczepienia.
- 3 - Aktualności związane ze szczepieniami oraz pandemią COVID-19
- 4 - Logowanie do serwisu
- 5 - Możliwość skontaktowania się z obsługą serwisu w razie problemów.

3.3.2 Menu po zalogowaniu - zapisane szczepienie

Jako użytkownik możemy zobaczyć potwierdzenie umówionego terminu szczepienia w zakładce „Szczepienie” gdzie mamy wgląd do naszego szczepienia w tym nazwiska pacjenta, terminu, producenta szczepionki oraz obecnego statusu szczepienia.



3.3.3 Menu po zalogowaniu - wybór placówki



Aby wybrać placówkę gdzie chcemy się zaszczepić mamy do wyboru moduły:

- 1 - panel boczny - nawigacja
- 2 - mapa do wyboru placówki przy pomocy kliknięcia na marker na mapie
- 3 - dostępne wolne terminy szczepienia w wybranej wcześniej placówce
- 4 - wylogowanie z serwisu

3.4 Personel medyczny

3.4.1 Zarządzanie szczepionkami

The screenshot shows the 'Szczepionki' section of the medical staff interface. On the left, a sidebar menu has item 1 (Szczepionki) highlighted with a red box. The main area contains a sub-menu with items 2 (Wybierz plik) and 3 (Dodaj), also highlighted with a red box. A large central panel displays 'Dane szczepionki' with fields for Firma (Johnson & Johnson, AstraZeneca, Pfizer, Abbott, moderna) and Numer szczepionki. To the right, a pink box titled 'Obecny stan placówki' shows a list of available vaccines with their counts: Johnson & Johnson (5), AstraZeneca (4), Pfizer (2), Abbott (1), and moderna (0). A dropdown menu above this list is also highlighted with a red box.

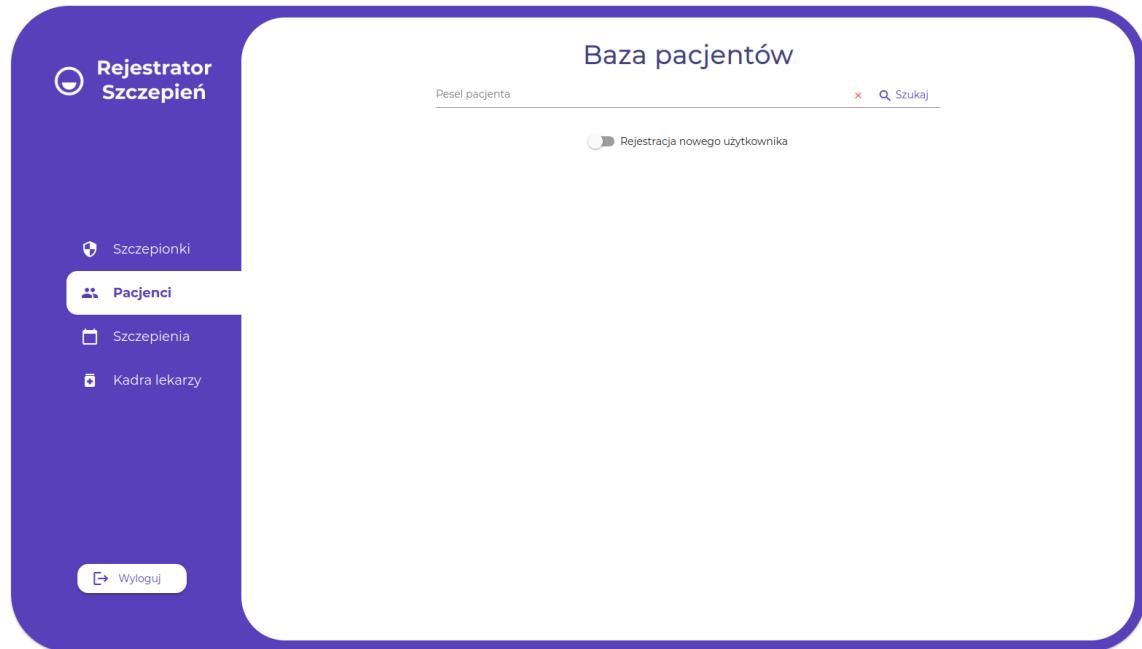
Firma	Liczba szczepionek
Johnson & Johnson	5
AstraZeneca	4
Pfizer	2
Abbott	1
moderna	0

