Architektura projektu

System składa się z frontendu będącego GUI użytkownika, serwera WWW oraz bazy danych. Poniżej opisane są z technicznego punktu widzenia poszczególne jego komponenty oraz sposób interakcji pomiędzy nimi.

Serwer WWW

Serwer WWW jest realizowany jako projekt we frameworku Django. Strona główna aplikacji stanowi pojedynczy widok (django view), który korzysta z szablonu, którego szkic z kolei jest zawarty w osobnym pliku .html. Styl szablonu jest określony w osobnym arkuszu css i dodatkowo modyfikowany za pomocą JavaScript w zależności od treści strony. Również za pomocą JavaScript zrealizowane jest chowanie i pokazywanie części zawartości strony, w zależności od czynności użytkownika. Za treść szablonu odpowiadają przede wszystkim wstawki do pliku .html generowane automatycznie przez Django na podstawie bazy danych i aktualnego stanu aplikacji. Filtrowanie wierszy tabeli jest zrealizowane z wykorzystaniem biblioteki django-filters, natomiast stronicowanie - z wykorzystaniem django - tables. W momencie otrzymania komunikatu od przeglądarki użytkownika serwer aktualizuje zawartość strony i odświeża ją. Ponadto serwer pamięta aktualne dane dotyczące stanu aplikacji, takie jak: obecna strona, kryteria filtrowania, sposób sortowania.

Baza danych

Baza danych jest bazą SQLite.
Zawiera ona dane dotyczące leków zgromadzone w pojedynczej tabeli o następujących kolumnach:
med name - pole tekstowe, max. długość 100
gtin - pole tekstowe, max. długość 50
registered funding - pole tekstowe
nonregistered funding - jak wyzej
pub date - pole daty
active substance - pole tesktowe, max. długość 100
med form - pole tekstowe, max. długość 40
dose - jak wyżej
pack size - pole tekstowe, max. długość 50
limit group - pole tekstowe
official price - liczba zmiennoprzecinkowa z dokładnością do dwóch cyfr po przecinku,
max. długość 8
wholesale price - -jak wyżej-
retail price - jak wyżej
refund limit - jak wyżej
payment level - enumerator, element ze zbioru 30%, 50%, ryczałt, bezpłatne
patient payment - liczba zmiennoprzecinkowa z dokładnością do dwóch cyfr po przecinku,
max. długość 8,
company name - pole tekstowe, max. długość 40,

last changed - pole daty,
diff pk - klucz obcy łączący rekordy opsisujące ten sam lek w różnych rozporządzeniach.
Zbiór kolumn: gtin, registered funding, nonregistered funding stanowi klucz główny tabeli. Tabela jest realizowana za pomocą dostępnego w Django mechanizmu modeli, to znaczy każdy wiersz jest obiektem klasy Medicine dziedziczącej po wbudowanej klasie Model. Każda kolumna ma w projekcie Django jest obiektem jednej z wbudowanych w Django klas dziedziczących po klasie Field (np. TextField, IntegerField), o parametrach (np. maksymalna długość) zgodnych z powyższym opisem. Kolumny: company name, last changed, diff pk nie są wyświetlane w tabeli. Pierwsze z nich służy do filtrowania po nazwie producenta, a drugie - do określenia, czy dane leku zmieniły się między jednym a drugim rozporządzeniem.

Forma komunikacji przeglądarka - serwer

Przeglądarka może do serwera wysłać następujące typy komunikatów (z których każdy obsługiwany jest poprzez wywołanie odpowiedniego widoku):
Filtruj (wciśnięcie jakiegoś guzika): wysyła do serwera polecenie sortowania wraz z zawartościami poszczególnych pól tekstowych filtrowania.
Posortuj: polecenie sortowania wg. danej kolumny, w treści zawarta jest nazwa tej kolumny.
Strona: wysyła polecenie pokazania danej strony, w treści zawarty jest numer.
Wszystkie komunikaty są realizowane za pomocą metody GET.

Forma komunikacji serwer - przeglądarka

Serwer w odpowiedzi na żądanie przeglądarki przesyła komunikat http zawierający odpowiednio zmodyfikowaną stronę w postaci pliku html. Należy zauważyć, że funkcjonalności zrealizowane za pomocą JavaScript nie wymagają w ogóle komunikacji pomiędzy przeglądarką a serwerem.

Forma komunikacji serwer - baza danych oraz baza danych - serwer

Komunikacja realizowana jest w całości poprzez wbudowane mechanizmy Django. Serwer używa bazodanowego API Django do otrzymywania danych z bazy w postaci obiektów Pythona.