

Inżynieria Oprogramowania

Specyfikacja projektu

Grupa M

Magdalena Falkowska

Dawid Alimowski

Tymoteusz Bartnik

Bartosz Błachut

Przemysław Dominikowski

Styczeń 2022

Spis treści

1	Opis systemu	4
1.1	Wstęp	4
1.2	Moduły systemu	4
1.3	Wymagania i funkcjonalności	5
1.3.1	Historyjki użytkowników	5
1.3.2	Wymagania użytkowników	6
1.4	Przypadki użycia	12
1.5	Diagram klas	14
1.6	Kluczowe obiekty biznesowe	16
1.7	Diagram stanów	17
1.7.1	Pacjent	17
1.7.2	Oferta szczepienia	18
1.7.3	Szczepienie	19
1.8	Diagram aktywności	20
1.8.1	Proces szczepienia	20
1.8.2	Procedura rejestracji na wizytę	21
1.8.3	Procedura wyszukiwania i przydzielania dostępnego lekarza	21
1.9	Diagram sekwencji	22
1.9.1	Zarządzenie kontem pacjenta	22
1.9.2	Szczepienie	23
1.9.3	Praca lekarza	24
1.9.4	Praca administratora	25
1.9.5	Pobieranie certyfikatu szczepienia	26
1.10	Dokumentacja API	26
2	Algorytmy biznesowe	26
2.1	Założenia algorytmu	26
2.2	Referencyjna implementacja	27
3	Wymagania techniczne	28
3.1	Wymagania systemu	28
3.2	Preferowane technologie	28
4	Testowanie i obsługa błędów	29
4.1	Testowanie	29
4.2	Obsługa błędów	29
4.2.1	Moduł rejestracji	29

4.2.2	Moduł punktu szczepień	29
4.2.3	Moduł administracyjny	30
5	Załączniki	30

1 Opis systemu

1.1 Wstęp

Projektowany system ma służyć do kompleksowej obsługi procesu szczepień. Korzystający z niego pacjent będzie miał możliwość zarejestrowania się w systemie i umówienia się na wizytę w punkcie szczepień w celu odbycia szczepienia na konkretną chorobę. Lekarz pracujący w punkcie szczepień będzie mógł przeprowadzić kwalifikację do szczepienia, przeprowadzić zabieg, a w razie konieczności ustalić również termin następnej wizyty. Administrator sprawuje kontrolę nad przebiegiem procesu szczepień, może edytować zapisy, generować raporty i zgłaszać błędy do działu IT. W systemie działa również algorytm odpowiedzialny za przydzielanie zaplanowanym wizytom pacjentów konkretnych godzin i lekarzy, a także za generowanie grafików pracy lekarzy.

1.2 Moduły systemu

Projektowany system składa się z trzech modułów:

- Moduł rejestracji
- Moduł punktu szczepień
- Moduł systemu administracyjnego

Głównym obiektem służącym do komunikacji i wymiany danych pomiędzy modułami będzie obiekt „osoba” zawierający dane pacjenta oraz informacje o jego szczepieniach (poszczególne wizyty, rodzaje przyjętych szczepionek, ilość przyjętych dawek konkretnej szczepionki).

1.3 Wymagania i funkcjonalności

1.3.1 Historyjki użytkowników

Jako	chciałbym	żeby
Pacjent	zobaczyć listę wolnych terminów dla poszczególnych rodzajów szczepionek	móc zapisać się na szczepienie przeciwko określonej chorobie
	zarządzać swoimi danymi kontaktowymi	otrzymywać potwierdzenia rejestracji i przypomnienia o zbliżającym się szczepieniu
	mieć możliwość uzyskania certyfikatu po szczepieniu	uzyskać potwierdzenie bycia zaszczepionym
Lekarz	mieć dostęp do zaplanowanych wizyt (w tym danych pacjenta)	potwierdzić tożsamość przed szczepieniem i dokonać odpowiednich przygotowań
	mieć możliwość ustalenia nowego terminu szczepienia dla wybranego pacjenta	ustalić inną datę szczepienia w razie negatywnej klasyfikacji, problemów z dostępnością szczepionki lub w przypadku wizyty na kolejną dawkę szczepionki
	mieć możliwość zapisania w systemie statusu szczepienia	potwierdzić fakt przyjęcia szczepionki przez pacjenta
Administrator	mieć dostęp do danych pacjentów	móc potwierdzić status szczepienia na daną chorobę
	mieć dostęp do edycji zapisów	być w stanie poprawić błędne zgłoszenia
	mieć możliwość zgłaszania błędów do działu IT	raportować krytyczne defekty systemu
Algorytm	przydzielić pacjentów zgłoszonych na szczepienie do konkretnych lekarzy	optymalnie ułożyć grafik szczepień na konkretny dzień
	generować grafiki pracy lekarzy	mieli oni wgląd w to, jak będzie wyglądała ich praca danego dnia
	przydzielać pacjentom godziny szczepień	wiedzieli, kiedy mają zjawić się na szczepienie

1.3.2 Wymagania użytkowników

Historyjka:

Jako pacjent chciałbym zobaczyć listę wolnych terminów dla poszczególnych rodzajów szczepionek, aby móc zapisać się na szczepienie przeciwko określonej chorobie.

Funkcjonalne:

1. Czy mogę zobaczyć listę wolnych terminów szczepień przeciwko konkretnej chorobie? [must]
2. Czy mogę zarejestrować się na wybrany termin szczepienia? [must]
3. Czy mogę zmienić wybrany termin? [should]
4. Czy mogę odwołać zaplanowaną wizytę? [should]
5. Czy mogę umówić się do konkretnego lekarza? [could]
6. Czy mogę filtrować wolne terminy względem rodzaju szczepionki? [could]

Niefunkcjonalne:

1. Czy proces rejestracji jest wydajny i obsłuży wiele sesji jednocześnie? [must]
2. Czy proces rejestracji na szczepienie jest intuicyjny w użyciu? [should]
3. Czy mogę zarejestrować się przez Profil Zaufany? [could]

Historyjka:

Jako pacjent chciałbym zarządzać swoimi danymi kontaktowymi, aby otrzymywać potwierdzenia rejestracji i przypomnienia o zbliżającym się szczepieniu.

Funkcjonalne:

1. Czy mogę sprawdzić swoje dane kontaktowe? [should]
2. Czy mogę wprowadzić, zmienić lub usunąć numer telefonu? [must]
3. Czy mogę wprowadzić, zmienić lub usunąć adres e-mail? [must]

Niefunkcjonalne:

1. Czy zapisanie danych kontaktowych wymaga potwierdzenia (w formie kodu wysłanego na podany telefon lub adres e-mail)? [should]
2. Czy weryfikowana jest poprawność wprowadzonych danych? [could]

Historyjka:

Jako pacjent chcę mieć możliwość uzyskania certyfikatu po szczepieniu, żeby uzyskać potwierdzenie bycia zaszczepionym.

Funkcjonalne:

1. Czy mam wgląd w swój certyfikat i mogę sprawdzić poprawność danych? [must]
2. Czy mogę sprawdzić, do kiedy jest ważny mój certyfikat? [must]
3. Czy mogę pobrać certyfikat w formie pliku pdf? [should]

Niefunkcjonalne:

1. Czy generowany plik pdf nadaje się do okazania w celu potwierdzenia odbycia szczepienia? [should]
2. Czy generowany plik pdf ma formę łatwą do złożenia w poręczny format (A6)? [should]
3. Czy generowany plik pdf jest przejrzysty i czytelny nawet dla starszych i niedowidzących osób? [could]

Historyjka:

Jako lekarz chciałbym mieć dostęp do zaplanowanych wizyt (w tym danych pacjenta), aby potwierdzić tożsamość przed szczepieniem i dokonać odpowiednich przygotowań.

Funkcjonalne:

1. Czy mogę sprawdzić wizyty zaplanowane na dzisiejszy dzień? [must]
2. Czy mogę sprawdzić dane pacjenta zapisanego na wybraną wizytę? [must]

Niefunkcjonalne:

1. Czy system zwraca listę wizyt w krótkim czasie (mniejszym od 1 sekundy)? [should]
2. Czy istnieje możliwość ukrycia zakończonych wizyt z dzisiejszego dnia? [should]
3. Czy system pokazuje liczbę pacjentów pozostałych w danym dniu? [could]

Historyjka:

Jako lekarz chciałbym mieć możliwość zapisania w systemie statusu szczepienia, aby potwierdzić fakt przyjęcia szczepionki przez pacjenta.

