Aplikacja webowa do ustalania odpowiednich miejsc

# Cel aplikacji

Aplikacja ma na celu udostępnienie użytkownikowi narzędzia decyzyjnego które na podstawie dostarczonych informacji przez użytkownika dobiera wydarzenie, miejsce lub aktywność które mogłoby zainteresować użytkownika w najbliższym czasie.

## Analiza danych

Decyzja podejmowana jest poprzez przeprowadzenie analizy na zgromadzonych danych. Dane te domyślnie powinny dotyczyć codziennego życia i zainteresowań użytkownika. Na ich podstawie przeprowadzona zostaje analiza kategoryzująca poszczególne informacje do określonych uprzednio kategorii informacji. Aby zapewnić innowacyjność rozwiązania informacje te powinny pochodzić z niekonwencjonalnych źródeł takich jak np. portale społecznościowe.

## Zwracany wynik

Wynikiem działania aplikacji powinny być wydarzenia lub miejsca znajdujące się na terenie miasta Kraków które według wyników aplikacji mogą wpasowywać się w profil osobowy użytkownika utworzony na podstawie analizy danych. Miejsca te powinny uprzednio zostać wprowadzone w bazę danych aplikacji w celu ich odpowiedniej konfiguracji pod model myślowy programu.

## Kategoryzacja danych

Zarówno profil użytkownika oraz informacje o miejscach powinny posiadać swój własny unikalny model. Modele te mają na celu ułatwienie kategoryzacji wyników analizy oraz połączenia ich z najbardziej trafnymi propozycjami z danych aplikacji. Powiązania powinny być na tyle silne, aby każdy wynik analizy użytkownika mógł zostać poprawnie przypisany do któregoś miejsca. Jednocześnie elastyczność i uniwersalność kategorii powinna być na tyle duża, aby każdy profil użytkownika mógł zostać poprawnie utworzony nawet z minimalnej ilości danych.

## Profile osobowe użytkownika

Profil powinien zostać utworzony na podstawie danych dostarczanych przez użytkownika. Decyzja o sposobie pobierania danych nie powinna wpływać na tworzenie modelów. Model powinien kategoryzować użytkownika względem pewnych cech osobowości, które można odpowiednio przyporządkować dowolnej osobie.

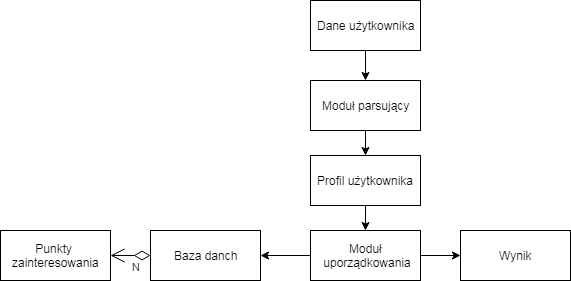
## Punkty zainteresowania

Zwracane jako wynik aplikacji punkty powinny być rozumiane jako miejsca bądź przejściowe lokacje oferujące aktywny sposób spędzania czasu, który spełnia wymagania profilu osobowego użytkownika. Punkty te podlegają kategoryzacji sprzężonej z kategoryzacją profili osobowych użytkowników.

## Diagram przejścia danych

## C:\Users\sg0304270\Downloads\Untitled Diagram (3).png

## Diagram wstępnej architektury



# Opis narzędzi

Aplikacja zostanie przygotowana jako aplikacja webowa udostepniająca klientowi procedury logowania i zarządzania swoim kontem. Od strony użytkownika zostanie przygotowany odpowiedni formularz pozwalający na poprawną konfiguracją konta użytkownika wraz z nadaniem pozwolenia aplikacji na przegląd danych osobowych.

## Język programowania i środowisko programistyczne

Projekt opierać będzie się głównie na języku Java. Jego korporacyjna struktura pozwoli na przygotowanie szkieletu rozległej aplikacji którą można będzie w przyszłości rozwijać lub rozszerzać. Za środowisko programistyczne posłuży produkt JetBrains – InteliJ IDEA. Oprócz standardowego JDK oraz IDE zostanie zastosowane narzędzie automatyzujące budowę Maven lub Gradle.

## Framework

W celu uzyskania webowego charakteru aplikacji projekt korzystać będzie z Frameworka Spring Boot.

