TELM

Aplikacja Web z prezentacją mobilną - e-karty gorączkowe dla oddziału szpitalnego

Antonina Andrzejewska, Maria Baczewska

**Główne założenia:**

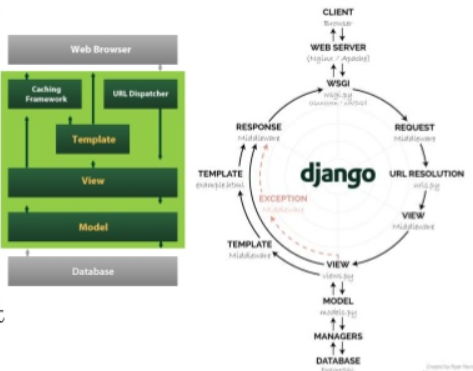
Celem pracy jest wykonanie aplikacji dla oddziału szpitalnego do kontrolowania pacjentów, w szczególności temperatury ciała.

**Funkcjonalność:**

Aplikacja spełniać będzie funkcję kontrolną dla pielęgniarek i lekarzy, którzy będą mogły monitorować stan temperatury pacjenta. Do aplikacji będzie miał dostęp administrator, lekarze oraz pielęgniarki. Dostęp będzie się otrzymywać poprzez logowanie na poszczególne spersonalizowane konta. Administrator będzie posiadać najwyższe uprawnienia – będzie mógł dodawać i usuwać użytkowników i nadawać im uprawnienia. Będzie miał również dostęp do wszystkich innych funkcjonalności. Lekarz będzie mógł dodawać nowych pacjentów oraz wypełniać informacje na temat temperatury pacjenta. Pielęgniarka w swoich uprawnieniach będzie mieć tylko dodawanie kolejnych wartości temperatury pacjenta oraz sprawdzanie poprzednich wyników.

Każdy z użytkowników będzie miał możliwość wyszukiwania i sprawdzenia danych pacjenta na podstawie imienia, nazwiska, PESEL.

**Struktura projektu:**



Rysunek 1 Struktura MVC Django

**Realizacja:**

Projekt aplikacji webowej zostanie zrealizowany w języku Python w wersji 3.6 na bazie otwartego frameworku Django. Jako środowisko programistyczne użyty będzie program ATOM. Do realizacji bazy danych będzie wykorzystany serwer Apache z mod\_pythonem. Baza będzie stworzona przy użyciu SQLite. Baza danych będzie zawierała następujące encje:

* Pacjenci
* Lekarze
* Pielęgniarki
* Informacja medyczna
* Pomiary

Encja pacjenci połączona będzie z encją Informacja medyczna, a ta z encją Pomiary. Drugą relacją jest połączenie encji Lekarze z encją Pomiary, trzecią encja Pielęgniarki z encją Pomiary.

Do realizacji wyżej opisanych funkcjonalności stworzono szereg modeli, szablonów oraz widoków.

**Modele:**

Stworzono modele MyUser, Doctor, Nurse, Patient, Karta, Measurement.

Model MyUser posiada informację o roli użytkownika oraz dane spersonalizowane konta (imię, nazwisko, hasło). Modele Doctor, Nurse przechowują dane o tożsamości lekarza i pielęgniarki oraz są w relacji z MyUser.

Model Patient posiada informacje o danych pacjenta, pacjent posiada lekarza (odniesienie do modelu Doctor). Każdemu pacjentowi przypisana jest karta, w której prócz danych pacjenta przechowywane są dane o dacie rejestracji oraz opis choroby. Ostatnim modelem jest model Measurement do którego przypisany jest pacjent oraz lekarz lub pielęgniarka. Jest to model używany do wpisywania temperatury, posiada zmienną value (wartość temperatury).

**Szablony:**

Stworzono trzy szablony w języku HTML.

Szablon index.html jest szablonem wejściowym, w którym mamy możliwość wyboru listy użytkowników lub listy użytkowników.

Szablon patients\_details.html jest szablonem, który służy do wyświetlania danych pacjenta wraz ze wszystkimi pomiarami temperatury. W tym szablonie stworzona jest możliwość dodawania temperatury.

Szablon patients\_list.html jest szablonem wykorzystywanym do wyświetlania listy pacjentów. Jest również możliwość dodawania pacjenta.

**Widoki:**

Stworzono widoki index, patients\_list, patients\_details, patient\_add, measurement\_add. Stworzona również wstępny widok do tworzenia użytkowników – user\_add. Widoki index, patients\_list, patients\_details odnoszą się do odpowiadającym im z nazwy szablonów. Widoki measurement\_add oraz patient\_add po wykonaniu się wracają na stronę z szablonu odpowiednio patients\_details i patients\_list.

URL:

Swtorzone zostały również odpoweidnie adresy URL:

*url(r'^$', views.index, name='index'),*

*url(r'^patient/list/', views.patient\_list, name='patient\_list')*

*url(r'^patient/([0-9]+)/', views.patient\_details, name='patient\_details'),*

*url(r'^patient/add/', views.patient\_add, name='patient\_add'),*

*url(r'^measurement/add/', views.measurement\_add, name='measurement\_add'),*

**Działanie:**

Aplikacja została wdrożona za pomocą pythonanywhere.com. PythonAnywhere to internetowe zintegrowane środowisko programistyczne (IDE) i usługa hostingu WWW oparta na języku programowania Python. Zapewnia dostęp w przeglądarce do bazujących na serwerach interfejsów Python i Bash Command-Line, a także edytor kodu z podświetlaniem składni. Pliki programów mogą być przesyłane do iz usługi przy użyciu przeglądarki użytkownika. Aplikacje internetowe hostowane przez tę usługę można zapisywać za pomocą dowolnej platformy aplikacji opartej na WSGI.

Stworzona strona powinna znajdować się pod adresem *baczi.pythonanywhere.com*. Wyświetla się jednak jedynie tylko szablon index.html. Aplikacja w wersji opisanej powyżej działa w pełni na lokalnym wirtualnym serwerze.

**Problemy:**

Program zostanie uzupełniony o stworzenie odpowiedniego modelu i szablonu tworzenia użytkownika i logowania. Zostanie naprawiony problem dostępności aplikacji na stronie *baczi.pythonanywhere.com.*

Niestety w pisaniu programu nie została zachowana konsekwencja w użyciu języka polskiego, przez zbyt duży podświadomy wpływ funkcji Pythona w języku angielskim.

**Podsumowanie:**

Aplikacja ma za zadanie wizualizację i monitorowanie wyników pomiaru temperatury pacjenta. Pomiarów mogą dokonywać lekarze i pielęgniarki. Są oni głównymi odbiorcami tworzonej aplikacji. Stwarza ona możliwość łatwego monitorowania zmian temperatury ciała i kontrolę nad pacjentem. Rozwiązanie w postaci budowania aplikacji na bazie frameworku Django daje pewność przejrzystości stworzonej aplikacji oraz prostoty późniejszego używania.