

Warszawa, 12.04.2016

Web-Notifier

Projekt techniczny systemu

Dokument opisuje projekt realizowany w ramach zajęć z Inżynierii Oprogramowania na wydziale MIM UW w roku akademickim 2015/16.

Wersja: 1.0

Zespół: Marek Mystkowski, Adam Sołtysik

Klient: mgr Grzegorz Grudziński

Spis treści:

1. Wprowadzenie
2. Serwer
3. Baza danych
4. Aplikacje
5. Działanie algorytmu
6. Powiadomienia

1) Wprowadzenie

Celem dokumentu jest przedstawienie ogólnej koncepcji oprogramowania wraz z omówieniem poszczególnych modułów i komponentów. Dokument ten stanowi uzupełnienie Wizji projektu oraz Wymagań projektowych.

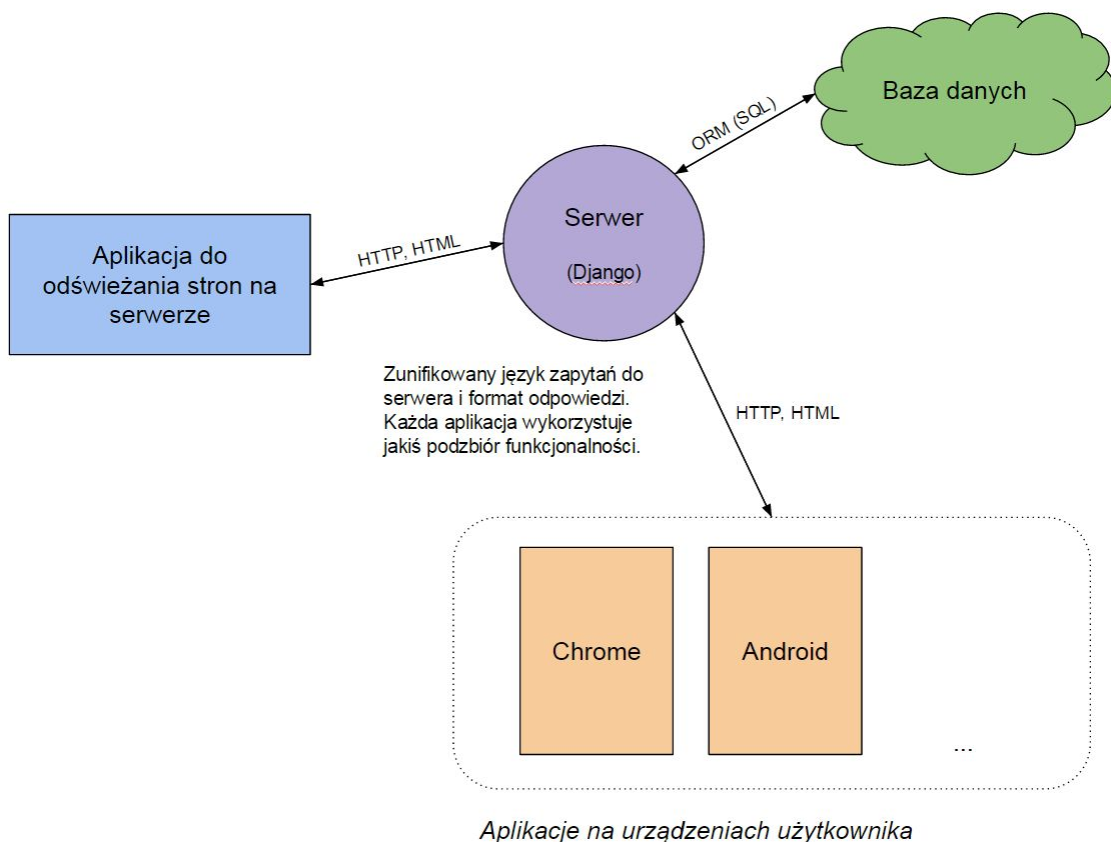
2) Serwer

Głównym zadaniem serwera jest obsługa bazy danych oraz wszystkich aplikacji. Baza danych będzie obsługiwana przy użyciu wbudowanego w Django systemu ORM. Serwer będzie przyjmował zapytania aplikacji za pomocą protokołu HTTP oraz generował odpowiedzi w formacie HTML.

