

# EAT – RESTAURANT/TAXI RECOMMENDATION SYSTEM

Adithya Ramakrishnan  
Shiyamsundar Soundara Rajan  
Vinoth Selvaraju

## 1. Introduction

The restaurant recommendation system focuses on providing the user with the basic information about the restaurant such as cuisine, rating, location, reviews, cost preference etc. The Uber and Google Maps API are integrated with the system to provide the user with information regarding transportation such as time taken, cost and vehicle capacity.

### 1.1.Context and Motivation

Today's restaurant recommendation systems provide the user with necessary information about the restaurant, but there is no system which integrates a taxi service with the system to make it easier for the customer. This system gives the user (especially tourists and people who do not have their own vehicle) a single point of access to everything they would need for the best dining experience.

### 1.2.Problems to be Addressed

With the advancement of computer and software technologies, it's quite easy to find out choices for a dinner. However, there's no single system where we can get all information required for the whole event. We use Yelp/Google to find list of restaurant nearby or to read review about the restaurant, and use different system such as Google Maps/Uber to look for directions and transportation services. We are trying to develop an innovative system where user can look up in one single system to get all detail to plan a dinner/lunch.

### 1.3.Previous related work

**1.3.1** A Chinese Restaurant Recommendation System Based on Mobile Context-Aware Services, 2013  
IEEE 14<sup>th</sup> International conference on Mobile Data Management.

Author: Chung-Hua Chu, Se-Hsien Wu, Year: 2013

<http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=6569074>

**1.3.2** Restaurant Recommendation for Facebook Users, Stanford University

Author: Qiaosha Han, Vivian Lin, Wenqing Dai, Year: 2012

<http://cs229.stanford.edu/proj2012/HanLinDai-RestaurantRecommendationForFacebookUsers.pdf>

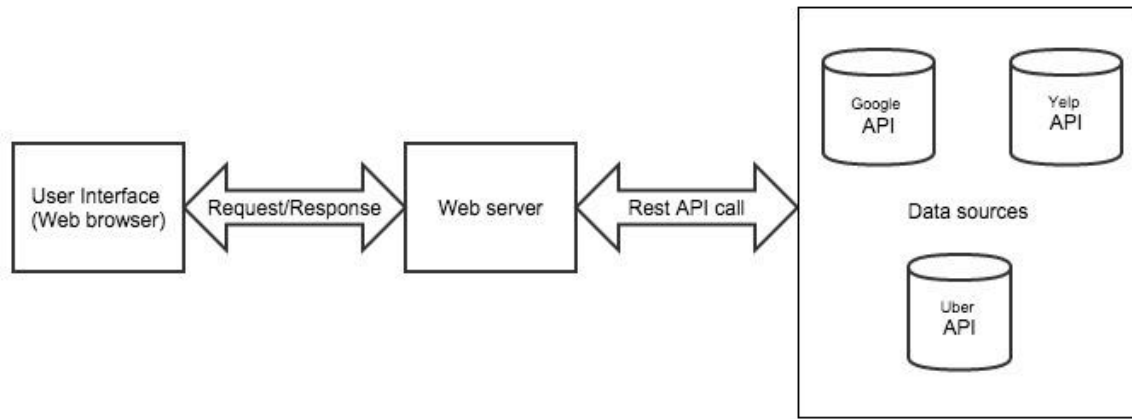
## 2. Technical Contribution

### 2.1.Basic Ideas

The basic idea of the System is provide restaurant and taxi/public transport recommendations based on user's query and be able to dynamically modify the results based on filter criteria's. The system is a web application that takes input from User in the form of free text and returns a list of restaurants based on the Yelp and Google data. The application ranks the output based on a custom ranking algorithm by assigning weights on different data. Additionally, the results are shown on a map with pointers based on geometric co-

