Departure Times

**Problem Statement**

Create a service that gives real-time departure time for public transportation. The app should geolocalize the user.

**Solution**

Solution focuses on backend.

1. Geolocalize the user to get the current coordinates (latitude and longitude). User can enter different coordinates if he/she wants to get departure times at some other location. JavaScript validations are provided for the values of latitude and longitude.
2. By using “[Google Places API Web Service](https://developers.google.com/places/webservice)” nearby transit stations are found.
3. From 511 API departure times of these nearby stops are found.
4. Output format: List of these details
   1. Agency Name of corresponding stop name
   2. Stop Name
   3. Route Name of corresponding stop name
   4. Departure times within next 90 minutes
5. Stop Names returned by Google API are different from the stop names of 511 API.

To solve this problem:

* Get all the agencies from 511 API.
* Get all the routes for all these agencies.
* Get all the stops on these routes.
* Compare these stop names with the stop names of nearby stops (Google API).
* To compare these stop names I have used edit distance with the equality factor of 3/4.
* If stop names are equal by equality factor of  at least ¾,  then find the departure time by using stop name of 511 API.

Another possible Approach:

* Instead of calculating the edit distance of the two stop strings, token matching can be done. Again if tokens match by factor ¾ then stop names are similar.

**Shortcomings**

1. By changing the equality factor accuracy can be improved.
2. Some stops names are starting with a common string and then differ only by a number. These stops might be geographically far from each other. These stops which does not match completely are also selected and departure times for these are found. Examples of such stop names: ‘Pier 1’ and ‘Pier 4’,
3. Also some stop names are almost similar Ex. ‘The Embarcadero & Folsom St’ and ‘The Embarcadero and Sansome St’. As these are equal by equality factor of ¾, but might be geographically far from each other are selected.
4. For ‘SF-MUNI’ agency, stops names provided by Google API has ‘&’ while stop names of 511 API use ‘and’ instead of ‘&’. So just by replacing all the ‘&’ with ‘and’ in stop names, departure times can be found. But this only works for ‘SF-MUNI’ agency.
5. Most of the difference with the stop name is because of the appended strings at the end of the stop names. So by using startsWith() function accuracy can be improved. But it still fails for some cases the shortcoming mentioned in the first point. For example ‘Pier 1’ and ‘Pier 19’ would pass this test. For these types of stop names complete string matching can be implemented.
6. To get departure time, for some stops I was getting ‘404-File or Directory not found’ error when used ‘GetNextDeparturesForStopName’ service from 511 API. Ex: http://services.my511.org/Transit2.0/GetNextDeparturesForStopName.aspx?token=123-456- 789&agencyName=BART&stopName=Richmond. Therefore ‘GetNextDeparturesByStopCode’ service needs to be used.
7. As first all the stop names from 511 API are found the execution time is increased.
8. Some preprocessing can be done on the stop names to increase the accuracy of matching. For example in some stop names ‘(SF)’ is appended. This can be removed before matching with the other stop name. By comparing these 511 API stop names with Google API stop names more dissimilarities can be found out, which can be preprocessed.

**Future enhancements**

* More details regarding Agency, Route and Stop can be retrieved.