

Projekt #1

Przychodnia lekarska

Kamil Opalach, Michał Remiszewski, Maciej Momot

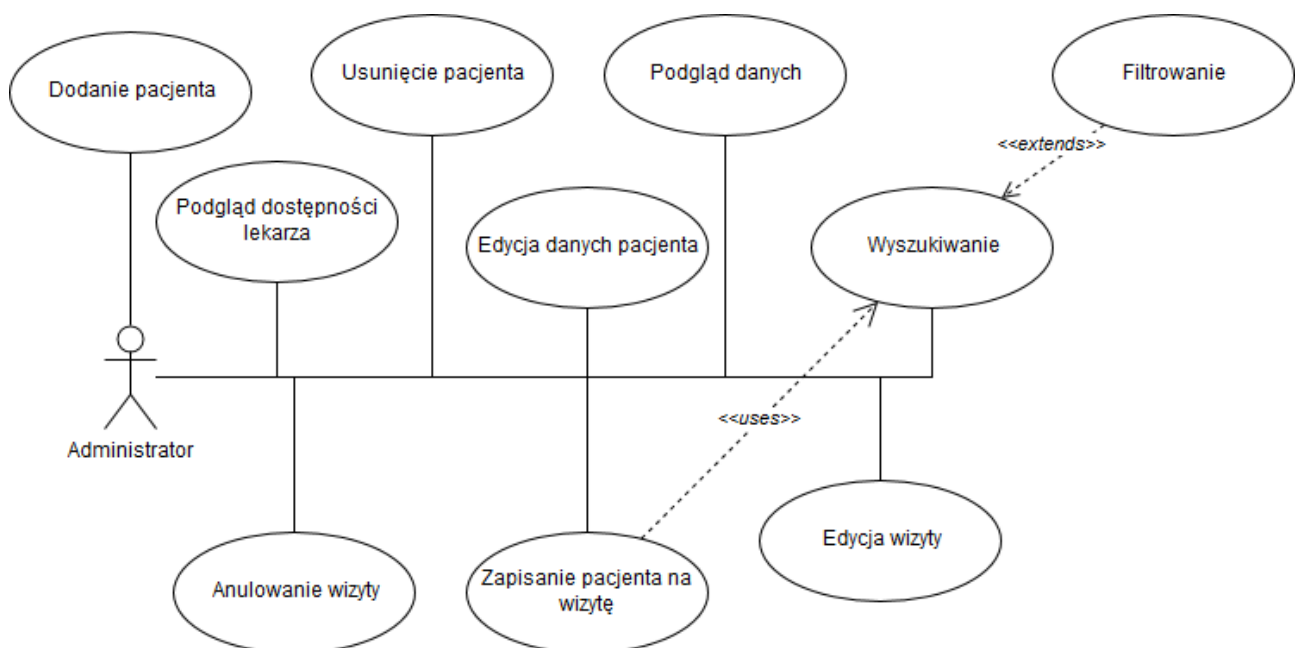
Wersja 1b

26.10.2017r.

1.1 Opis biznesowy

Niniejszy system służy do gromadzenia i edycji danych dotyczących pacjentów, lekarzy i placówek medycznych. Możliwość edycji danych jest dostępna jedynie dla administratora. Zarządza on bazą danych i ustala harmonogram wizyt pacjentów w poszczególnych placówkach, przypisuje ich do wybranych medyków.

1.2 Wymagania funkcjonalne



Rysunek 1. Przypadki użycia przez administratora

Aktor	Nazwa	Opis	Odpowiedź systemu
Administrator systemu	Dodanie pacjenta.	Dodanie nowego pacjenta do bazy danych.	Potwierdzenie operacji.
Administrator systemu	Usunięcie pacjenta.	Usunięcie pacjenta z bazy danych.	Potwierdzenie operacji.
Administrator systemu	Wyszukanie pacjenta.	Wyszukanie pacjenta na podstawie wprowadzonych danych.	Wypisanie szczegółowych danych

			pacjenta.
Administrator systemu	Podgląd danych.	Pobranie listy pacjentów z możliwością dodania kryteriów filtrowania listy wynikowej.	Tabela pacjentów spełniających wybrane kryteria.
Administrator systemu	Podgląd dostępności lekarza.	Podgląd dostępności lekarza w określonym terminie.	Wyświetlenie wyników zapytania.
Administrator systemu	Przypisanie pacjenta na wizytę.	Przypisanie pacjenta na wizytę do określonego lekarza na dany termin.	Potwierdzenie operacji.
Administrator systemu	Anulowanie i edycja wizyt.	Edycja wprowadzonych do systemu wizyt.	Potwierdzenie operacji.
Administrator systemu	Edycja danych pacjenta.	Edycja wprowadzonych wcześniej danych pacjenta.	Potwierdzenie operacji.

