

Warszawa, 25.04.2016

# **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:** 1.1

**Zespół:** Tomasz Kacperek, 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.

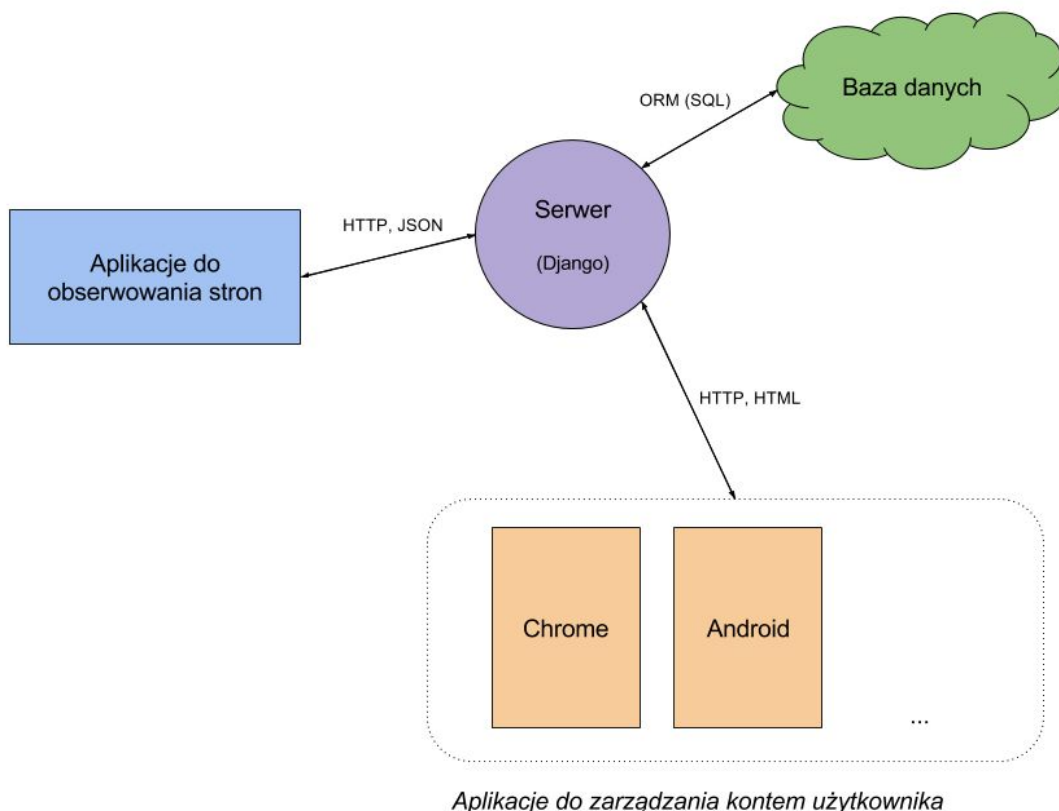
## 2) Serwer

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 za pomocą protokołu HTTP oraz generował odpowiedzi w formacie HTML (dla użytkownika) oraz JSON (w przypadku wewnętrznych danych dla aplikacji).

Zapytania HTTP wysyłane na serwer będą umożliwiały:

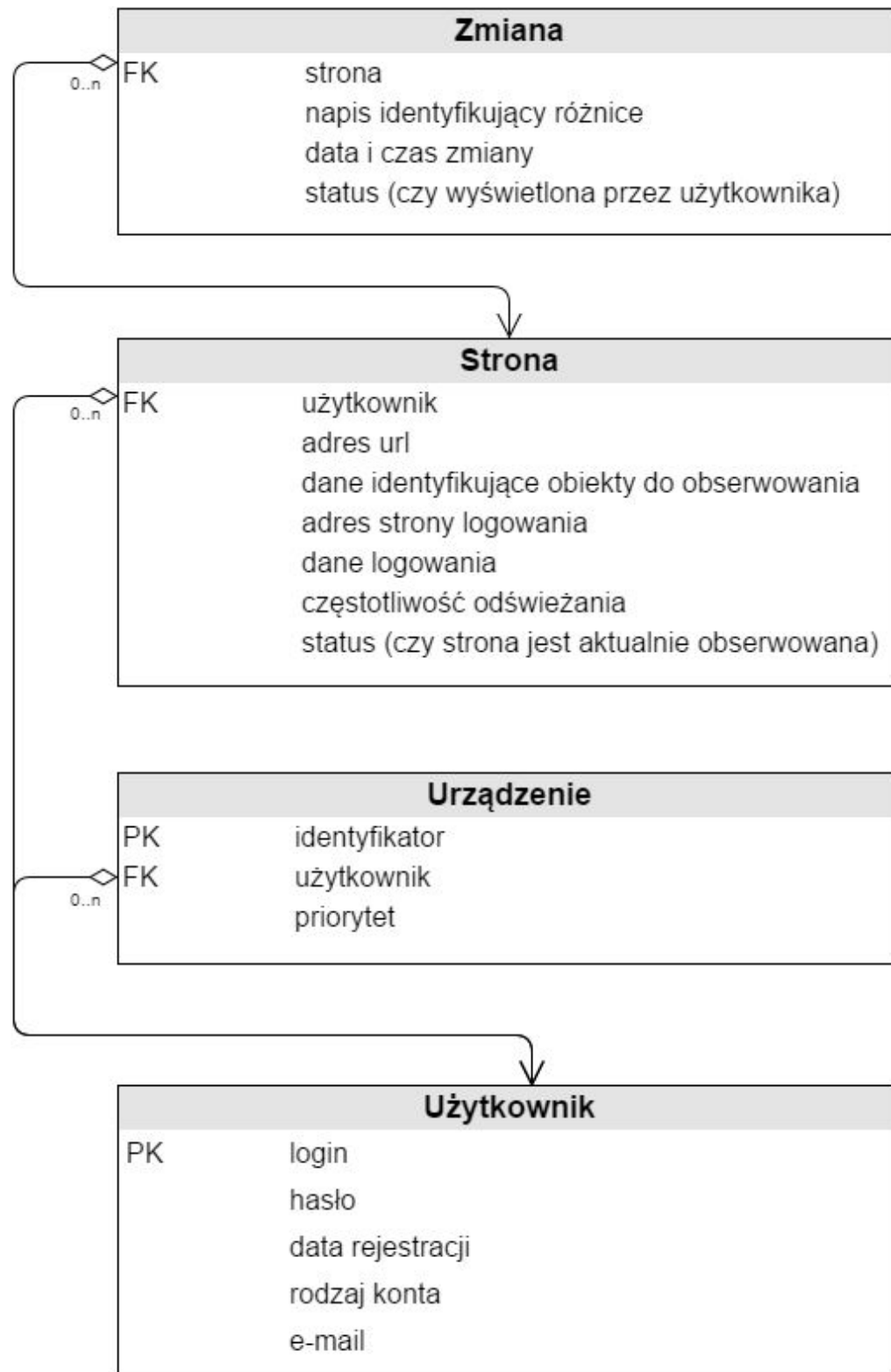
- rejestrację, logowanie użytkownika oraz edycję jego danych,
- dodawanie nowych stron do obserwacji i zmianę ich ustawień,
- wiązanie urządzeń użytkownika z jego kontem i edycję priorytetów,
- wprowadzanie informacji o zmianach i pobieranie ich historii.

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 i historię zmian, a także zmieniać priorytety dla swoich urządzeń powiązanych z kontem. 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. Rozszerzenie pobiera dane logowania od użytkownika oraz umożliwia wskazanie odpowiednich obiektów do monitorowania bezpośrednio na danej stronie w przeglądarce.

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 do odświeżania stron**

Aplikacja będzie napisana w Pythonie i będzie implementowała główną funkcjonalność systemu. Aplikacja może mieć uprawnienia jednego użytkownika lub całego serwera nadane przez administratorów. W przypadku braku specjalnych uprawnień, przy pierwszym uruchomieniu aplikacja generuje swój unikalny klucz, który zostaje powiązany z kontem użytkownika. W trakcie działania program w określonych odstępach czasu pobiera z serwera listę stron, które następnie odświeża 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.

### **5) Działanie algorytmu**

Urządzenie, na którym ma się odbywać odświeżanie stron użytkownika, będzie wybierane na podstawie priorytetów ustalonych 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 wysyła informację do serwera, który następnie zapisuje zmianę w bazie danych i wysyła powiadomienia do użytkownika.

Jeżeli strona wymaga logowania, program musi za każdym razem dodatkowo sprawdzić, czy nie nastąpiło przerwanie sesji, oraz w razie potrzeby zalogować się ponownie.

Algorytm musi być ponadto odporny na błędy takie jak: zmiana struktury strony, nieaktualne dane logowania, przekroczony czas połączenia ze stroną. W przypadku stwierdzenia błędu, aplikacja przesyła informację do serwera, który następnie zmienia status danej strony w bazie oraz wysyła powiadomienie do użytkownika z prośbą o aktualizację danych.

### **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.