

Architektura aplikacji “Tańsza Recepta”

Paweł Czajka, Maciej Kraiński, Jola Mozyrska, Marcin Mazurek

07.04.20

Podział projektu

“Tańsza Recepta” jest aplikacją webową napisaną w Pythonowym module Dash. Składają się na nią trzy części: baza danych, serwer aplikacji oraz front-end, obsługiwany przez przeglądarkę.

Baza danych

Wykorzystywana w projekcie baza danych korzystająca z PostgreSQL przechowuje informacje o refundowanych w Polsce lekach, w pojedynczej tabeli w następującym formacie:

```
CREATE TABLE refunds (  
    ean varchar(13),  
    announcement_date date,  
    refund_level varchar,  
    description varchar NOT NULL,  
    active_ingredient varchar NOT NULL,  
    form varchar NOT NULL,  
    dose varchar NOT NULL,  
    unit_price numeric(5, 2) NOT NULL,  
    CONSTRAINT refunds_pkey  
        PRIMARY KEY (ean, announcement_date, refund_level)  
);
```

Każdy wiersz tabeli “refunds” odpowiada wystąpieniu w jednym z oświadczeń informacji o refundacji leku na pewnym poziomie odpłatności. W bazie przechowywane są jedynie informacje o lekach które należą do podzbioru leków opisanego w wymaganiach funkcjonalnych aplikacji.

Kolumny tabeli refunds zawierają zaimportowane dane z oświadczeń publikowanych przez Ministerstwo Zdrowia, ze szczególnym uwzględnieniem:

- Kolumna description zawiera połączone dane z kolumny trzeciej i czwartej oświadczenia ministerstwa. Wartość ta będzie stanowiła opis leków w front-endzie aplikacji.
- Kolumny dose oraz form zawierają dane o odpowiednio dawce i postaci leku wyekstrahowane z trzeciej kolumny oryginalnego dokumentu. Postać leku zostanie

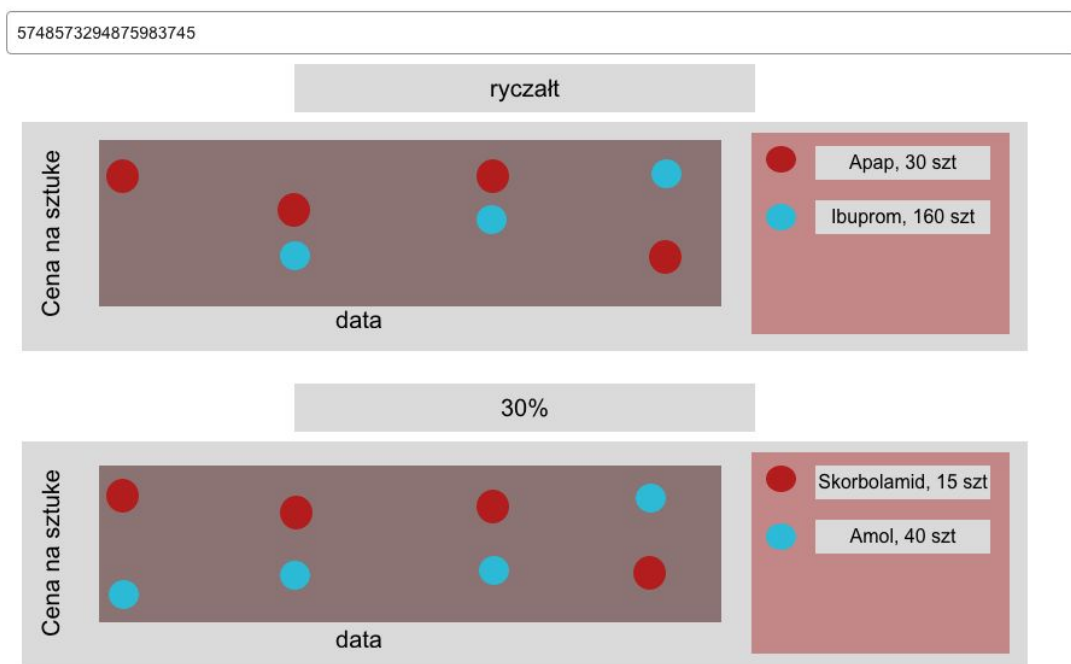
ponadto uproszczona do jednej z przyjętych form (tabletki lub czopki). Z kolei dawka (dose) podana będzie w miligramach (mg) lub jednostkach międzynarodowych (UI).

- Kolumna unit_price zawiera wysokość dopłaty świadczeniobiorcy pochodzącą z ostatniej kolumny oświadczenia ministerstwa podzieloną przez liczbę jednostek leku z czwartej kolumny tego oświadczenia.
- Pozostałe kolumny powstają poprzez nieskomplikowane zaimportowanie wartości z odpowiednich kolumn oświadczenia Ministerstwa Zdrowia.