Funkcjonalne:

1. Czy mogę zapisać przyjęty rodzaj szczepionki? [must]
2. Czy mogę zapisać numer i serię szczepionki? [must]
3. Czy mogę zapisać numer przyjętej dawki? [must]

Niefunkcjonalne:

1. Czy w razie awarii systemu mogę wprowadzić dane z opóźnieniem, po jego naprawie? [must]
2. Czy poprawność wprowadzanych danych jest weryfikowana? [should]
3. Czy część danych jest automatycznie uzupełniania na podstawie danych pacjenta (rodzaj szczepionki, numer dawki)? [could]

Historyjka:

Jako administrator chciałbym mieć dostęp do danych pacjentów, aby móc potwierdzić status szczepienia na daną chorobę.

Funkcjonalne:

1. Czy mogę sprawdzić status szczepienia wybranej osoby przeciwko konkretnej chorobie? [must]
2. Czy mogę wygenerować raporty szczepień (pod względem wieku, płci, chorób, przeciwko którym odbyto szczepienia, rodzaju przyjętych szczepionek)? [should]

Niefunkcjonalne:

1. Czy posiadam wymagane uprawnienia do podglądu statusu szczepienia danej osoby? [must]
2. Czy generowany raport zawiera wykresy i dopracowane czcionki? [should]

Historyjka:

Jako administrator chciałbym mieć dostęp do edycji zapisów, żeby być w stanie poprawić błędne złożenia.

Funkcjonalne:

1. Czy mogę wyświetlać dane o zapisach? [must]
2. Czy mogę edytować dane o zapisach? [must]
3. Czy użytkownik, którego zapis poprawiam, dostanie powiadomienie o tej zmianie? [should]

Niefunkcjonalne:

1. Czy mam odpowiednie uprawnienia do dokonywania zmian w bazie danych zapisów? [must]
2. Czy dokonywanie zmiany w bazie danych zapisów odbywa się bez wystąpień błędów? [must]
3. Czy zapisywanie dokonywanych zmian wymaga dodatkowego potwierdzenia, poprzez kliknięcie przycisku w wyskakującym popup'ie? [could]

Historyjka:

Jako administrator chciałbym mieć możliwość zgłaszania błędów do działu IT, żeby móc składać raport o defektach systemu.

Funkcjonalne:

1. Czy mogę wygenerować raport o zaistniałym błędzie? [must]
2. Czy mogę wysłać wygenerowany raport do odpowiedniego zespołu? [should]
3. Czy mogę w łatwy sposób skontaktować się z działem IT? [could]

Niefunkcjonalne:

1. Czy raport o błędzie generuje się poprawnie? [must]
2. Czy można szybko (w łatwy sposób) skopiować raport / przesłać go dalej? [could]

Historyjka:

Jako algorytm chcę przydzielić pacjentów zgłoszonych na szczepienie do konkretnych lekarzy, żeby optymalnie ułożyć grafik szczepień na konkretny dzień.

Funkcjonalne:

1. Czy mogę wprowadzić ilość lekarzy dostępnych danego dnia? [must]
2. Czy mogę zmienić lekarza w przypadku jego nieobecności i zastępstwa przez inną osobę? [must]
3. Czy w przypadku braku dostępnych lekarzy mogę wyznaczyć pacjentom inny termin szczepienia? [must]

Niefunkcjonalne:

1. Czy interfejs systemu do przydzielania pacjentów i lekarzy jest przejrzysty? [should]
2. Czy system jest intuicyjny w obsłudze? [should]
3. Czy algorytm jest optymalny i wydajny? [should]

Historyjka:

Jako algorytm chcę generować grafiki pracy lekarzy, żeby mieli oni wgląd w to, jak będzie wyglądała ich praca danego dnia.

Funkcjonalne:

1. Czy mogę wygenerować grafik w formie pliku pdf? [should]
2. Czy mogę wprowadzać zmiany w grafiku? [should]
3. Czy mogę generować grafiki dla konkretnych lekarzy? [should]

Niefunkcjonalne:

1. Czy grafik w formie pliku pdf wygląda estetycznie? [should]
2. Czy grafik w formie pliku jest czytelny? [should]
3. Czy algorytm jest optymalny i wydajny (generowanie zajmuje mniej niż 1 sekundę)? [should]

Historyjka:

Jako algorytm chcę przydzielać pacjentom godziny szczepień, żeby wiedzieli, kiedy mają zjawić się na szczepienie.

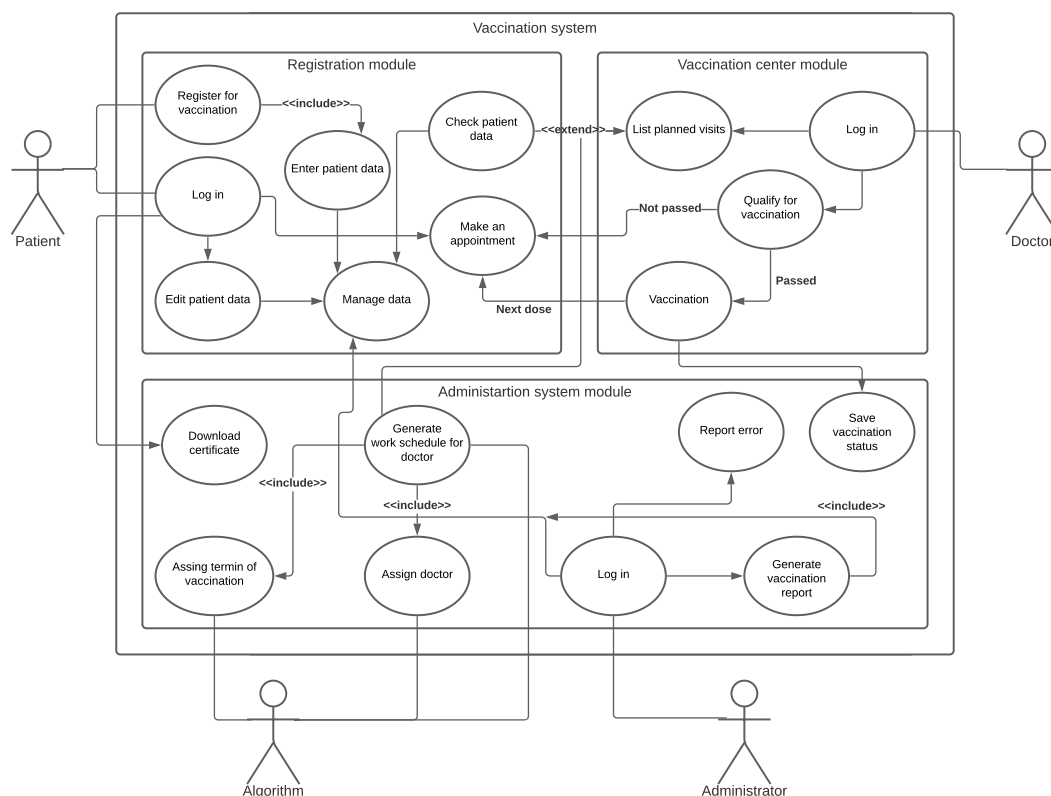
Funkcjonalne:

1. Czy sloty czasowe przydzielane są optymalnie? [should]
2. Czy nie występują sytuacje, gdy dwie osoby zostały zapisane na jednakowy termin? [must]
3. Czy przydzielane są najbliższe możliwe terminy? [should]
4. Czy funkcjonalność przydzielania terminu jest zintegrowana z przydzielaniem dostępnego lekarza? [must]
5. Czy jest możliwe ręczne wprowadzanie zmian i nie powoduje ono błędów? [must]
6. Czy informacja o przydzielonym terminie jest wysyłana do pacjenta? [must]

Niefunkcjonalne:

1. Czy system decydujący o przydzieleniu określonego terminu pacjentowi jest intuicyjny w obsłudze? [should]
2. Czy informacja o przydzielonym terminie trafia do pacjenta niezwłocznie (w ciągu minuty)? [should]
2. Czy informacja o godzinie szczepienia w powiadomieniu do klienta jest znacząco wyróżniona? [should]

1.4 Przypadki użycia



Schemat 1: Diagram przypadków użycia

Diagram przypadków użycia przedstawia główne funkcjonalności projektowanego systemu, jednocześnie odzwierciedlając przynależność danej funkcjonalności do konkretnego modułu. Aplikacja składa się z 3 głównych modułów opisanych w rozdziale 1.2.