Tabela 1. Opisy przypadków użycia dla administratora

User stories

1. Interfejs administratora

1.1. Jako administrator dodaję nowego użytkownika, aby móc zapisać go na wizytę.

Administrator może dodać nowego użytkownika do bazy danych lub w dowolnym momencie edytować dodane dane, po czym ustalonego pacjenta może przypisać do wybranej kliniki zapisując na wizytę do konkretnego lekarza dostępnego w danym terminie.

1.2. Filtrowanie danych pacjentów pod kątem daty ostatniej wizyty w celu uzyskania listy nieaktywnych użytkowników.

Administrator może wyświetlić listę pacjentów z dodanym filtrem, który wyselekcjonuje osoby spełniające podane kryteria wyszukiwania. Dane personalne może następnie wyeksportować do pliku.

1.3 Wymagania niefunkcjonalne

Obszar wymagań	Numer wymagania	Opis
Użyteczność	1	Wszystkie funkcjonalności aplikacji dostępne dla użytkownika muszą mieścić się na pojedynczym ekranie przy rozdzielczości 1920x1080 i czcionce nie mniejszej niż 12pt.
Niezawodność	2	Aplikacja ma być dostępna w czasie pracy administratora pomiędzy godziną 7:00 a 14:00 z pojedynczą przerwą obiadową nie dłuższą niż godzinę dziennie.
Wydajność	3	Aplikacja powinna dodawać nowe obiekty do systemu w czasie nie dłuższym niż sekunda.
	4	W czasie kontaktu z bazą danych np. podczas listowania aplikacja nie zakłóca pracy Systemu.
Utrzymanie	5	Aplikacja powinna zachować wsteczną zgodność z interfejsem dostępu do poszczególnych obiektów.

Tabela 2. Lista wymagań niefunkcjonalnych

1.4 Harmonogram projektu

Faza pierwsza:

- 29.10 – stworzenie bazy danych, do której można będzie dodać, usunąć, edytować użytkownika, a także wyświetlić jego dane,

Faza druga:

- 3.11 – wyszukiwanie użytkowników z użyciem filtrów, wyszukiwanie wolnych terminów wizyt i możliwość przypisania do nich pacjentów,
- 10.11 – pełna funkcjonalność oraz testy jednostkowe wraz z dokumentacją.

1.5 Architektura rozwiązania

Podział danych ze względu na ich zmienność:

- Dane niezmiennające się – dane statyczne (static data),
- Dane wolnozmiennne (slow changed dynamic data),
- Dane szybko zmieniające się (dynamiczne data),
- Dane czasu rzeczywistego (real time data).

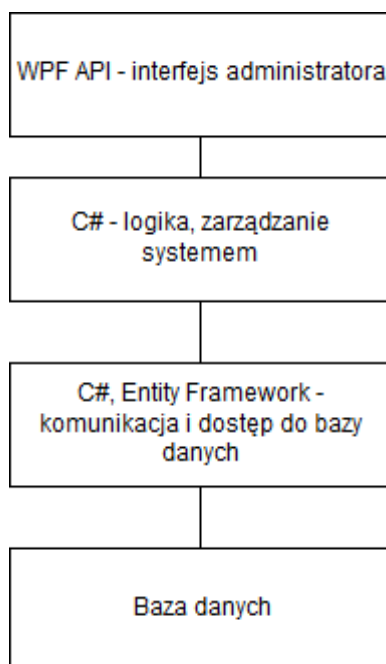
Podział danych ze względu na typ źródła danych:

- Dane nietabelaryczne,
- Dane tabelaryczne,
- Bazy danych.

Budowana platforma powinna zapewniać dla administratora następujące funkcje:

- Dostęp do katalogu danych z opisem poszczególnych API.
- Jeden punkt dostępu do API.
- Możliwie spójna struktura API do różnych zbiorów danych i funkcji (serializacja i format przesyłanych danych, sposób autoryzacji, kodowanie polskich znaków, układ współrzędnych dla danych geograficznych itp.).
- Możliwość dokonywania konwersji zapytań i odpowiedzi pomiędzy platformą i pierwotnym źródłem danych.

Aplikacja będzie składała się z czterech warstw. Komunikacja między nimi będzie odbywała się z wykorzystaniem fasady.



Rysunek 2. Architektura systemu

Do budowy aplikacji założono wykorzystanie następujących komponentów:

- Frontend – Aplikacja okienkowa w technologii Windows Presentation Foundation,
- Warstwa logiki – C#,
- Komunikacja z bazą danych – C#, Entity Framework.