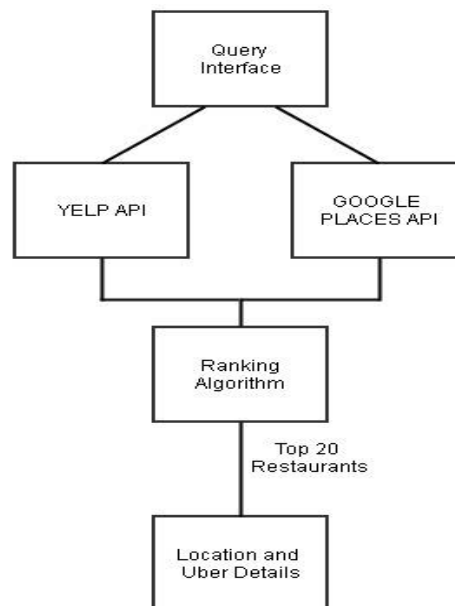
ordinates. The application provides abundant amount useful details in single system such as address, contact phone number, cuisine, rating, review count, deals, pictures, geometric co-ordinates, list of available Uber taxis, expected time to get the taxi, price estimate for the entire travel, directions, public transit etc. for selected query result.

The general idea has been plot as a block diagram as shown below,



## 2.2.Methodology

User query is passed to Yelp & Google Places APIs to retrieve a list of restaurants. The results from both are then combined and ranked, with weights assigned to rating and review count. The search results of the query is embed in a Map using Google Maps API. Once the restaurant is selected, Google Maps API is used to visually represent locations of the selected restaurant and user along with public transit details such as directions, schedule and time estimates. Finally the Uber API is used to get the list of all available Uber taxis based on filters such as, capacity, cost, etc.



## 2.3.Approach

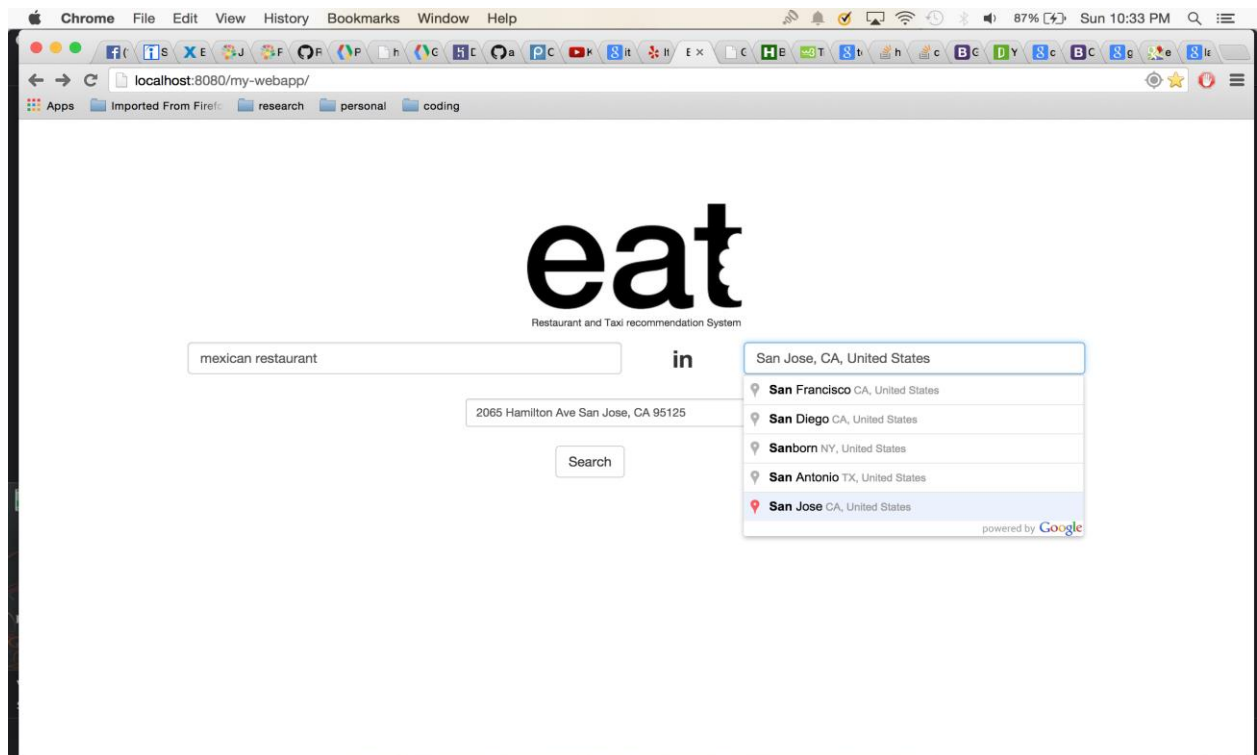
The system is based on the integration of multiple APIs. The necessary information from each service is retrieved and combined. The result is then projected to the user in a sequence which makes it easier and more comfortable to view and use. The user is provided with a set of filters (price range, rating, distance etc.) to help order the results in the way he/she wants such as