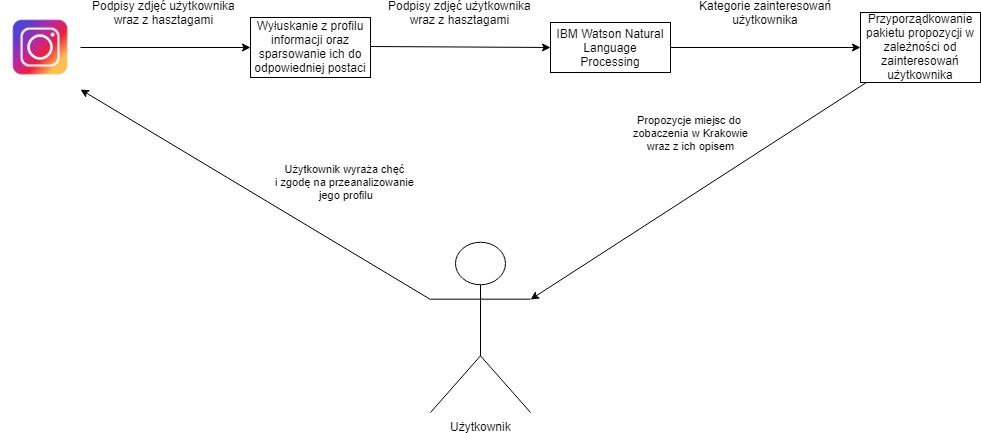
Wykorzystane zostaną odpowiednie funkcjonalności dotyczące Webservices oraz Security. Oprócz tego Spring oferuje bogatą wersję ORM wraz z automatyczną konfiguracją bazy danych. W celu uproszczenia projektu powinny zostać wprowadzone starania uniknięcia konieczności zastosowania osobnego Front-endowego Frameworka.

## Baza danych

Za bazę danych może posłużyć dowolna baza danych SQL. Dla prostoty testowania baza powinna być wbudowana w serwer Tomcat dla zastosowań testowych. Baza będzie zawierać

* Dane konta użytkowników
* Profile osobowe użytkowników
* Punkty zainteresowań
* (Opcjonalne) Historyczne dane wprowadzone przez użytkowników

## Watson

Watson to pakiet narzędzi oferowanych przez IBM, który zawiera między innymi kognitywne usługi serverless. Z pakietu wykorzystane zostanie narzędzie Natural Language Processing, które na podstawie przesłanego tekstu poda informacje o kategoriach zainteresowań użytkownika.

# Twitter

Twitter jest serwisem społecznościowym udostępniającym usługę mikroblogowania. Zarejestrowany użytkownik może wysyłać tweety, czyli krótkie wiadomości tekstowe (do 280 znaków), wyświetlane na profilu użytkownika oraz odczytywać tweety użytkowników go obserwujących. Wiadomości dotyczą spraw zwykle powiązanych z życiem zawodem lub zainteresowaniami użytkownika. Na ich podstawie człowiek może jest określić preferencje obserwowanego użytkownika dotyczących muzyki, architektury, kuchni oraz wielu innych. Zadaniem naszego algorytmu będzie określenie preferencji użytkownika na podstawie jego tweetów oraz wykorzystanie ich do polecenia mu miejsc, które potencjalnie mogą wzbudzić jego zainteresowanie.

## Twitter API

Twitter udostępnia rozbudowane API dla deweloperów, które umożliwia wyszukiwanie tweetów między innymi filtrując po tematyce, użytkowniku, popularności tweetu. Przykładowe zapytanie do API bez uwzględnienia nagłówków autoryzujących.

### <https://api.twitter.com/1.1/search/tweets.json?q=nasa&result_type=popular>

Zwraca ono najpopularniejsze tweety, zawierające frazę nasa. Wybrane możliwe parametry do zapytań Twitter API:

* q – wymagany, kodowany w UTF-8, zakodowane zapytanie URL o maksymalnej długości 500 znaków. Zapytania mogą być dodatkowo ograniczone złożonością.
* result\_type – opcjonalnu, Określa typ wyników wyszukiwania, które chcesz otrzymywać. Bieżące ustawienie domyślne to „mieszane”. Możliwe wartości to:
  + mieszane: uwzględnij w odpowiedzi zarówno wyniki popularne i nowe.
  + ostatnie: zwróć tylko najnowsze wyniki w odpowiedzi.
  + popularne: zwraca tylko najbardziej popularne wyniki w odpowiedzi.
* geocode – opcjonalny, zwraca tweety użytkowników znajdujących się w danym promieniu od punktu o podanej szerokości i długości geograficznej.