Aktorami w diagramie są

- **Pacjent**, który ma bezpośredni dostęp do Modułu rejestracji.
- **Lekarz** połączony z modułem punktu szczepień.
- **Administrator** - osoba sprawująca opiekę nad całym systemem.
- **Algorytm** odpowiedzialny za optymalne przypisywanie pacjentów do szczepień oraz lekarzy.

Każdy aktor będący realnym klientem systemu (tzn. Pacjent, Lekarz i Administrator) łączy się z nim poprzez logowanie lub dodatkowo dla pacjenta poprzez krok Rejestracji na szczepienie.

Poniżej zostaną opisane kluczowe ścieżki użycia funkcjonalności dla konkretnych agentów:

- Pacjent

- **Rejestracja na szczepienie:** użytkownik chcący zapisać się na szczepienie musi najpierw się zarejestrować i uzupełnić swoje dane.
- **Ustalenie terminu wizyty:** do ustalenia terminu wizyty potrzebna jest autoryzacja pacjenta.
- **Pobranie certyfikatu:** każdy zautoryzowany i zaszczepiony pacjent ma możliwość pobrania certyfikatu szczepienia.

- Lekarz

- **Wyświetlanie listy zaplanowanych wizyt:** zalogowany lekarz może pobrać listę wizyt przypisanych do niego w konkretnym dniu.
- **Kwalifikacja na szczepienie:** każdy pacjent przed przyjęciem szczepionki musi przejść przez kwalifikację. Jeśli lekarz stwierdzi, że pacjent jest niezdolny do przyjęcia szczepionki, to może go w następnym kroku zapisać na kolejną wizytę. W przypadku pozytywnej kwalifikacji, lekarz aplikuje szczepionkę i zapisuje status szczepienia w systemie. W przypadku, gdy szczepionka wymaga przyjęcia kilku dawek, lekarz jest też uprawniony do określenia terminu przyjęcia przez pacjenta kolejnej dawki.

- Administrator

- **Generacja raportów o szczepieniach:** zalogowany administrator może wygenerować raporty szczepień w celach analitycznych.
- **Zarządzanie danymi:** administrator ma możliwość modyfikacji danych zarówno pacjentów jak i lekarzy. Może także tworzyć konta dla nowych lekarzy.
- **Zgłaszanie błędów:** w przypadku wystąpienia błędu w systemie, administrator ma możliwość zgłoszenia go bezpośrednio do deweloperów.

- Algorytm

- **Brak logowania:** w porównaniu z innymi aktorami algorytm nie ma kroku logowania, ponieważ algorytm jest programem, który stanowi integralną część systemu, więc nie musi się autoryzować.
- **Generacja grafików lekarzy:** algorytm zajmuje się optymalizacją grafików

lekarzy. Ten przypadek użycia jest też rozdzielony na dwa mniejsze, które są w nim zawarte: Przydzielenie pacjentowi godziny szczepienia oraz Przydzielenie pacjentowi lekarza.

1.5 Diagram klas

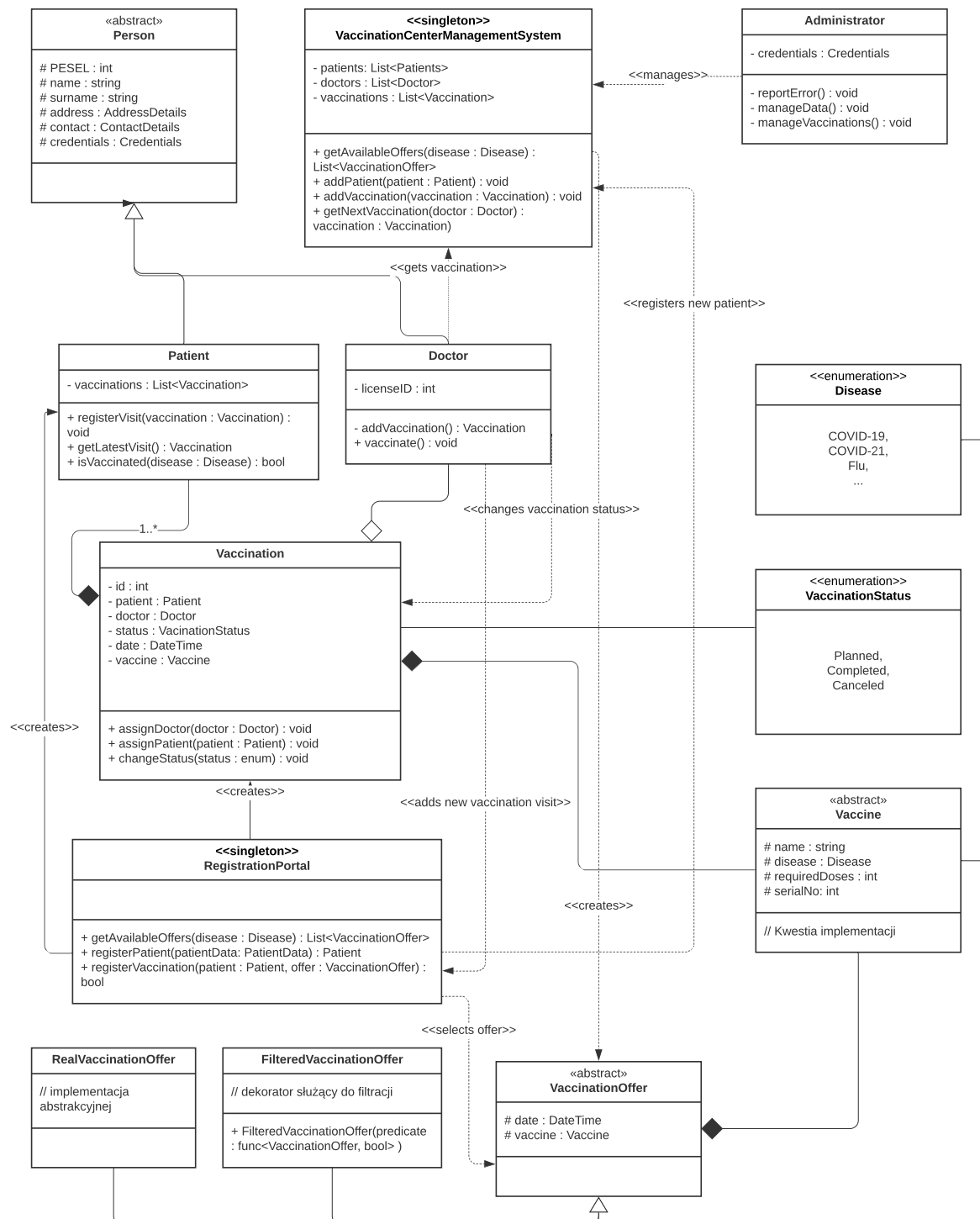
Na diagramie klas przedstawione są kluczowe elementy systemu szczepień oraz najważniejsze powiązania między nimi:

- **RegistrationPortal**, za pomocą którego pacjenci rejestrują się i spośród przedstawionych im **VaccinationOffer** wybierają dogodną wizytę.
- **VaccinationCenterManagementSystem** pełniący rolę centralnej bazy danych.
- **Vaccination**, czyli poszczególne wizyty pacjentów - zarówno te zaplanowane, zakończone jak i odwołane - który zawiera wszystkie kluczowe informacje dotyczące danej wizyty, tj. odpowiedniego pacjenta, lekarza, wybraną szczepionkę, datę etc.
- **Patient**, agregujący dane pacjenta.
- **Doctor**, który przeprowadza faktyczne szczepienie.
- **Administrator** sprawujący opiekę nad systemem.

Oprócz nich na diagramie znajdują się też klasy służące do przechowywania odpowiednich danych:

- **Person**, w której znajdują się dane osobowe i z której dziedziczy **Patient** oraz **Doctor**.
- **Vaccine** opisująca szczepionkę na daną chorobę.
- **VaccinationOffer** będąca szkieletową wersją **Vaccination** służącą do przedstawienia pacjentowi możliwych terminów szczepień.