Użytkownik jako personel medyczny może dodawać szczepionki, przeglądać ich stan w placówce, miećgląd do pacjentów zarejestrowanych w placówce oraz szczepień a także dodawać nowych lekarzy.

- 1 - panel boczny - nawigacja
- 2 - możliwość dodawania większej ilości szczepionek przy pomocy przygotowanego wcześniej pliku z danymi wejściowymi
- 3 - stan szczepionek w placówce wraz z możliwością sortowania po: dostępnych szczepionkach, przypisanych już do pacjentów oraz tych już wykorzystanych

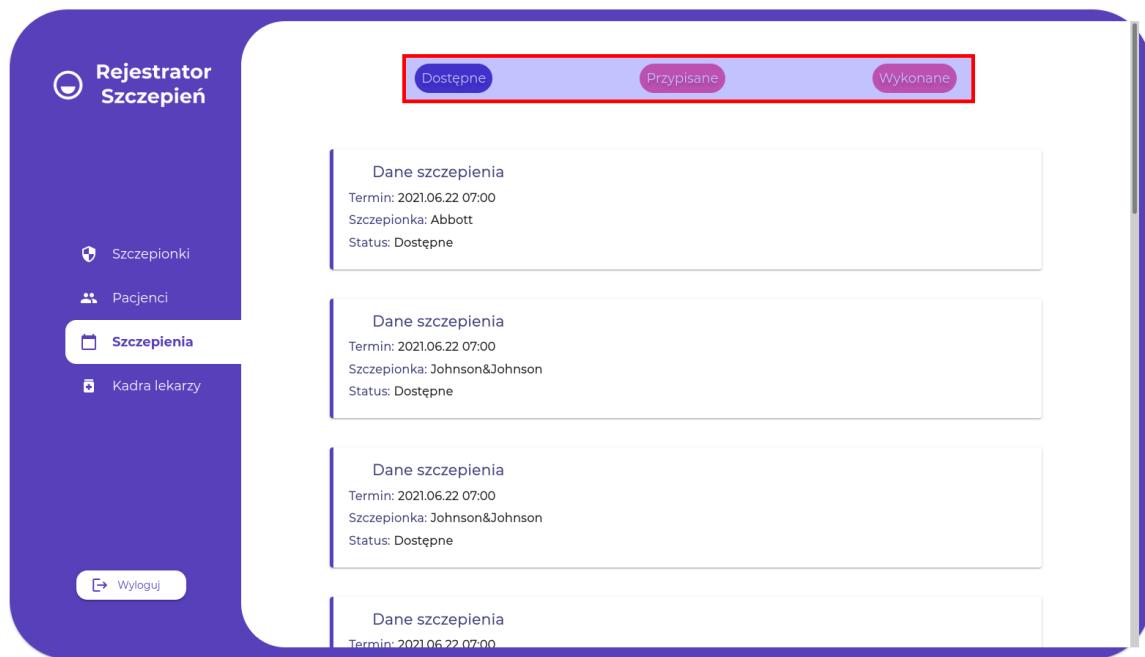
3.4.2 Zarządzanie pacjentami

Personel medyczny może wyszukiwać pacjentów po numerze PESEL a także po przesunięciu suwaka „Rejestracja nowego użytkownika” dodawać pacjentów bezpośrednio do systemu bez konieczności rejestracji przez nich osobiście.