## Biblioteki do obsługi Twitter API

Powstało wiele bibliotek ułatwiających obsługę Twitter API, dla Javy są to:

Twitter4J - to nieoficjalna biblioteka Java dla API Twittera, dzięki które można łatwo zintegrować aplikację Java z Twitter API. Zalety Twitter4J:

* W pełni zgodna z Twitter API 1.1
* Działa na dowolnej platformie Java w wersji 5 lub nowszej
* Obsługuje platformę Android i silnik Google App Engine
* Zależność zerowa: nie są wymagane dodatkowe pliki jar
* Wbudowana obsługa OAuth oraz gzip

Wirtualna informacja turystyczna

# Opis aplikacji

Aplikacja ma udostępnić użytkownikowi zautomatyzowaną metodę wyszukiwania interesujących miejsc na obszarze Krakowa które odpowiadają preferencjom użytkownika. Komunikacja z użytkownikiem powinna odbywać się poprzez rozmowę głosową podczas której użytkownikowi zostają zadany zbiór uprzednio przygotowanych pytań, które mają dostarczyć informacji na temat docelowego przedziału miejsc które użytkownik jest skłonny odwiedzić. Następnie za pomocą danych zgromadzonych z profilu społecznościowego użytkownika dotyczących ogólnych zainteresowań, spośród zbioru miejsc zdefiniowanych w aplikacji zostanie wyodrębniony i zaprezentowany użytkownikowi jak najbardziej trafny wynik działania programu.

# Środowisko aplikacyjne

Aplikacja domyślnie powinna być dostępna jako aplikacja webowa umożliwiająca komunikację poprzez przeglądarkę internetową. Podczas tworzenia środowiska powinny zostać uwzględnione czynniki krytyczne dla poprawnego działania aplikacji

* Możliwość szybkiej dwustronnej komunikacji za pomocą mikrofonu użytkownika oraz synchronizowanej mowy ze strony aplikacji
* Wysoka zdolność obliczeniowa dla
  + Przetwarzania mowy na tekst
  + Synchronizacji mowy
  + Przetwarzania języka naturalnego
  + Algorytm wyszukania odpowiednich miejsc
* Odpowiednia przepustowość łącza internetowego

# Środowisko programistyczne

Aplikacja powinna być wdrażana w środowisku pozwalającym na korporacyjną architekturę. Modułowość aplikacji pozwoli na jednoczesny rozwój każdego aspektu aplikacji równocześnie i bezkolizyjnie. Jednocześnie środowisko powinno udostępniać framework który odciąży programistów od zbędnego tworzenia obsługi protokołów komunikacyjnych. Wszystkie powyższe oczekiwania powinny zostać spełnione przy jednoczesnym zachowaniu minimalnej ilości użytych technologii.

# Moduły programu

## Rozpoznawanie i przetwarzanie mowy naturalnej

## Syntezowanie lub odtwarzanie głosu

## Zbieranie informacji z mediów społecznościowych

## Budowanie preferencji

## Wybór najbardziej trafnych miejsc

## Nawigowanie

## Architektura programu

W celu utworzenia rozbudowanej aplikacji potrzebne będzie środowisko o architekturze korporacyjnym w którym moduły aplikacji mogłyby być rozwijane jednocześnie i bezkolizyjnie. Taka architektura gwarantowałaby dalszy rozwój aplikacji przez kolejne zespoły ponieważze względu na jej złożoność jest nieprawdopodobnym aby została ukończona przez jeden zespół programistyczny

# Opis przebiegu implementacji modułów

## Rozpoznawanie i przetwarzanie mowy naturalnej

Aktualnie wszystkie narzędzia rozpoznania mowy naturalnej pozostają pod płatną licencją. Oprócz płatnych webowych wersji rozpoznania mowy pozostają biblioteki open source wymagające od użytkownika odpowiedniego przygotowania modelu algorytmów. Jednym z takich programów jest shpinx4 (<https://github.com/cmusphinx/sphinx4>). Inna możliwością byłoby wykorzystanie środowiska Python bogatego w wiele bibliotek rozpoznawania mowy.

## Syntezowanie lub odtwarzanie głosu

## Zbieranie informacji z mediów społecznościowych

## Budowanie preferencji

Przetwarzania mowy naturalnej jest istotne zarówno dla analizy rozmowy z użytkownikiem jak i analizy profilu społecznościowego.

Dla początkowej wersji przetwarzania mowy zostanie wykorzystany bazowy algorytm „Bag of Words”. Jego działanie powinno być skuteczne dla przypadku profili społecznościowy taki jak Twitter ponieważ złożoność zdaniowa wypowiedzi na nim umieszczonych ogranicza się do krótkich, chwytliwych haseł co jest idealnym przypadkiem dla działania tego algorytmu.

Kolejnym udoskonaleniem przetwarzania mowy powinna być ewolucja „BoW” do modelu n-gram dla małych wartości n.

Tak przygotowane próbki mogą zostać później sklasyfikowane przy pomocy podstawowych metod takich jak drzewa decyzyjne które będą rzutować wyniki na pewne kategorie uprzednio zdefiniowane grupy zainteresowań. Tutaj następuje syn

W celu uniknięcia konieczności ręcznej implementacji tych podstawowych algorytmów zostanie wykorzystana biblioteka obliczeniowa Weka3 (<https://www.cs.waikato.ac.nz/ml/weka/>) udostępniana jako plik JAR ze strony uniwersytetu Waikato.

## Wybór najbardziej trafnych miejsc

## Nawigowanie