Back-end

Jest zaimplementowany przy pomocy pakietu Dash. Aplikacja reprezentowana jest przy pomocy obiektu app wyprodukowanego przez fabrykę dash.Dash. Aplikacja składa się z jednego endpointu, który służy zarówno do przyjmowania od użytkownika kodu EAN leku, jak i do wyświetlania uzyskanych wykresów. Serwer aplikacji odpowiada za komunikowanie się z front-endową warstwą oraz z bazą danych.

Front-end



Działająca w przeglądarce warstwa front-endowa aplikacji będzie również zaimplementowana przy pomocy pakietu Dash. Wykorzystamy do tego atrybut app.layout, do którego przypiszemy obiekty reprezentujące html-owe elementy, które zaprezentowane zostaną użytkownikowi przez przeglądarkę.

Wszystkie elementy prezentowane użytkownikowi zamknięte są w obiekcie `dash_html_components.Div`. Element ten ma następujące dzieci:

- `dash_core_components.Dropdown`, o id "dropdown", który stanowi input przyjmujący ean od użytkownika. Jest on zainicjowany eanami oraz skrótowymi opisami leków występujących w bazie danych. Użytkownik w trakcie wpisywania eanu zobaczy zawężającą się listę leków, których eany mają wpisany prefix.
- `dash_html_components.Div` o id "content", którego dziećmi są wykresy odpowiadające poziomom odpłatności znalezionych alternatyw leku. Wykresy te wyświetlane będą jeden pod drugim. Po pierwszym uruchomieniu strony element ten nie posiada dzieci.

Dziećmi elementu o id "content" są obiekty klasy `dash_core_components.Graph`. Elementy te zainicjowane są danymi o poziomie odpłatności świadczeniobiorcy na tabletkę lub czopek. Każdy wykres odpowiada jednemu poziomowi odpłatności leku. Wykresy wyświetlają listę znalezionych alternatyw leków w postaci legendy wykresu. Zgodnie z wymaganiami funkcyjnymi kliknięcie na element tej listy powoduje pojawienie się lub zniknięcie na wykresie punktów odpowiadających danej alternatywie leku. Akcje te są obsługiwane przez przeglądarkę i nie wymagają komunikacji z serwerem. Tytuł wykresu to poziom odpłatności, któremu odpowiada wykres.

Na początku rozdziału znajduje się makietą przedstawiająca wygląd uzyskanej strony po wyszukaniu alternatyw leku. Przed wyszukaniem wyświetlony jest jedynie pole do wyszukiwania.

Połączenia

W trakcie funkcjonowania aplikacji poszczególne części, opisane w poprzednim rozdziale tego dokumentu, komunikują się ze sobą w następujący sposób.

Serwer - Baza danych

Gdy serwer posiada wprowadzony przez użytkownika EAN leku (poniżej przedstawiony jako {user_ean}), może zadać następujące zapytanie:

```
SELECT refund_level,  
       description,  
       array_agg(announcement_date) AS announcement_dates,  
       array_agg(unit_price) AS unit_prices  
FROM refunds  
WHERE (active_ingredient,  
       dose,  
       form) IN
```

```
(SELECT active_ingredient,
        dose,
        form
FROM refunds
WHERE ean = {user_ean})
GROUP BY refund_level,
        description
ORDER BY refund_level;
```

Z odpowiedzi powyższego zapytania serwer jest w stanie uzyskać dane potrzebne, by umieścić punkty na wszystkich prezentowanych wykresach. Po stronie serwera znajduje się obowiązek posortowania przedstawionych w legendach wykresów alternatyw zgodnie z wymaganiami funkcjonalnymi.

Serwer - Front-end

Połączenie między przeglądarką a serwerem realizowane jest przy pomocy dashowego dekoratora oraz funkcji `get_graphs`:

```
@app.callback(Output('content', 'children'), [Input('dropdown', 'value')])
def get_graphs(ean):
```

Moduł dash uruchomi funkcję `get_graphs` na serwerze za każdym razem, gdy użytkownik wybierze ean leku przy pomocy elementu “dropdown” (będzie to wiązać się z komunikacją między przeglądarką użytkownika oraz serwerem).

We wnętrzu funkcji zachodzi komunikacja z bazą danych opisana w poprzednim rozdziale. Funkcja `get_graphs` zwraca obiekt reprezentujący nową postać html-owego elementu o id “contents”. Wykresy znajdujące się w zwróconym elemencie zostały zainicjowane znalezionymi danymi. Moduł dash zajmuje się wysłaniem do przeglądarki komunikatu nakazującego jej ponowne wyrenderowanie zmienionych elementów.