*Rating* - The ratings for each restaurant is calculated based on the ratings gathered from yelp and google places

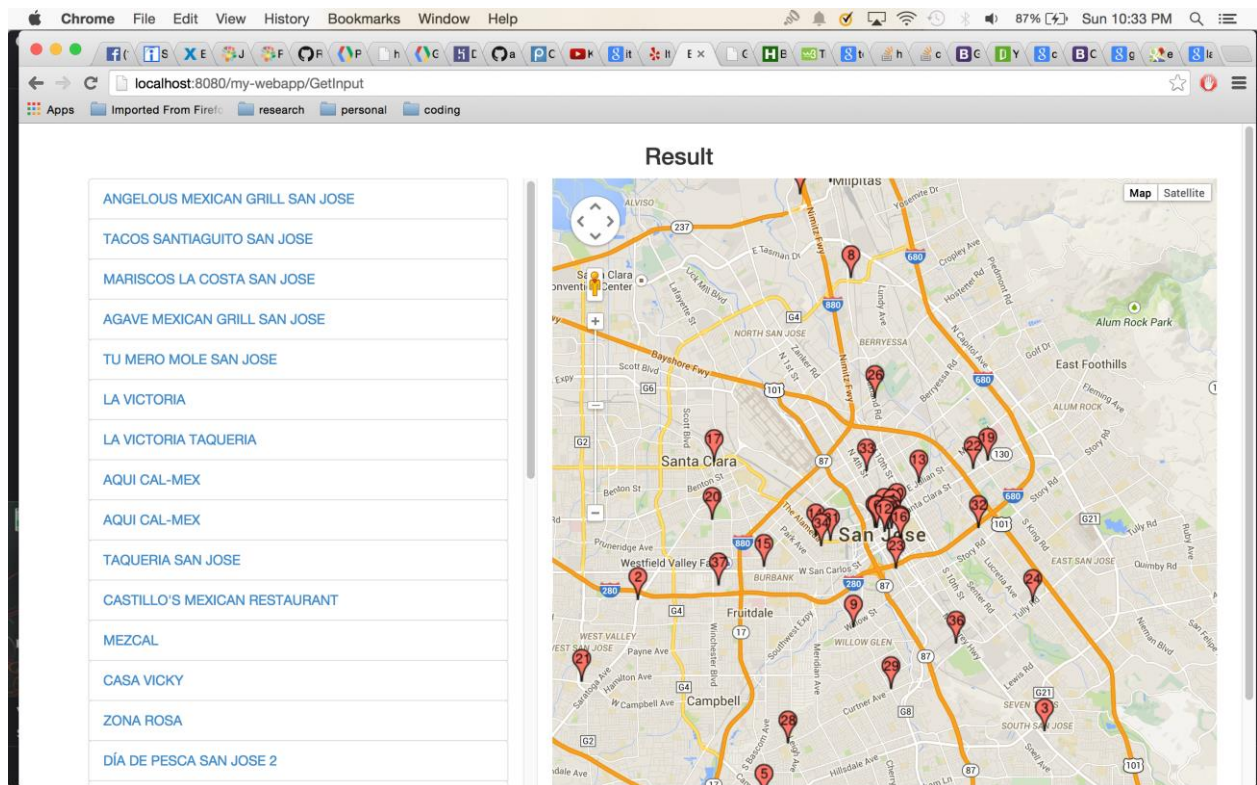
*Distance* - The Maps API is used to give the user information about the location of the restaurant and the time taken to reach it.

