

# TECHNIKI ANALIZY SIECI SPOŁECZNYCH

Koncepcja wykonania projektu

Projekt nr 9 – O jakie miejsca lub podmioty pytają obywatele Urzędu Miasta w Lublinie w postaci wniosków o udostępnienie informacji publicznej? Wyniki zobrazuj na mapie.

Wojciech Kaczmar (246402)

Michał Haponiuk (249371)

Listopad 2016

## SPIS TREŚCI

Cel Aplikacji .....	2
Wytypowanie źródeł danych.....	2
Technologie pobrania i przechowywania danych .....	2
Przetwarzanie danych .....	2
Scenariusz użycia aplikacji.....	3
Technologia wykonania aplikacji.....	3
Część Server-Side .....	3
Część Client-Side .....	4
1. Google Maps JavaScript API .....	4
2. Google Maps Geocoding API .....	4
3. Date range picker .....	4

## CEL APLIKACJI

Podejrzewamy, iż użytkownicy składający wnioski o udzielenie informacji publicznej, robią to często z powodu podejrzeń o nieprawidłowości w działaniu instytucji publicznych. Aplikacja pozwoliłaby na sprawne wyszukiwanie miejsc bądź regionów w Lublinie, odnośnie których mogły zostać popełnione przez urzędników niewłaściwe bądź nawet niezgodne z prawem decyzje. Pomocna mogłaby okazać się dla np. reporterów i dziennikarzy w ich codziennej pracy.

## WYTYPOWANIE ŹRÓDEŁ DANYCH

Jako źródło danych posłuży nam strona <https://danepubliczne.gov.pl>, na której znajdują się zbiory danych różnych dostawców oraz urzędów. Na potrzeby naszego projektu skorzystamy ze zbioru udostępnionego przez Urząd Miasta Lublin. Podmiotem naszej analizy będzie zbiór wniosków o udzielenie informacji publicznej.

## TECHNOLOGIE POBRANIA I PRZECHOWYWANIA DANYCH

Do pobierania wniosków jest dostarczone API, które w odpowiedzi na żądanie HTTP GET zwraca wnioski w postaci pliku JSON. Informacje dotyczące źródła danych jak i same dane dostępne są pod adresem <https://danepubliczne.gov.pl/dataset/wnioski-o-udzielenie-informacji-publicznej>.

Wnioski, które już zostały złożone nie będą się zmieniać. Z tego powodu nasza aplikacja pobierze dostępne wnioski i po naszej stronie będą przechowywane w bazie danych w celu zapewnienia szybkiego wielodostępu oraz umożliwienia optymalnego wyszukiwania wniosków z zadanego okna czasowego. Aplikacja będzie cyklicznie sprawdzać czy się pojawiły nowe wnioski i w razie potrzeby dociągnie je w celu zapewnienia aktualności posiadanych przez nas danych. Takie rozwiązanie pozwoli ograniczyć niepotrzebny ruch sieciowy i zminimalizuje niezbędną ilość zapytań do serwerów rządowych.

Część serwerowa wystawi własne API HTTP, pozwalające na pobieranie potrzebnych danych aplikacji klienckiej.

## PRZETWARZANIE DANYCH

Odpowiedź serwera rządowego zwraca wnioski zawierające pola:

- id - identyfikator wniosku
- link - link do strony zawierającej wniosek
- pyt - treść wniosku
- odp - treść odpowiedzi
- data - data złożenia wniosku

Pole 'pyt' jest dla naszej aplikacji kluczowe. W znacznej ilości przypadków zawiera dane dotyczące lokalizacji obiektu będącego zainteresowaniem osoby składającej wniosek. Treść w tym polu może być dowolna, nie ma określonej struktury. Aby wyciągnąć potrzebne dane wykorzystamy wyrażenia regularne poszukujące fragmenty zawierające takie słowa jak np. *ulica, aleja, skwer, plac, park* itp.

Wnioski przechowywane w naszej bazie danych będą posiadały dodatkowo wyekstrahowaną informację w celu uproszczenia późniejszego procesu nanoszenia markerów na mapę.

## SCENARIUSZ UŻYCIA APLIKACJI

Celem projektu jest dostarczenie aplikacji webowej, która będzie obrazowała na mapie miejsca oraz podmioty, o jakie pytali mieszkańcy Lublina w postaci wniosków o udostępnienie informacji publicznej.