Algorytm odpowiedzialny za optymalne przydzielanie pacjentom godzin szczepień oraz konkretnych lekarzy jest w dużej mierze kwestią implementacji i powinien w szerokim zakresie współpracować z obiektem **VaccinationCenterManagementSystem**, zatem na diagramie nie zostały umieszczone szczegóły dotyczące jego projektu.



Schemat 2: Diagram klas

1.6 Kluczowe obiekty biznesowe

System szczepień opiera się na trzech kluczowych obiektach biznesowych: pacjencie, szczepieniu i ofercie szczepień reprezentowanymi odpowiednio przez klasy **Patient**, **Vaccination**, **Vaccination Offer**.

Obiekt **Patient** zawiera dane pacjenta, który korzysta z systemu szczepień, jak również całą jego historię szczepień zapisaną w postaci listy obiektów **Vaccination**. Obiekt ten zawiera również dane logowania umożliwiające skorzystanie z modułu rejestracji.

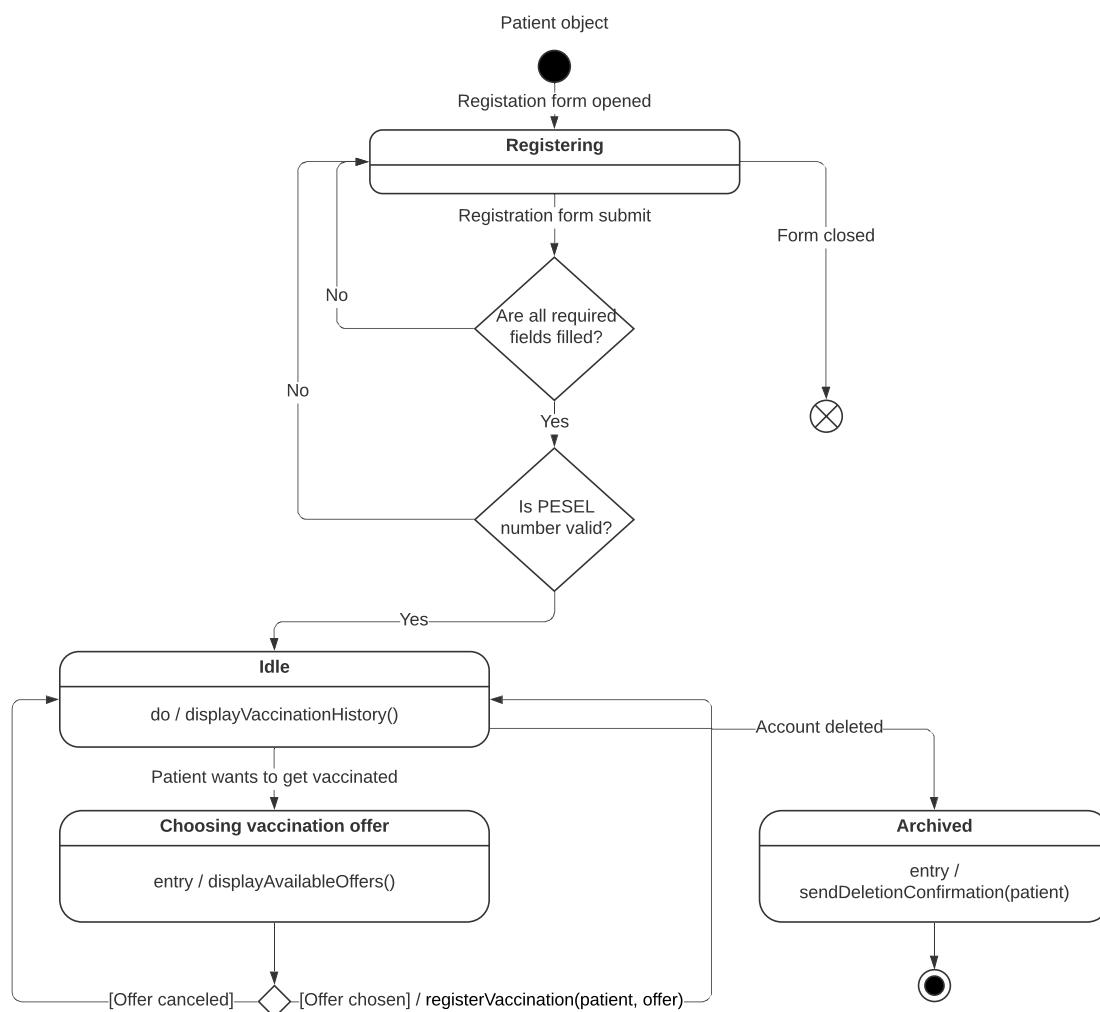
Obiekt **Vaccination** jest podstawowym obiektem służącym do zapisu informacji o pojedynczej wizycie. W trakcie rejestracji i oczekiwania na wizytę służy jako informacja o nadchodzącym szczepieniu, zaś po wykonaniu zabiegu służy jako rekord potwierdzający fakt zaszczepienia.

Obiekt **VaccinationOffer** jest uproszczoną wersją obiektu **Vaccination**, która jest wykorzystywana przez system do udostępnienia rejestrującemu się pacjentowi listy możliwych terminów szczepień.

1.7 Diagram stanów

Diagramy stanów przedstawiają cykl życia kluczowych obiektów biznesowych wykorzystywanych przez system szczepień oraz przejścia powodujące zmiany ich stanów.

1.7.1 Pacjent



Schemat 3: Diagram stanów obiektu Patient

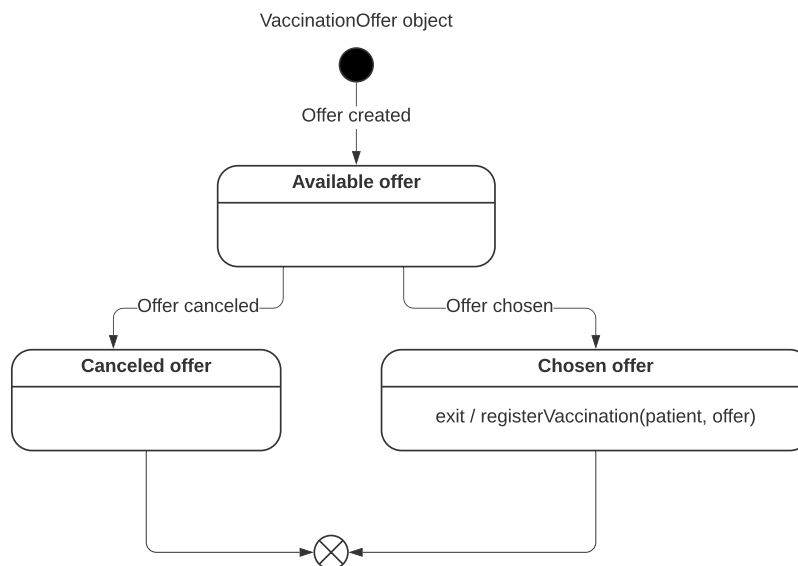
Obiekt **Patient** jest tworzony po otwarciu przez pacjenta formularza rejestracyjnego. Początkowym stanem tego obiektu jest stan *Registering*. Po przesłaniu formularza wprowadzane dane są weryfikowane, sprawdzane jest m.in. czy wszystkie pola zostały wypełnione i czy wprowadzony numer PESEL jest poprawny. W przypadku gdy weryfikacja na dowolnym z etapów się nie powiedzie, obiekt powraca do stanu *Registering*, w którym pacjent ma możliwość edycji i poprawy danych wprowadzonych w formularzu. W razie rezygnacji z rejestracji (tj. zamknięcia przez pacjenta formularza) obiekt jest niszczone.

W przypadku gdy formularz zostanie przesłany i weryfikacja zakończy się pomyślnie, obiekt przechodzi w stan *Idle*, w którym pacjent ma możliwość sprawdzenia swojej historii szczepień oraz rozpoczęcia procesu zapisu na szczepienie.

Rozpoczęcie procesu rejestracji na szczepienie powoduje przejście obiektu ze stanu *Idle* na *Choosing vaccination offer*. Pacjent ma możliwość wybrania jednej z przedstawionych przez system ofert szczepień. Jeżeli pacjent wybierze ofertę, następuje rejestracja na szczepienie, po czym obiekt wraca do stanu *Idle*. W przypadku rezygnacji z rejestracji na szczepienie, obiekt powraca do stanu *Idle* bez wykonywania żadnych akcji.

