**SPDB Dokumentacja wstępna**

Prawidłiowa identyfikacja przystanków oraz obiektów znajdujących się wokół nich na obszarze miasta Białystok.

Kamil Bachanek

Karol Rogowski

Adam Zieliński

1. **Założenia wstępne**

Dla danych dotyczących przejazdów komunikacji miejskiej należy dokonać identyfikacji przystanków z przypisaną błędną lokalizacją i określić ich rzeczywiste położenie (np. przez analizę pozycji pojazdów o prędkości 0 w zadanym promieniu o istniejącej pozycji). Projekt zakłada również znajdowanie dodatkowych popularnych miejsc najczęściej odwiedzanych przez ludzi (np. siłownia, basen, hala sportowa, kino, biurowiec korporacyjny) w pobliżu znajdujących się przystanków autobusowych.

1. **Ogólny zarys realizacji problemów**
2. Identyfikacja przystanków z przypisaną błędną lokalizacją

Należy wyznaczyć miejsca, w których autobus się zatrzymywał, wprowadza to dodatkowe dane, uwzględniające postój na trasie w korkach, zatrzymywanie na światłach oraz przystanki. Po zgromadzeniu tych danych w kontekście całego dnia, należałoby wyszczególnić godziny, w których autobus jest szczególnie narażony na stanie w korku oraz zatrzymywanie na światłach. Skutkowałoby to różnicą w ilości punktów zatrzymań dla poszczególnych godzin. Porównując ze sobą dane dla poszczególnych okresów czasowych możliwe jest wyszczególnienie oraz odfiltrowanie punktów, których powtarzalność jest mniejsza od przyjętego empirycznie progu.

1. Lokalizacja ciekawych miejsc wokół przystanków

Po określeniu właściwego położenia przystanku należy wyznaczyć popularne obiekty w jego sąsiedztwie. W tym celu wykorzystane zostanie Google Maps Api. W ustalonym przez użytkownika promieniu nastąpi wyszukanie miejsc ze zdefiniowanej listy, przykładowe miejsca to sklep, siłownia, basen, hala itp. Możliwe byłoby również pokazanie wszystkich typów miejsc zdefiniowanych na liście w określonym promieniu wokół obranego przystanku. Umożliwione również zostałoby naniesienie poszczególnych typów miejsc, bądź też wszystkich ciekawych miejsc na trasę przejazdu autobusu w określonym promieniu od każdego z przystanków. Realizacja nastąpi przy wykorzystaniu serwisu google.maps.places.PlacesService wraz z metodą nearbySearch.

1. **Architektura i technologie**

Aplikacja zostanie zrealizowana w postaci aplikacji Webowej. Architektura wyróżnia dwie warstwy: klient oraz serwer. Warstwa klienta zakłada wykorzystanie technologii HTML5, angularJS oraz Google Maps Api. Część serwerowa zostanie zrealizowana przy wykorzystaniu języka Python w wersji 3.6 wraz z wykorzystaniem biblioteki Flask zajmującej się zarządzaniem zapytaniami, wywoływaniem odpowiednich funkcji oraz wysyłaniem odpowiedzi. W celu komunikacji z bazą danych zostanie wykorzystany framework SQLAlchemy wraz ze sterownikiem Psycopg2. Architektura zakłada położenie nacisku na optymalizację czasu przetwarzania zapytań bazodanowych. W tym celu dane miałyby być przetrzymywane w pamięci RAM. Oznacza to, że w przypadku gdy użytkownik zdecyduje się wykorzystać te same parametry, nie będzie wysyłane zapytanie do bazy danych, gdyż informacje zostaną wcześniej zapisane w cache.