3.4.3 Zarządzanie szczepieniami

Personel medyczny widzi wszystkie szczepionki, które zostały kiedykolwiek zarejestrowane w placówce. Przy pomocy menu górnego w postaci przycisków możemy wyświetlić wszystkie dostępne szczepienia, te które zostały już przypisane oraz te wykonane dotychczas w placówce



3.4.4 Zarządzanie kadrą lekarzy

W zakładce „Kadra lekarzy” personel medyczny może dodawać nowych lekarzy, którzy dołączyli do placówki.



4 Instrukcja wdrożeniowa

4.1 Wymagania

Zostaną tu przedstawione wymagania co do uruchomienia systemu, najlepszym rozwiązaniem jest uruchomienie skryptu konfiguracyjnego z sekcji 4.1.4.

4.1.1 Ogólne

- maszyna wirtualna z systemem Ubuntu,
- Docker 20.10.5.

4.1.2 Frontend

- NodeJS v14.17.1.
- angular/cli v12.05.
- Angular Material 12.04.

4.1.3 Backend

- Java 11,
- Maven 3.8.1,
- MySQL 5.7.

4.1.4 Skrypt konfiguracyjny

Listing 1: Konfiguracja wymagań i bazy danych

```
$ sudo apt-get update
# wymagania ogólne
$ sudo apt-get install docker-ce docker-ce-cli containerd.io

# wymagania frontendowe
$ npm install -g @angular/cli
$ ng add @angular/material

# wymagania backendowe
$ sudo apt-get install openjdk-11-jdk
$ sudo apt install maven
$ sudo apt install mysql-server
```

4.1.5 Konfiguracja bazy danych

Uzyskanie dostępu do shell'a MySQL:

```
$ mysql -u root -p
```

Utworzenie użytkownika o wybranym loginie i haśle, wartości *your_username* oraz *your_password* należy zamienić na wybrane przez siebie, o odpowiednim poziomie bezpieczeństwa:

```
mysql> CREATE USER 'username'@'host' IDENTIFIED WITH 'your_username',
BY 'your_password';
```

Utworzenie bazy danych:

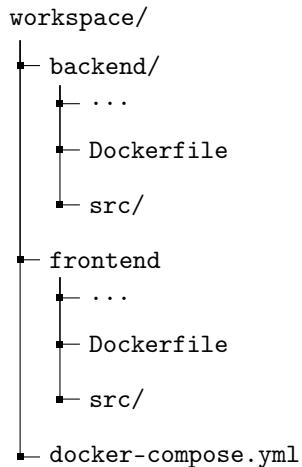
```
mysql> CREATE DATABASE vacc_db;
```

Dodanie uprawnień do bazy danych wcześniej utworzonemu użytkownikowi:

```
mysql> GRANT ALL PRIVILEGES ON *.* TO 'your_username'@'localhost',
WITH GRANT OPTION;
mysql> FLUSH PRIVILEGES;
mysql> exit
```

4.2 Uruchomienie serwisów

Należy utworzyć osobny katalog *workspace* dla kodu serwera, aplikacji klienckiej oraz skryptu Dockera tworząc następującą strukturę:



Plik *docker-compose.yml* ma następującą strukturę uruchamiającą wszystkie serwisy:

```
version: '3'

services:
    angular-service:
        build: ./frontend-pipeline
        ports:
            - 4200:80

    db:
        image: mysql:8.0.19
        ports:
            - 3306:3306
        container_name: mysql_container
        environment:
            - MYSQL_ROOT_PASSWORD=admin12345@@
            - MYSQL_PASSWORD=admin12345@@
            - MYSQL_USER=root
        volumes:
            - dbdata:/var/lib/mysql

    docker_backend:
        image: backend_image:latest
        container_name: backend_container
        environment:
            - OPENCAGEDATA_API_KEY=84aa8a50b45c4ec7b1c9f39164b39521
        ports:
```

```
    - 8080:8080
depends_on:
  - db

volumes:
  dbdata:
```

Uruchomienie serwisów powinno odbyć się w następujący sposób:

```
$ cd backend
$ docker build -t backend_image . && cd ..
$ cd frontend
$ ng build --prod && cd ..
$ docker-compose up -d
```