Aktualnie zbiór danych zawiera 973 wnioski. Aplikacja będzie umożliwiała ograniczenie wyświetlanych wyników na mapie za pomocą zdefiniowania zakresu dat, z którego będą wyświetlane wnioski. Inicjalnie, będą wyświetlane miejsca w postaci markerów z ostatnich 30 dni. W momencie zmiany zakresu dat przez użytkownika, aplikacja kliencka wykona zapytanie do bazy danych po listę wniosków z nowego zakresu, a następnie odświeży wyniki na mapie. Po kliknięciu myszką w dane miejsce zaznaczone, w postaci markera, użytkownikowi wyświetlą się dodatkowe informacje na temat tego miejsca, w postaci dymku. Informacjami tymi będą m.in. adres miejsca, data złożenia wniosku związanego z tym miejscem oraz link do pełnej treści wniosku.

## TECHNOLOGIA WYKONANIA APLIKACJI

### Część Server-Side

Backend zostanie napisany w języku Python. Do pobrania danych z serwerów rządowych wykorzystamy bibliotekę *requests* - określanej mianem "http dla ludzi" (z ang. "http for humans": [docs.python-requests.org](https://docs.python-requests.org/)).

W celu wyekstrahowania danych lokalizacyjnych z wniosków, użyjemy biblioteki wyrażeń regularnych *re* z biblioteki standardowej Python'a ([docs.python.org/3/library/re.html](https://docs.python.org/3/library/re.html)).

Przechowywać dane będziemy w relacyjnej bazie danych *SQLite* ([sqlite.org](https://sqlite.org)).

Do komunikacji z bazą wykorzystamy mapper obiektowo-relacyjny *SQLAlchemy* ([sqlalchemy.org](https://sqlalchemy.org)).

Usługa serwująca dane do aplikacji klienckiej zostanie zaimplementowana z użyciem serwera HTTP *Flask* ([flask.pocoo.org](https://flask.pocoo.org)).

# Część Client-Side

## 1. GOOGLE MAPS JAVASCRIPT API

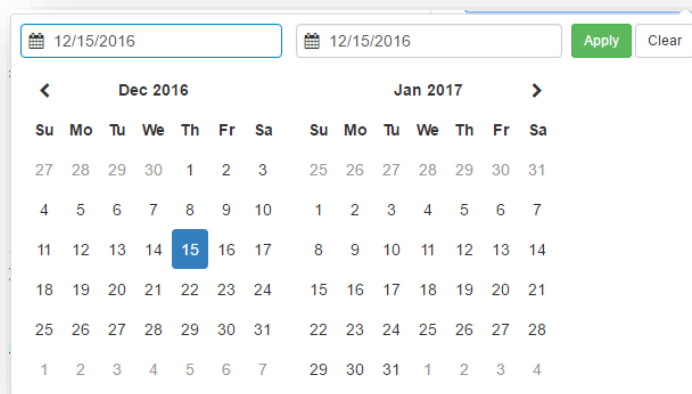
Aplikacja kliencka zostanie wykonana w technologii JavaScript, w wersji ecmaScript 2015. Głównym elementem aplikacji będzie mapa. Do wyświetlania mapy wykorzystamy Google Maps JavaScript API, które pozwala na wyświetlenie mapy wyśrodkowanej w konkretnym punkcie (w naszym przypadku będzie to centrum miasta Lublin) oraz zaznaczanie na mapie punktów charakterystycznych w postaci markerów. Mapa będzie posiadała standardowe funkcje dostępne w aplikacjach korzystających z google maps, takie jak: zbliżanie/oddalanie mapy, przesuwanie, przełączanie się pomiędzy widokiem drogowym i satelitarnym

## 2. GOOGLE MAPS GEOCODING API

W celu narysowania markera w odpowiednim miejscu, będziemy potrzebować współrzędnych geograficznych tego miejsca. Do konwersji adresu pocztowego na jego współrzędne geograficzne wykorzystamy Google Maps Geocoding API. Z racji tego, że w Polsce może być kilka miejsc z tą samą nazwą ulicy i numerem budynku, zapewnimy, że zostaną zakodowane miejsca tylko z Lublina, poprzez dopisanie „Lublin” w adresie.

## 3. DATE RANGE PICKER

Użytkownik będzie mógł określać zakres dat, z której będą wyszukiwane wnioski. W tym celu skorzystamy z gotowej biblioteki JavaScript Date Range Picker, którą zaimportujemy do naszej aplikacji. Komponent pozwala na ręczne wpisanie daty w polu tekstowym lub wybór daty z kalendarza wyświetlającego się w momencie kliknięcia w pole. Date Range Picker posiada szereg walidacji np. walidacja poprawnego formatu daty. Do poprawnego działania komponentu należy również zaimportować JQuery, Moment (biblioteki JavaScript) oraz Bootstrap.css



*Komponent Date Range Picker*