Jeżeli pacjent zażąda usunięcia swojego konta, obiekt przechodzi ze stanu *Idle* na stan *Archived* (stan finalny), zaś pacjent otrzymuje potwierdzenie wykonania akcji. W tym stanie konto pacjenta zostaje zarchiwizowane i nie jest już dłużej aktywne.

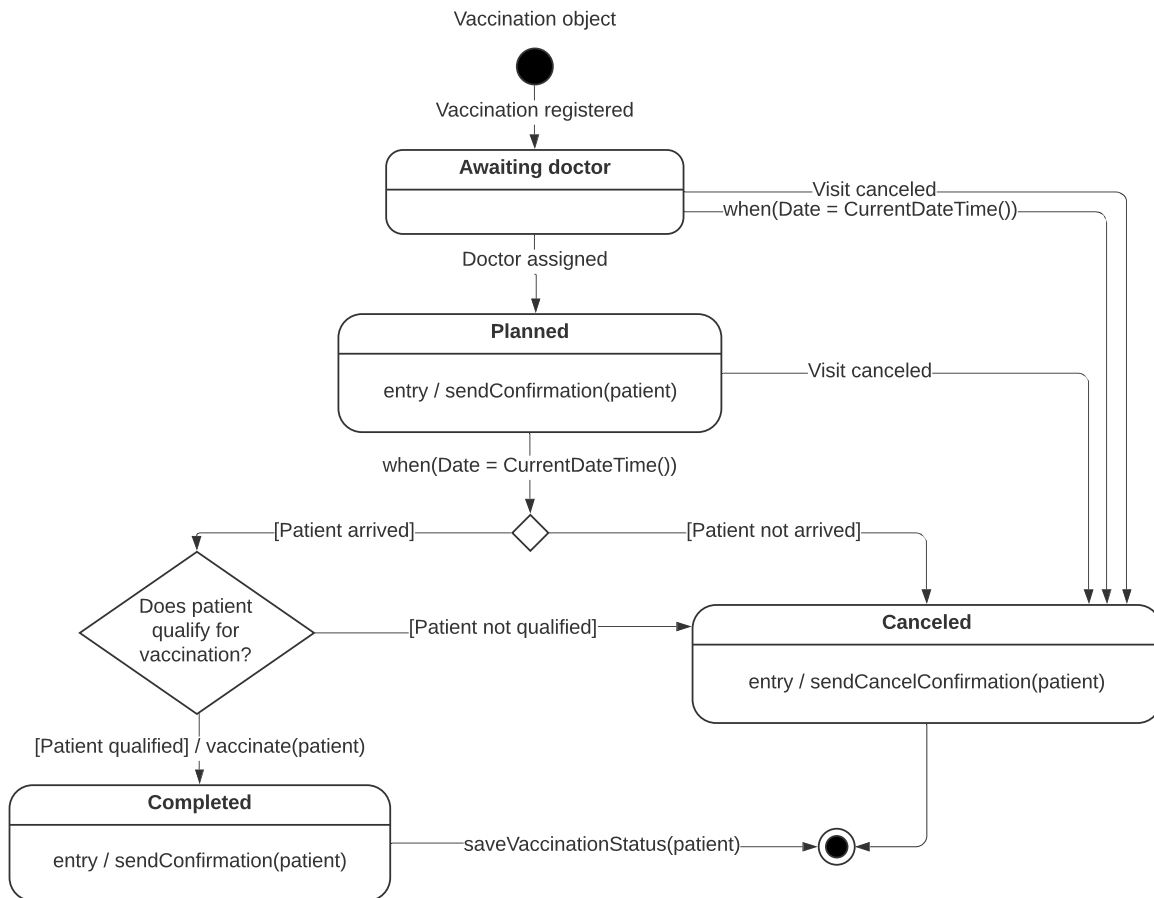
1.7.2 Oferta szczepienia



Schemat 4: Diagram stanów obiektu VaccinationOffer

Obiekt *Vaccinationoffer* jest tworzony po rozpoczęciu przez pacjenta procesu rejestracji na szczepienie i zawiera jedną z ofert szczepienia wygenerowaną przez system. Początkowym stanem tego obiektu jest stan *Available offer*. Jeżeli oferta zostanie wybrana przez pacjenta, obiekt przechodzi do stanu *Chosen offer*, gdzie następuje rejestracja na szczepienie. Jeżeli oferta nie zostanie wybrana, obiekt przechodzi do stanu *Canceled offer* bez wykonywania żadnych akcji. Po wykonaniu ewentualnych czynności w stanach *Chosen offer* i *Canceled offer* obiekt jest niszczone.

1.7.3 Szczepienie



Schemat 5: Diagram stanów obiektu Vaccination

Obiekt *Vaccination* jest tworzony po rejestracji przez pacjenta na wybrany termin szczepienia. Początkowym stanem tego obiektu jest stan *Awaiting doctor*, w którym ustalony jest już termin i rodzaj szczepionki, ale do wizyty nie jest jeszcze przypisany przez system żaden lekarz. W momencie gdy nadejdzie czas szczepienia, a obiekt będzie znajdował się w stanie *Awaiting doctor*, wizyta zostanie anulowana, zaś obiekt przejdzie w stan *Canceled*.

W momencie gdy system przypisze lekarza do wizyty, obiekt przechodzi do stanu *Planned*, w którym wysyłane jest do pacjenta potwierdzenie wizyty. Gdy nadejdzie termin wizyty weryfikowana jest obecność pacjenta w punkcie szczepień. Jeżeli pacjent jest obecny, następuje jego kwalifikacja do szczepienia. Jeżeli przejdzie on kwalifikację, to zostaje zaszczepiony, zaś obiekt przechodzi w stan *Completed*. Jeżeli pacjent nie pojawi się lub nie przejdzie kwalifikacji, to obiekt przechodzi w stan *Canceled*, a zabieg nie jest wykonywany.

Gdy obiekt znajduje się w stanach *Awaiting doctor* lub *Planned*, pacjent ma możliwość anulowania wizyty. Obiekt przechodzi wtedy w stan *Canceled*. Po wykonaniu czynności w stanach *Completed* i *Canceled* obiekt przechodzi w stan finalny.

Po nadejściu czasu wizyty moduł punktu szczepień dokonuje sprawdzenia, czy pacjent stawiał się na wizytę. W przypadku, gdy pacjent nie pojawił się na czas wizyty zostaje anulowana. Jeżeli pacjent stawiał się w punkcie szczepień, lekarz przystępuje do kwalifikacji na szczepienie. W przypadku, gdy pacjent nie zakwalifikuje się do zabiegu, lekarz wyznacza mu nowy termin wizyty zgodnie z procedurą rejestracji na wizytę. Jeżeli pacjent zostanie zakwalifikowany pozytywnie, lekarz przystępuje do wykonania zabiegu, po czym w zależności od potrzeb może również wyznaczyć termin następnej wizyty (przykładowo gdy wymagane jest podanie kolejnej dawki szczepionki). Następnie w module administracyjnym zapisywany jest aktualny status szczepienia i wysyłane jest potwierdzenie zakończonej wizyty.

W przypadku anulowania wizyty, moduł rejestracji wysyła sygnał *Visit canceled by patient*, który jest odbierany przez moduł administracyjny, który dokonuje anulowania wizyty i wysyła pacjentowi potwierdzenie anulowania.

1.8.2 Procedura rejestracji na wizytę

Podczas procedury rejestracji na wizytę pacjent wybiera z pomocą modułu rejestracji rodzaj choroby, na którą chciałby się zaszczepić, po czym otrzymuje listę dostępnych ofert, spośród których może wybrać tę, która najbardziej mu odpowiada. Jeżeli pacjent wybierze ofertę, moduł rejestracji wysyła sygnał *New visit registered*, po czym procedura się kończy. Jeżeli pacjent zrezygnuje z wyboru oferty, będzie on mógł wrócić do listy dostępnych ofert i wybrać inną.