## 2.4.User Interface

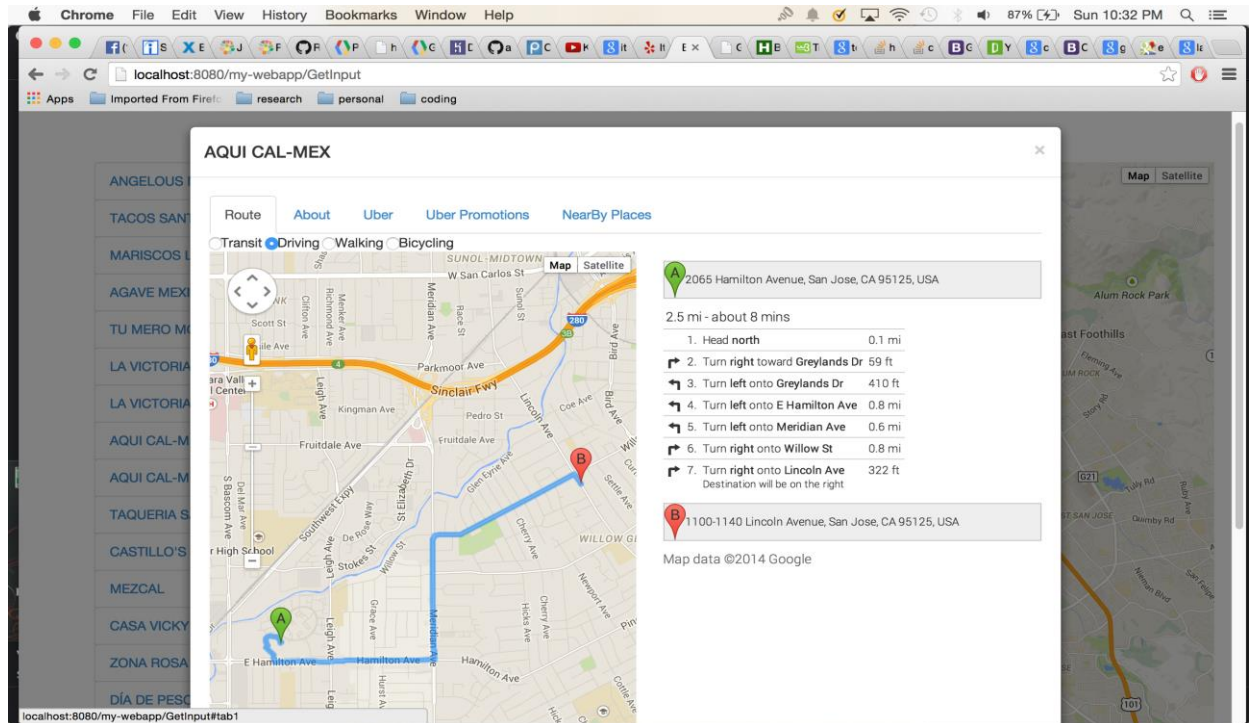
### Home Page:



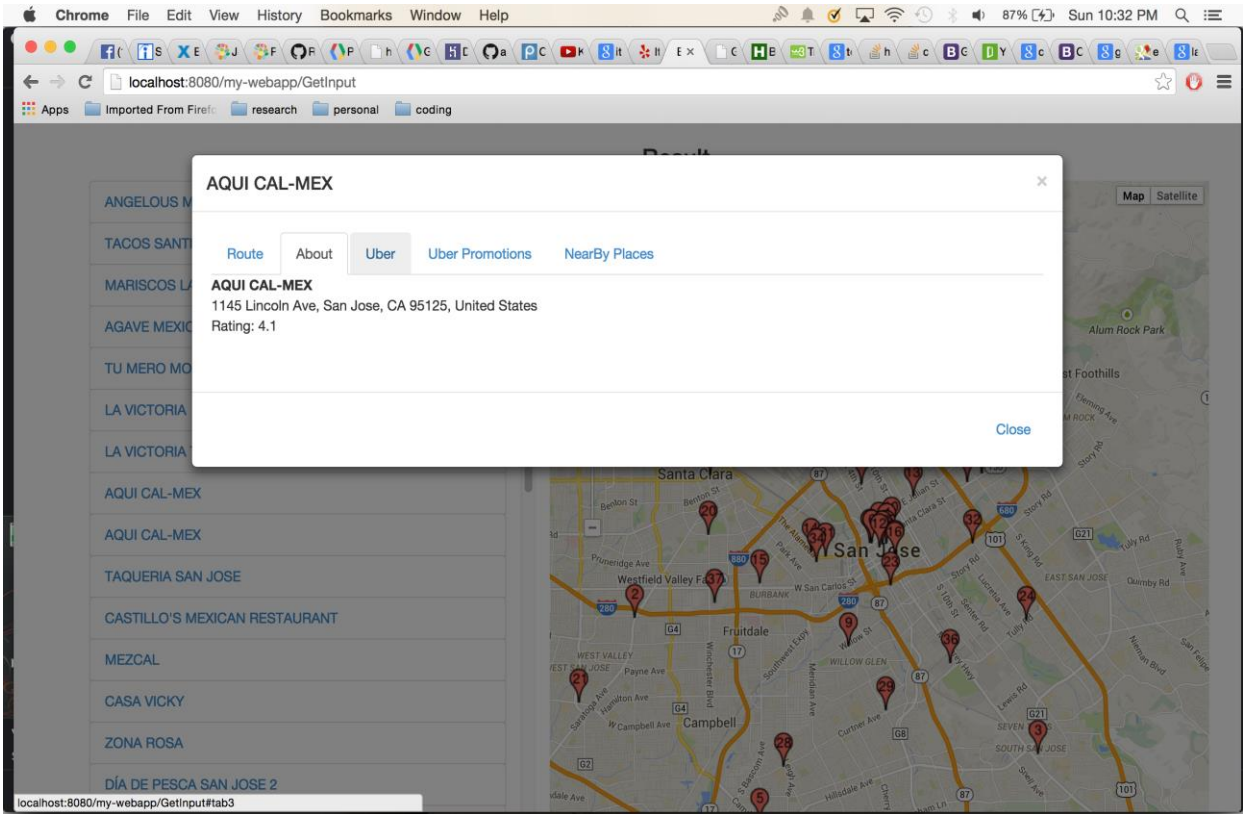
## Result:



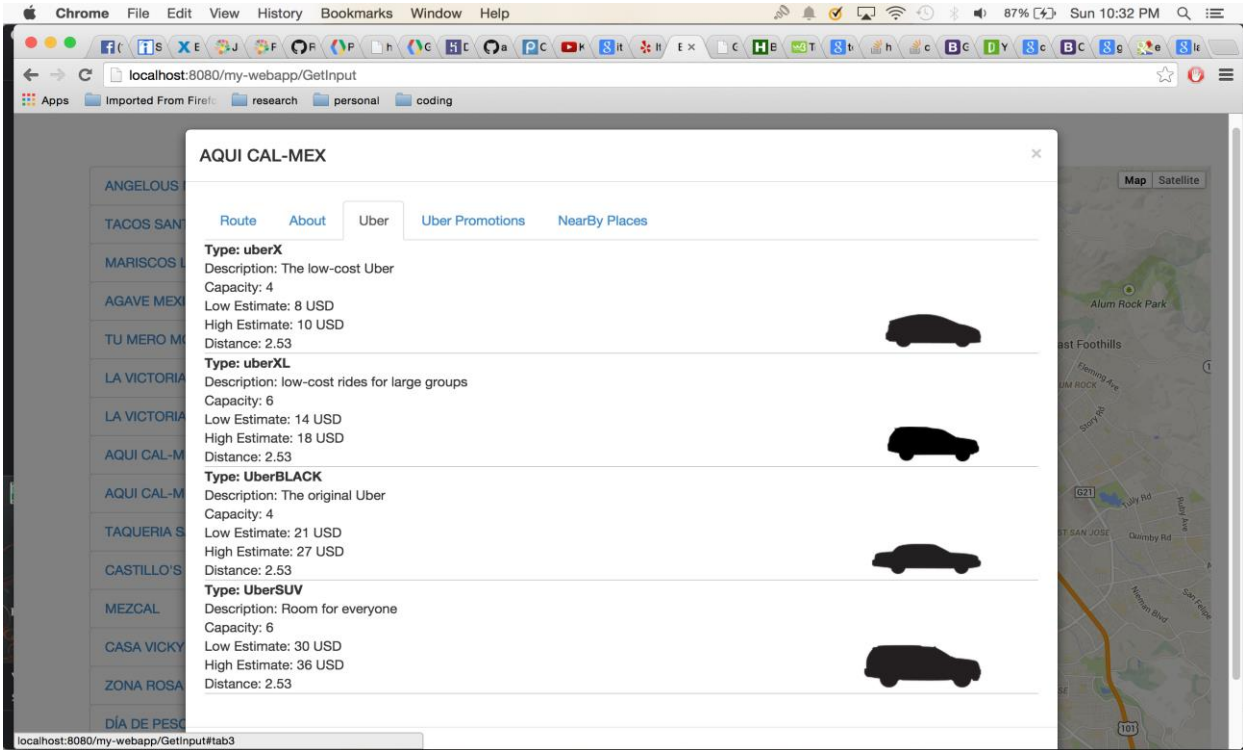
## Route:



About:

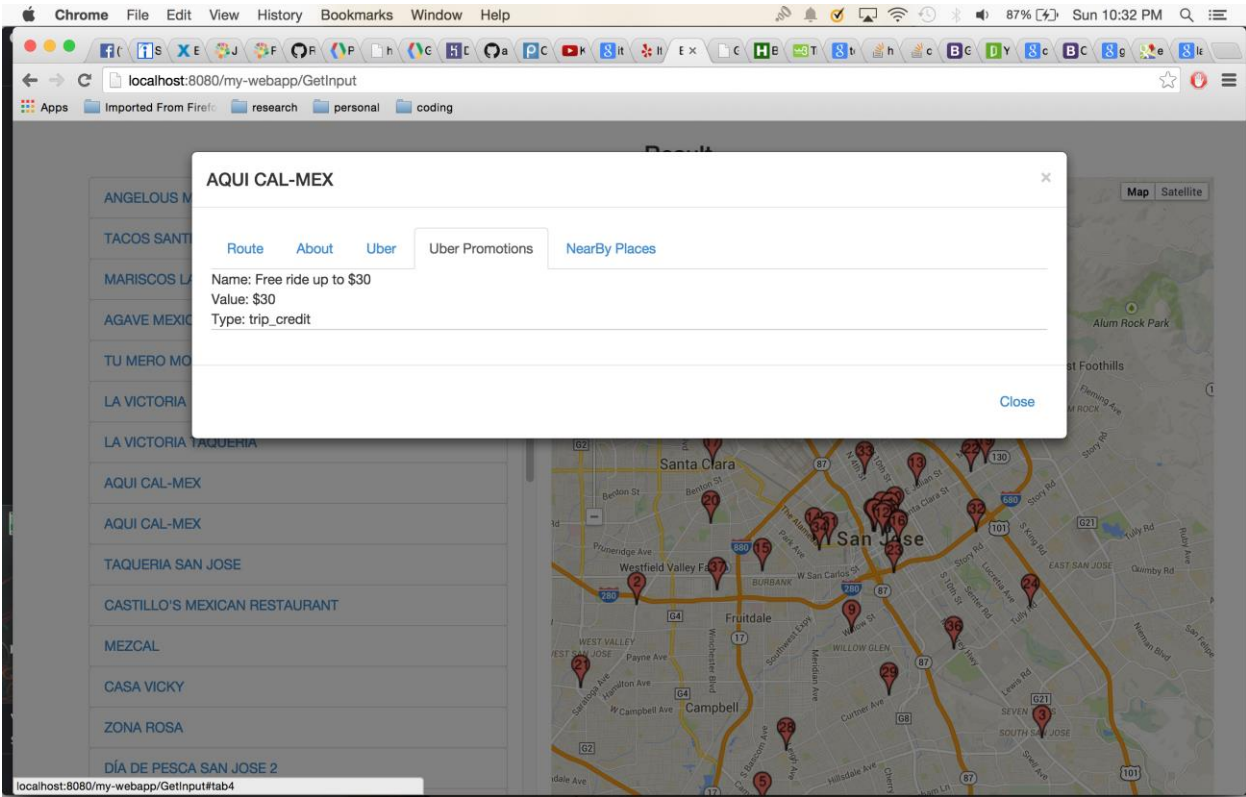


Uber:

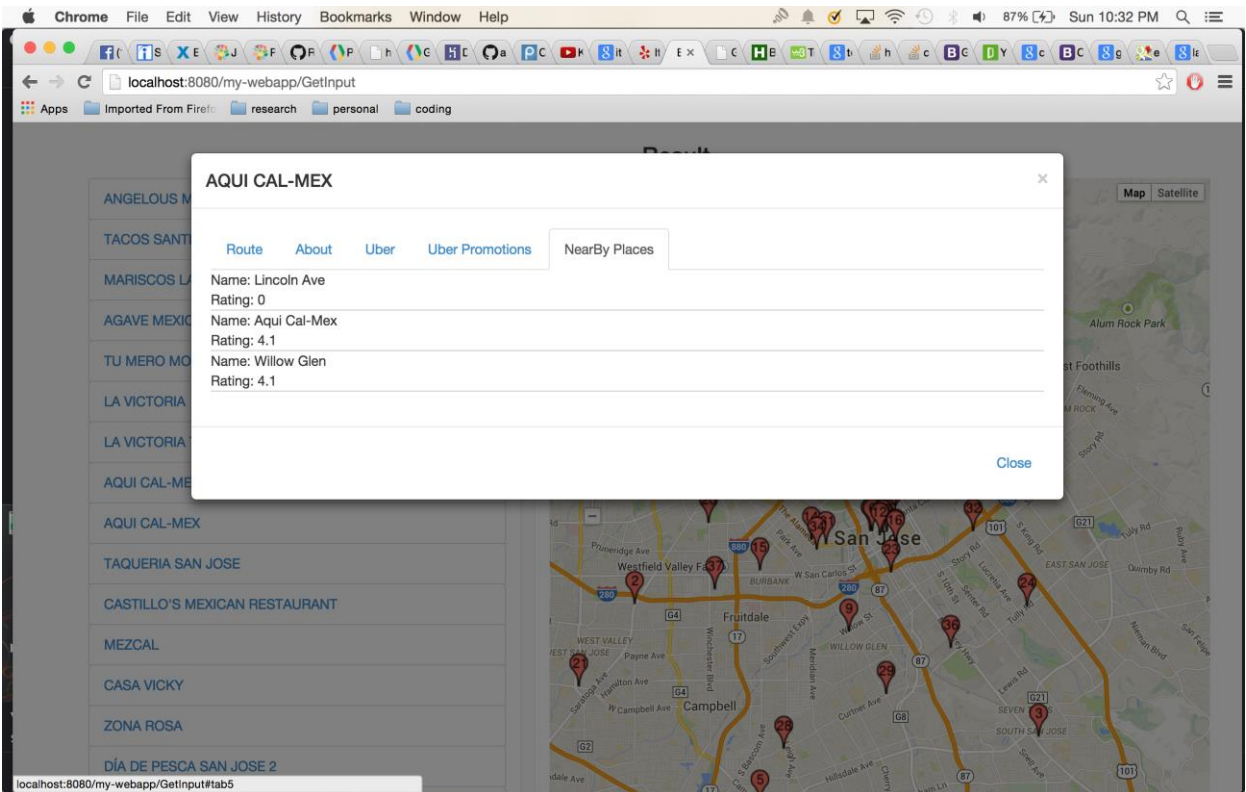




## Uber Promotions:



## NearBy Places:



## 2.5. Justification of Design Choices

### JSON Object:

- It is used as JSON is a more compact form as compared to XML, thereby proving to be better choice when latency is concerned.
- It is also an easier option when the data needs to be pushed using javascript.

### Bootstrap:

- It is extremely flexible, and making changes or adding features to existing web sites is very easy.
- Highly customizable which helps in tailoring the interface based on our needs.

### Test Driven Development:

- Helps every standalone piece of logic to be tested separately thereby allowing us to make changes to the application while making sure that each module is still working correctly.
- Helps in pin pointing the reason behind the issue. Thereby saving development time.

### MVC Architecture:

- Allows us to reuse the business logic over different applications.
- Changes to the User Interface does not affect the existing codebase.