Zapytania HTTP wysyłane na serwer będą umożliwiały:

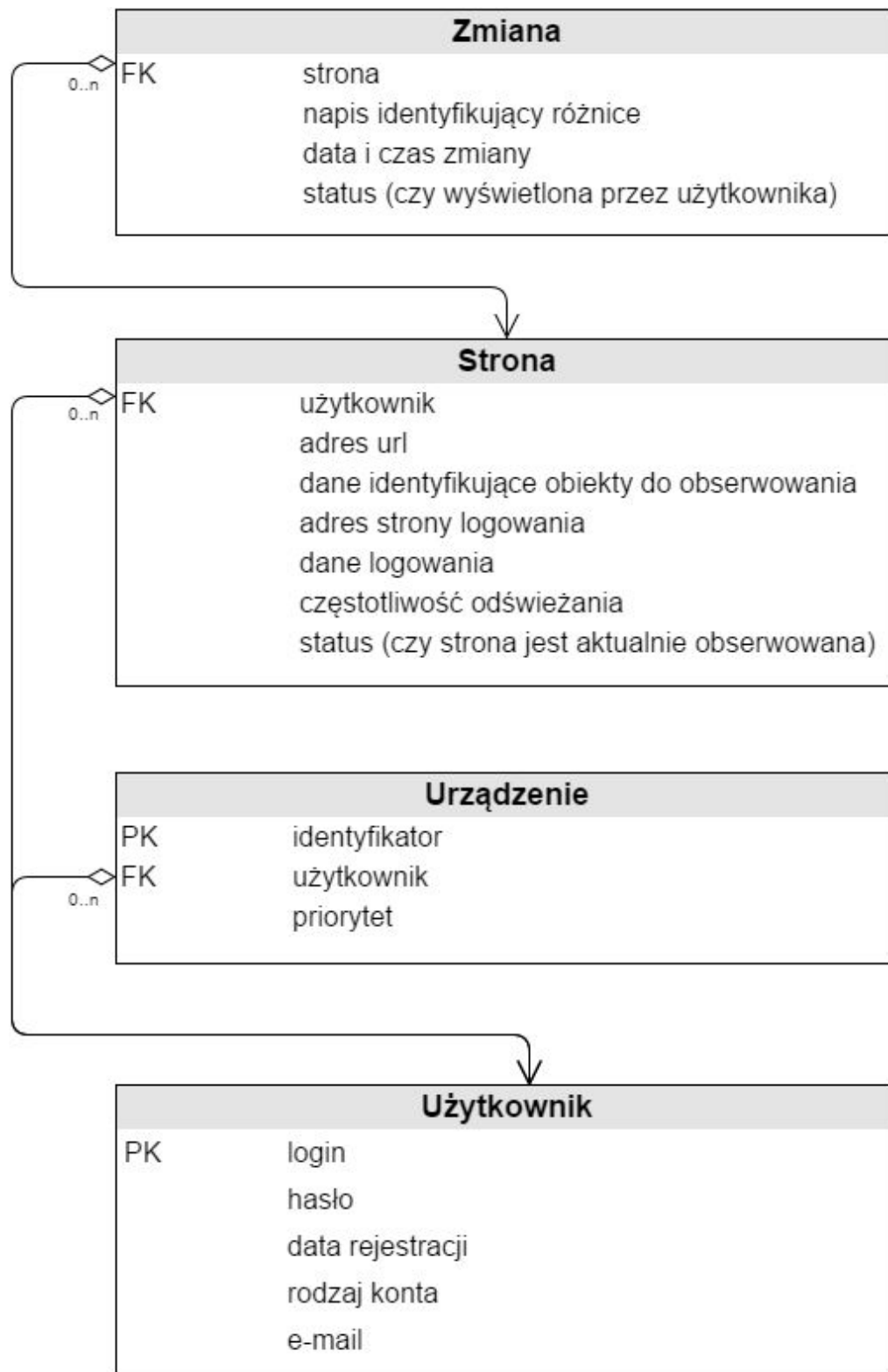
- rejestrację, logowanie użytkownika oraz edycję jego danych,
- dodawanie nowych stron do obserwacji i zmianę ich ustawień,
- wprowadzanie informacji o zmianach i pobieranie ich historii.