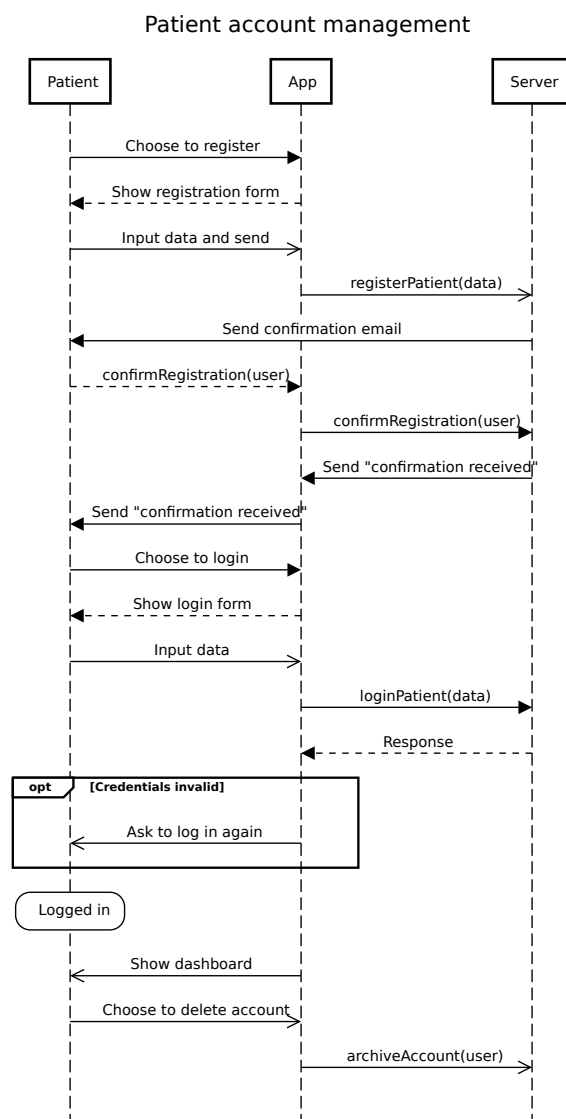
1.8.3 Procedura wyszukiwania i przydzielania dostępnego lekarza

Procedura wyszukiwania i przydzielania dostępnego lekarza wyszukuje z pomocą modułu administracyjnego lekarza, który mógłby wykonać szczepienie w wybranym przez pacjenta terminie. Moduł administracyjny oczekuje na dostępność lekarza, po czym przydziela go do rozważanej wizyty, po czym procedura kończy działanie. Moduł administracyjny może przerwać tę procedurę po upływie ustalonego czasu.

1.9 Diagram sekwencji

Wszystkie diagramy sekwencji oparte są o podział Użytkownik - Aplikacja (frontend) - Serwer (backend / baza danych).

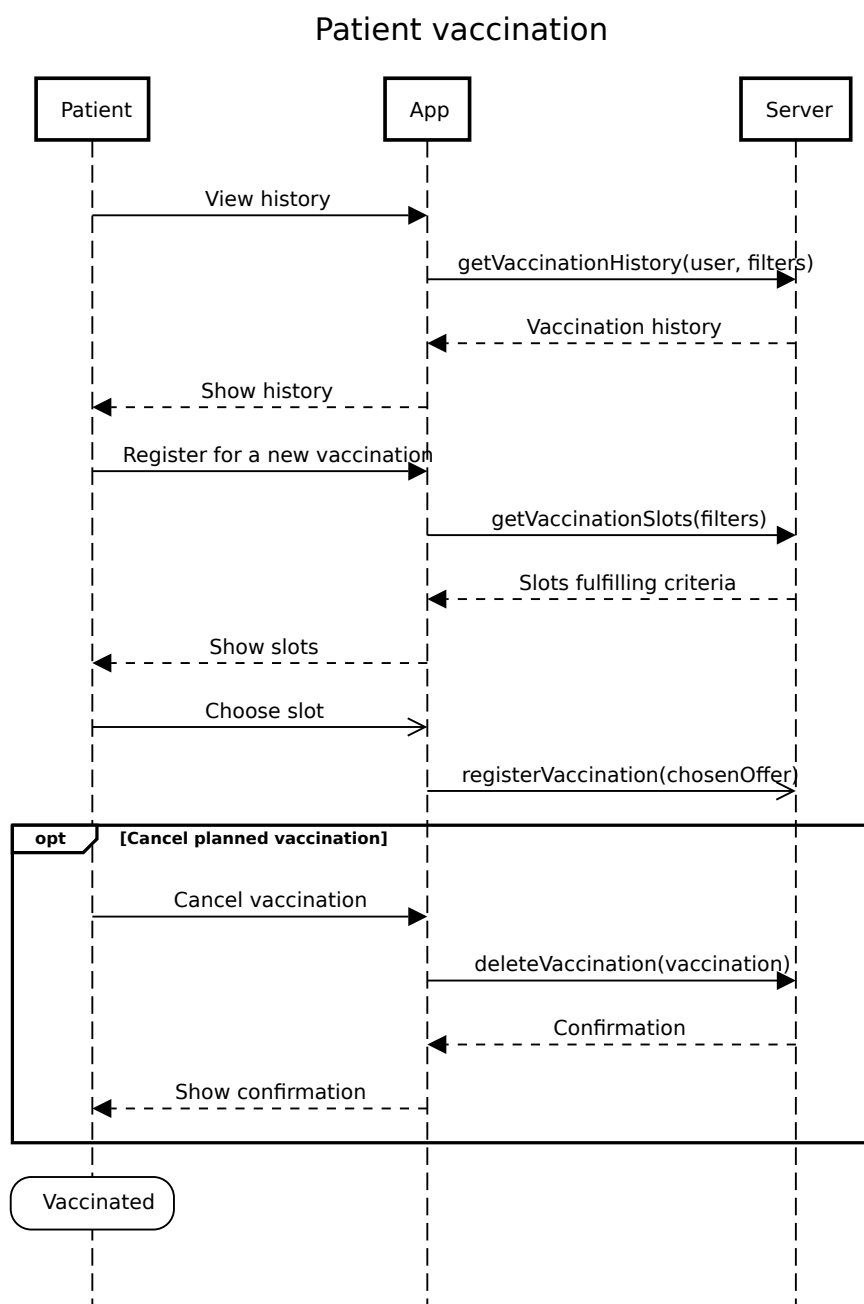
1.9.1 Zarządzanie kontem pacjenta



Schemat 7: Diagram sekwencji zarządzania kontem pacjenta

Diagram zarządzania kontem pacjenta pokazuje, jak wygląda komunikacja między aplikacją a serwerem w odpowiedzi na żądania użytkownika, który przechodzi przez kolejne etapy tworzenia swojego konta. Serwer odpowiada na zapytania dotyczące zarejestrowania nowej osoby w bazie danych, wysyła odpowiedni mail aktywacyjny, umożliwia faktyczne zalogowanie w systemie i zarchiwizowanie konta. Reszta funkcjonalności, np. ewentualnie sprawdzenie poprawności danych, pokazywanie odpowiednich formularzy, etc. leży po stronie aplikacji, z którą użytkownik wchodzi w interakcję.

