Architektura aplikacji

April 27, 2023

1 Serwer

1.1 Technologie dotyczące części aplikacyjnej

Serwer aplikacji zostanie napisany w języku Rust z użyciem technologii Axum. Będzie udostępniał endpointy, do których będą mogły być wysyłane zapytania REST API. Serwer będzie pośrednikiem pomiędzy użytkownikiem a bazą danych i będzie jedynym miejscem, z którego będą mogły być wykonywane zapytania bezpośrednio na bazie danych.

1.2 Technologie dotyczące komunikacji z bazą danych

Komunikacja zostanie zrealizowana z użyciem biblioteki SQLx. Zapytania będą konstruowane w języku SQL, a potem importowane jako argumenty odpowiednich funkcji lub makr.

1.3 Udostępnione endpointy

Dokumentacja api udostępnianego przez serwer znajduje się pod tym linkiem.

2 Interfejs użytkownika

Interfejs użytkownika zostanie stworzony z użyciem HTML5, CSS3 oraz JavaScript. Strony opisane w specyfikacji (to znaczy Strona Główna, Wybór Choroby, Lista wyników) to będą strony z odpowiednimi wartościami metody GET. Ściślej, będą zgodne z następującym GUI:

- 'GET /' - kod HTML strony głównej.
- 'GET /search?q=<fraza>&p=<strona>'- kod HTML na zapytanie przekazane w parametrze 'q' i numerze strony przekazanym w parametrze 'p'.
- 'GET /selection/<ean>' - strona z wyborem poziomu refundacji odpowiadająca produktowi o kodzie EAN '<ean>'.

- 'GET /<id>' - strona z dokładnymi informacjami o leku i jego odpowiednikach (po wyborze poziomu refundacji) odpowiadająca produktowi, któremu zostało przypisane '<id>' wewnątrz

2.1 Strona główna

Użytkownik będzie wpisywał do etykiety <input type='search'> (znajdującego się na środku strony) wyrażenie do wyszukiwania. Po kliknięciu przycisku ENTER będzie wysyłane zapytanie do serwera na adres '/api/search'. Wynikiem będzie posortowana lista leków na podstawie algorytmu opisanego w specyfikacji.

2.2 Wyniki wyszukiwania

Wyświetlona zostanie tabela wyników, maksymalnie 50 wyników na raz, w przypadku, gdy miałoby być zwróconych ich więcej, na dole strony będzie się znajdowała Nawigacja stron z wynikami. Po najechaniu kursorem na dany wiersz tabeli wyników, zostanie on podświetlony, sugerując że może zostać kliknięty przez użytkownika (zostanie użyta opcja HTML 'hover' oraz CSS w tym celu). Po kliknięciu odpowiedniej etykiety <input type='button'>, w przypadku posiadania przez lek wielu różnych wskazań refundacyjnych, użytkownik zostanie przeniesiony do okna wyboru choroby. Kliknięcie wyśle do serwera zapytanie na adres '/api/groups/<group_id>.json'.

2.2.1 Nawigacja stron

Po kliknięciu kursorem etykiety <input type='button'>, użytkownik zostanie przeniesiony na odpowiedni fragment listy wyników (część wyników będzie niewidoczna dla użytkownika, bo w HTML będzie oznaczona jako 'hidden', JS zmieni widoczność).

2.3 Wybór choroby

Po najechaniu kursorem na dany wiersz tabeli wyników, zostanie on podświetlony. Po kliknięciu (odpowieda etykiecie <input type='button'>) przeglądarka wyśle zapytanie do serwera na adres '/api/selection/<ean>.json'.

2.4 Dokładne informacje o leku

Po najechaniu kursorem na dany wiersz tabeli wyników, zostanie on podświetlony. Po kliknięciu (odpowieda etykiecie <input type='button'>) przeglądarka wyśle zapytanie do serwera na adres '/api/details/<id>.json'.

3 Baza danych

Dane przetwarzane przez projekt będą obsługiwane przez bazę danych PostgreSQL. Tabele w bazie będą przechowywane w schemacie utworzonym przez

polecenie:

```
CREATE SCHEMA odpowiedniki;
```

Tabela zawierająca skatalogowane leki:

```
CREATE TABLE odpowiedniki.leki (
    id INTEGER PRIMARY KEY,
    name VARCHAR NOT NULL,
    ean VARCHAR NOT NULL,
    refund VARCHAR NOT NULL,
    price INTEGER NOT NULL,
    amount INTEGER NOT NULL,
    amount_unit VARCHAR,
    group_id INTEGER NOT NULL
);
```

W przypadku leku, pole `"amount_unit"` będzie zawierało jednostkę w której została podana ilość sztuk w opakowaniu. Jeśli lek nie został przetworzony, pole `"amount_unit"` pozostaje puste, oraz pole `"group_id"` zawiera wartość -1.

3.1 Zapytania bazy danych

Interfejs użytkownika będzie realizowany za pomocą następujących zapytań w bazie danych:

jd