Ponadto na serwerze będzie działała aplikacja służąca do globalnego obserwowania stron dla wszystkich użytkowników.



3) Baza danych

Baza danych będzie zawierała następujące tabele zdefiniowane jako modele w aplikacji Django:



4) Aplikacje

a) Strona WWW

Strona WWW będzie wyświetlana z użyciem HTML5 i CSS. Po zalogowaniu użytkownik może przeglądać listę obserwowanych stron oraz historię zmian. Dodawanie nowych stron oraz ich edycja jest możliwa dzięki współpracy z rozszerzeniem do przeglądarki.

b) Rozszerzenie do przeglądarki (Chrome)

Rozszerzenie napisane będzie z użyciem HTML5 i Javascript i będzie współpracować ze stroną WWW. Jego zadaniem jest umożliwienie łatwego dodawania i edycji obserwowanych stron. Pierwszym krokiem jest przejście na stronę, na której użytkownik wskazuje elementy, w obrębie których interesują go zmiany w treści. Drugim (opcjonalnym) krokiem jest przejście na stronę logowania i wpisanie tam swoich danych.

Wszystkie niezbędne informacje są następnie przesyłane na serwer, gdzie są weryfikowane. W przypadku nie stwierdzenia błędów, serwer dodaje nowy wpis do bazy danych. W przeciwnym razie użytkownik proszony jest o poprawienie danych.

c) Aplikacja desktopowa (Windows)

Aplikacja uruchamiana na komputerze będzie napisana w Javie i będzie implementowała główną funkcjonalność systemu. Po uruchomieniu aplikacji oraz w określonych odstępach czasu z serwera pobierana będzie lista stron obserwowanych przez użytkownika, które aplikacja następnie będzie odświeżać z ustaloną częstotliwością w celu poszukiwania zmian w treści.

d) Aplikacja mobilna (Android)

Aplikacja mobilna będzie napisana w Javie. Służyć będzie do wyświetlania informacji o koncie użytkownika oraz będzie umożliwiała przeglądanie historii zmian na obserwowanych stronach. Dodatkowo urządzenie mobilne będzie wyświetlać powiadomienia informujące użytkownika o nowych zmianach od razu po ich wykryciu.

e) Aplikacja na serwerze

Ta aplikacja również będzie napisana w Javie i jej funkcjonalność będzie pokrywała się z funkcjonalnością zwykłej aplikacji desktopowej. Jej zadaniem jest odświeżanie wszystkich stron tych użytkowników, którzy w danym momencie nie są zalogowani na żadnym urządzeniu, z wyjątkiem tych stron, które przekraczają limity przypisane do danego konta.

5) Działanie algorytmu

Urządzenie, na którym ma się odbywać odświeżanie stron użytkownika, będzie wybierane na podstawie priorytetów ustawionych przez użytkownika. Wybrane zostanie urządzenie o najwyższym priorytecie spośród tych, na których aktualnie jest uruchomiona aplikacja. W przypadku braku dostępnych urządzeń, działanie algorytmu zostanie przekierowane do aplikacji na serwerze.

Elementy do obserwacji identyfikowane będą za pomocą ścieżki w drzewie DOM. Algorytm przy każdym sprawdzeniu porównuje kod HTML wybranych elementów na stronie z ich stanem z poprzedniej iteracji. W przypadku stwierdzenia zmiany, aplikacja wysła informację do serwera, który następnie zapisuje zmianę w bazie danych i wysła powiadomienia do użytkownika.

Jeżeli strona wymaga logowania, program musi za każdym razem dodatkowo sprawdzić, czy nie nastąpiło wygaśnięcie sesji. W przypadku stwierdzenia przekierowania na stronę logowania, aplikacja wprowadzi dane użytkownika i ponownie zaloguje się na docelową stronę.

6) Powiadomienia

Powiadomienia będą wysyłane przez serwer za pomocą usługi Google Cloud Messaging oraz na adres e-mail użytkownika. Usługa GCM zapewni wyświetlanie powiadomień w przeglądarce użytkownika oraz na urządzeniach z systemem Android.