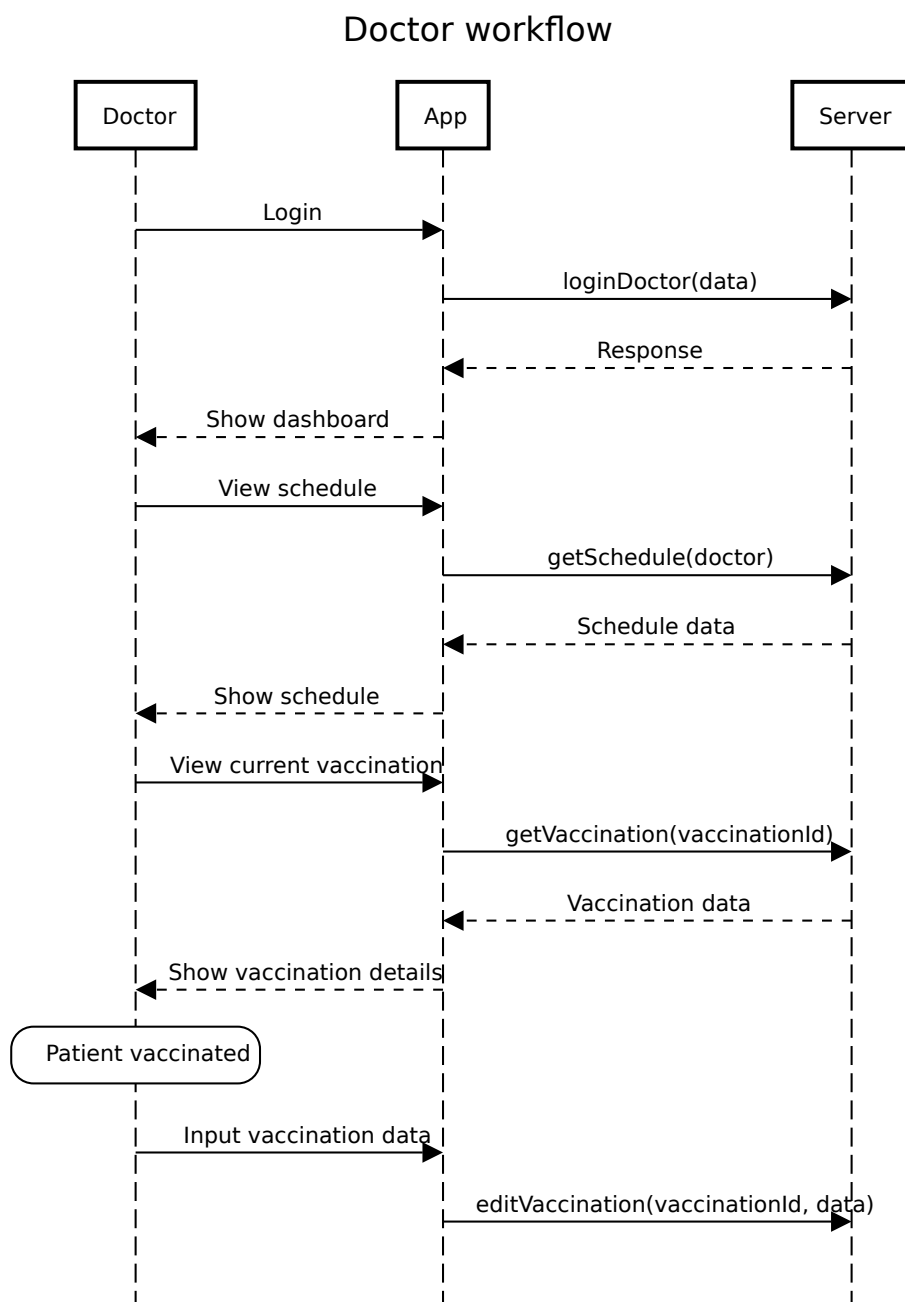
1.9.2 Szczepienie



Schemat 8: Diagram sekwencji procesu szczepienia

Na tym diagramie widać, jak przebiega proces rejestracji na nowe szczepienie. Po uprzednim zalogowaniu w systemie użytkownik może sprawdzić historię swoich poprzednich wizyt, również z możliwością filtracji wg. odpowiednich kryteriów, a następnie przejść do formularza rejestracyjnego. System odsyła wolne terminy dla szczepień, które interesują użytkownika, spośród których ten może dokonać najlepszego dlań wyboru. Możliwe jest również anulowanie zaplanowanej wizyty - w takim przypadku na odpowiednie żądanie system odsyła potwierdzenie wykonania zlecenia.

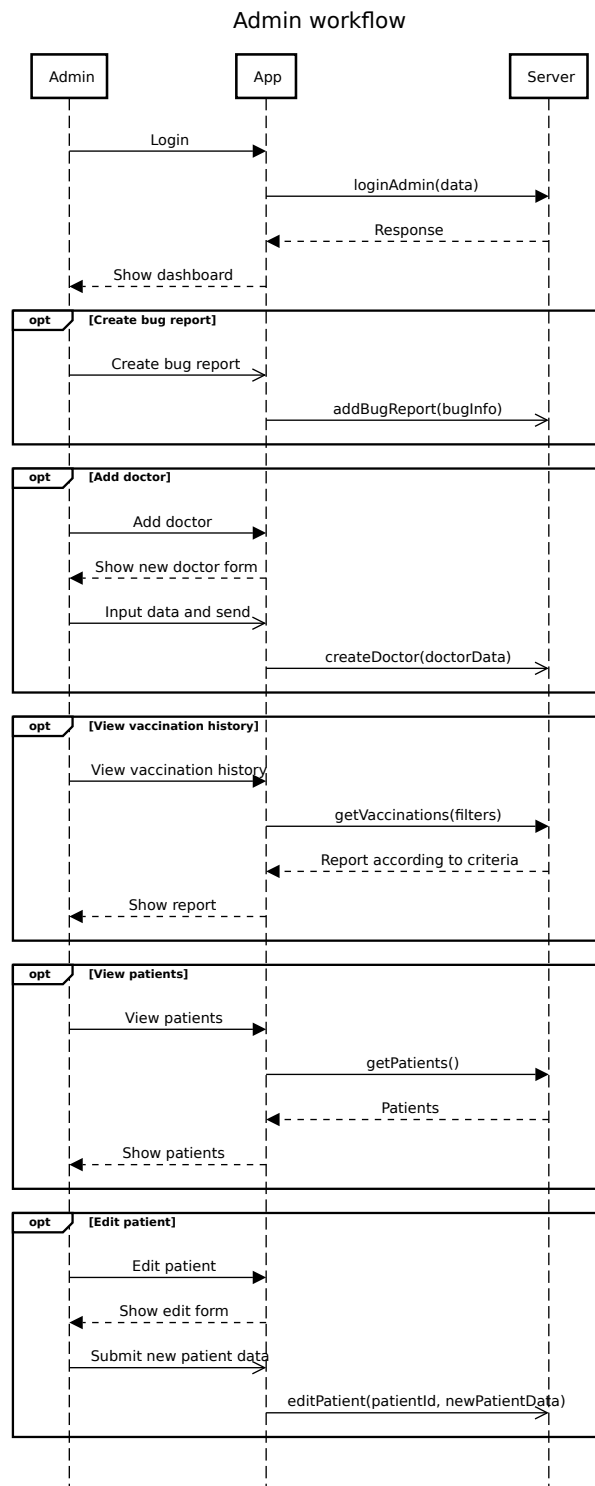
1.9.3 Praca lekarza



Schemat 9: Diagram sekwencji pracy lekarza

Diagram opisuje, jak wygląda wycinek pracy lekarza od momentu zalogowania w systemie do momentu zapisania nowego szczepienia - po sprawdzeniu swojego grafiku na dany dzień, lekarzowi po kolei wyświetlają się dane poszczególnych zaplanowanych wizyt. Po wykonaniu szczepienia lekarz wpisuje odpowiednie dane do udostępnionego mu formularza, które są następnie zapisywane do systemu.

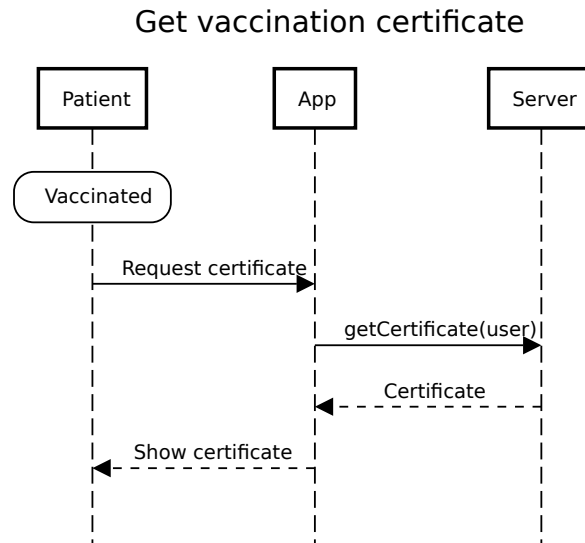
1.9.4 Praca administratora



Schemat 10: Diagram sekwencji pracy administratora

Diagram z administratorem pokazuje, jakich akcji może on dokonać w systemie oraz jak przebiegają związane z nimi procesy. Administrator ma dostęp do CRUD'a zarówno dla pacjentów, jak i doktorów, może zapisywać w systemie sprawozdania z błędów do obsługi przez dział IT oraz generować rozmaite raporty na podstawie danych ze szczepień.

1.9.5 Pobieranie certyfikatu szczepienia



Schemat 11: Diagram sekwencji pobierania certyfikatu szczepienia

Diagram obrazuje prostą interakcję, która zachodzi w przypadku, kiedy użytkownik systemu pragnie pobrać certyfikat szczepień, które odbył.

