

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 składa się z widoków, który realizują ten sam, napisany w html i stylizowany za pomocą arkusza css szablon. W momencie otrzymania komunikatu od przeglądarki użytkownika serwer wywołuje odpowiedni widok i odświeża szablon, a następnie wysyła go do przeglądarki. Serwer pamięta aktualne dane dotyczące stanu aplikacji, takie jak: obecna strona, obecne kryteria filtrowania, obecny sposób sortowania.

### Baza danych

Baza danych jest bazą postgreSQL.

Zawiera ona dane dotyczące leków zgromadzone w pojedynczej tabeli o następujących kolumnach:

med\_name - pole tekstowe, max. długość 100

gtin - pole tekstowe, max. długość 20

registered\_funding - pole tekstowe, max. długość 200

nonregistered\_funding - -----;-----

pub\_date - pole typu DateField

active\_substance - pole tekstowe, max. długość 100

med\_form - pole tekstowe, max. długość 40

med\_dose - -----;-----

pack\_size - pole tekstowe, max. długość 50

limit\_group - pole tekstowe, max. długość 100

official\_cost - liczba zmiennoprzecinkowa z dokładnością do dwóch cyfr po przecinku, max. długość 5

wholesale\_cost - -----;-----

retail\_cost - -----;-----

refund\_limit - -----;-----

payment\_level - enumerator, element ze zbioru {30%, 50%, ryczałt, bezpłatne}

patient\_payment - liczba zmiennoprzecinkowa z dokładnością do dwóch cyfr po przecinku, max. długość 5

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.

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: jest to osobny typ komunikatu dla każdej kolumny, po której można sortować.

Następna/poprzednia strona: bez względu na aktualną stronę wywoływany jest ten sam widok.

Wszystkie komunikaty są realizowane za pomocą metody POST.

Forma komunikacji serwer - przeglądarka

Serwer w odpowiedzi na żądanie przeglądarki przesyła komunikat http zawierający odpowiednio zmodyfikowaną stronę w postaci pliku html.

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 Python, filtrowania tych obiektów oraz sortowania.