1.10 Dokumentacja API

Dokumentacja API wykorzystywanego w systemie została opisana w załącznikach do niniejszej specyfikacji:

- admin.html,
- doctor.html,
- patient.html.

Do każdego z powyższych plików zostały również dołączone źródła w formacie OpenAPI.

2 Algorytmy biznesowe

W rozważanym systemie szczepień szczególne znaczenie ma algorytm odpowiadający za przydzielenie lekarza do zaplanowanej wizyty.

2.1 Założenia algorytmu

Z perspektywy trwającego procesu szczepień najwyższym priorytetem jest zapewnienie pacjentom możliwości zaszczepienia się w dogodnym, możliwie najszybszym terminie, dlatego wykorzystany algorytm na pierwszym miejscu powinien skupić się na umożliwieniu

rejestrację na wizytę jeśli tylko możliwe jest znalezienie lekarza na ten termin. Dopiero na kolejnym miejscu znajdować powinna się kwestia optymalnego wykorzystania czasu lekarzy i optymalnego dobrania wielkości personelu pracującego w danym dniu.

Z punktu widzenia systemu szczepień dopuszczalne jest zastosowanie algorytmu o naturze hybrydowej - tj. stosującego nieoptymalną strategię w celu przypisania lekarzy do poszczególnych terminów na bieżąco, w miarę kolejnych rejestracji (przykładowo przy pomocy podejścia zachłannego), a następnie wykonaniu co określony czas reorganizacji grafiku przy zastosowaniu optymalniejszej strategii. Przy zastosowaniu takiego podejścia należy jednak zagwarantować brak przerw w działaniu systemu oraz przezroczystość procesu z perspektywy pacjenta, który nie zna lekarza przypisanego do swojej wizyty. W szczególności nie może dojść do sytuacji, w której przy optymalizacji grafiku pacjent straci lekarza przypisanego do swojego terminu, a w konsekwencji jego wizyta zostanie anulowana.

2.2 Referencyjna implementacja

Krok 1. Algorytm otrzymuje prośbę o przydzielenie lekarza do wizyty, która odbędzie się w podanym terminie.

Krok 2. Algorytm pobiera listę lekarzy pracujących w punkcie szczepień, którzy są dostępni w podanym terminie.

Krok 3. Jeżeli w wyniku działania kroku 2. znaleziono więcej niż jednego lekarza, algorytm wybiera lekarza, dla którego pozostałe wizyty są najbliższe podanego terminu - co jest rozumiane jako różnica czasu zakończenia ostatniej wizyty przed podanym terminem i czasu rozpoczęcia aktualnie rozważanej wizyty lub różnica czasu zakończenia rozważanej wizyty i czasu rozpoczęcia pierwszej wizyty po podanym terminie.

Krok 4. Jeżeli w wyniku działania kroku 3. znaleziono więcej niż jednego lekarza, algorytm wybiera lekarza, który w danym dniu ma przypisaną większą liczbę wizyt.

Krok 5. Jeżeli w wyniku działania kroku 4. znaleziono więcej niż jednego lekarza, algorytm wybiera pierwszego lekarza z przedstawionej listy kandydatów.

Krok 6. Jeżeli w wyniku działania kroku 2. nie odnaleziono żadnego lekarza, algorytm rozpoczyna oczekiwanie na pojawienie się dostępnych lekarzy, po czym wraca do rozważań z kroku 2. Oczekiwanie może zostać przerwane przez czynnik zewnętrzny (moduł administracyjny) po upływie zdefiniowanego maksymalnego czasu oczekiwania.

Krok 7. Jeżeli w wyniku działania powyższych procedur został wyznaczony lekarz, jest on zwracany przez algorytm.

3 Wymagania techniczne

3.1 Wymagania systemu

Ze względu na zakładaną powszechną dostępność systemu szkielet komponenty wykorzystywane przez użytkowników końcowych (*frontend*) powinny zostać zaimplementowane w formie aplikacji internetowych, które będą poprawnie działały na aktualnych wersjach najpopularniejszych przeglądarek internetowych (Google Chrome, Edge, Firefox, Opera) oraz przeglądarek internetowych na urządzeniach mobilnych (Google Chrome, Safari).

Ze względów praktycznych (dostępność systemów i komputerów do testowania i uruchamiania aplikacji) silnik systemu szkielet (*backend*) powinien zostać zaimplementowany w formie programu działającego na systemie Windows 10 (lub nowszym) oraz Windows Server 2019 (lub nowszym). Obsługa innych systemów operacyjnych (wynikająca przykładowo z zastosowania technologii multiplatformowej) jest opcjonalna.

3.2 Preferowane technologie

Przy implementacji systemu szkielet proponowane jest wykorzystanie jednego z popularnych frameworków frontendowych takich jak React, Vue lub Angular, zaś w przypadku backendu jednego z frameworków takich jak ASP.NET (C#), Spring (Java), Laravel (PHP) i podobne.

Wśród dostępnych technologii preferowane (ale nie wymagane) jest wykorzystanie frameworka Vue dla frontendu, zaś ASP.NET do backendu. Za wyborem tych technologii stoi kilka istotnych czynników, takich jak:

- dobra znajomość języków JavaScript i C# wśród grupy programistów (studentów MiNI) zajmujących się implementacją systemu,
- popularność frameworków Vue oraz ASP.NET oraz ich dobra dokumentacja,
- duża liczba zewnętrznych bibliotek do wyżej wymienionych frameworków,
- dobra integracja języka C# z innymi preferowanymi narzędziami firmy Microsoft (chmura Azure, baza danych MS-SQL),
- spójność środowiska .NET i dostępne narzędzia ułatwiające szybkie stworzenie rozważanego systemu, wsparcie dla multiplatformowości,
- wysoka jakość i stabilność produktów firmy Microsoft w porównaniu do konkurencyjnych rozwiązań,

4 Testowanie i obsługa błędów

4.1 Testowanie

Do testów aplikacji proponujemy wykorzystanie następujących rozwiązań:

- testowanie UI/UX - cypress,
- testy jednostkowe na backendzie C#, ASP.NET- natywne mechanizmy środowiska .NET,
- testy jednostkowe na frontendzie - jest,
- testy wydajnościowe - skrypt generujący odpowiednio dużą liczbę zapytań dla REST API.

4.2 Obsługa błędów

4.2.1 Moduł rejestracji

Proces rejestracji na szczepienie powinien zostać zaimplementowany w postaci transakcji. W przypadku błędu systemu podczas całego procesu rejestracji (przykładowo wynikającego z błędu zapisu w bazie danych), zapis powinien zostać wycofany, zaś pacjent powinien otrzymać komunikat o błędzie systemu uniemożliwiającym ukończenie rejestracji na wybrany termin z prośbą o spróbowanie ponownie.

W przypadku utraty komunikacji pomiędzy modułem rejestracji a modułem administracyjnym, moduł rejestracji powinien wyświetlać komunikat o awarii i nie pozwalać na kolejne rejestracje, aż do momentu odzyskania łączności.

4.2.2 Moduł punktu szczepień

Proces zapisywania procesu szczepień powinien umożliwiać pracę offline (tj. wprowadzenie danych o zrealizowanym szczepieniu z opóźnieniem, po ustaniu ewentualnych problemów z systemem), dlatego w przypadku błędu systemu uniemożliwiającego zapis informacji o wykonanym zabiegu, dane powinny zostać zapisane w buforze wykorzystanym do pracy offline i zapisane ponownie po ustaniu awarii.

W przypadku utraty komunikacji pomiędzy modułem punktu szczepień a modułem administracyjnym, moduł punktu szczepień powinien wyświetlać komunikat o awarii i umożliwić pracę w trybie offline.

4.2.3 Moduł administracyjny

W przypadku utraty komunikacji pomiędzy modułem administracyjnym a modułem rejestracji na etapie wyszukiwania lekarza, system powinien kontynuować przydzielanie lekarza i wysłać potwierdzenie wizyty po znalezieniu. Każdą inną operację należy traktować jako przerwana i anulowaną.

5 Załączniki

Dokumentacja API wykorzystanego w systemie:

- admin.html,
- doctor.html,
- patient.html.

Do każdego z powyższych plików zostały również dołączone źródła w formacie OpenAPI:

- admin.json,
- doctor.json,
- patient.json.