## 3. Experimental Evaluation

### 3.1. Data Sources

The system utilizes data from multiple data sources through Web services. The following Web services API are used in the project,

#### 3.1.1. Uber API

- Populate list of available Uber taxis at a particular location based on geometric co-ordinates
- Estimate on the time expected to get the taxi at the start point
- Price estimate for the entire travel based on the geometric co-ordinates of start and end location

#### 3.1.2. Yelp API

- Retrieve restaurant list based on user's query. Ex: "Restaurants in buffalo"
- Populate more details on each result such as address/contact/cuisine/rating/review count/deals/geometric co-ordinates etc.
- Sort results based on ratings/distance

#### 3.1.3. Google Maps API

- Populate dynamic map with pointers for search results
- Retrieve public transit info such as directions, public transit details (next and future), time estimate etc. based on start and stop locations

#### 3.1.4. Google Places API

- Retrieve details such as reviews, ratings, photo etc. for query result
- Near-by suggestions based on the selected query result

Detailed information on the API's that will be used in the system is collected and described as a matrix below,

API Name/Description	API functionality	Input	Output
<b>Uber</b> Web service that returns list of Uber taxis available/capacity/price & time estimates etc. based on the geomeric co-ordinates	Taxis available based on the location attributes GET /v1/products	Latitude Longitude	json  product_id description display_name capacity image
	Approx. price estimate for the travel GET /v1/estimates/price	start_latitude start_longitude end_latitude end_longitude	json  product_id currency_code display_name estimate low_estimate high_estimate surge_multiplier
	Approx. time estimate for getting taxi GET /v1/estimates/time	start_latitude start_longitude customer_uuid (optional) product_id (optional)	json  product_id display_name estimate
<b>YELP</b> Web service that retrieves restaurants with details such as deals/offers, ratings, reviews based on search query, sort results by ratings/distance	Restaurant based on user's query GET /v2/search	search term limit offset sort categories filter radius filter	json region latitude longitude rating review count phone distance address deals gift certificates
<b>Google Maps - Embed API</b> Google Maps Embed API uses a simple HTTP request to return a dynamic, interactive map	Directions mode - Direction Map (distance/travel time) based on the source and destination  https://www.google.com/maps/embed/v1/directions?key=API_KEY&origin=Oslo+Norway&destination=Telemark+Norway&avoid=tolls highways	origin destination  Optional: waypoints mode bicycling avoid units	Directly embed map that can be used within iframe tag  Ex: <iframe width="450" height="250" frameborder="0" style="border:0" src="https://www.google.com/maps/embed/v1/search?key=API_KEY&q=record+stores+in+Seattle"> </iframe>
	Public transit information from start to end locations  https://maps.googleapis.com/maps/api/directions/json?origin=Boston,MA&destination=Concord,MA&waypoints=Charlestown,MA L	origin destination  Optional: mode (driving/walking/bicycling/transit) waypoints	json



	exington,MA&key=API_KEY	alternatives avoid language units region departure_time (Either departure_time or arrival_time is mandatory while choosing mode as transit) arrival_time	
<b>Google Maps - Javascript API v3</b>	Javascript API functions to populate map with multiple markers based on the search result		
<b>GooglePlaces</b> The Google Places API is a service that returns information about Places — defined within this API as establishments, geographic locations, or prominent points of interest — using HTTP requests. Place requests specify locations as latitude/longitude coordinates.	Finding nearby attractions  <a href="https://maps.googleapis.com/maps/api/place/nearbysearch/json?location=-33.8670522,151.1957362&amp;radius=500&amp;types=food&amp;name=cruise&amp;key=AddYourOwnKeyHere">https://maps.googleapis.com/maps/api/place/nearbysearch/json?location=-33.8670522,151.1957362&amp;radius=500&amp;types=food&amp;name=cruise&amp;key=AddYourOwnKeyHere</a>	key location radius rankby  Optional: keyword language minprice and maxprice name opennow rankby (prominence/distance) types pagetoken zagsatselected	json/xml
	Google place search  <a href="https://maps.googleapis.com/maps/api/place/textsearch/xml?query=restaurants+in+Sydney&amp;key=AddYourOwnKeyHere">https://maps.googleapis.com/maps/api/place/textsearch/xml?query=restaurants+in+Sydney&amp;key=AddYourOwnKeyHere</a>	query key  Optional: location radius language minprice and maxprice opennow types zagsatselected	json/xml
	Place details such as reviews/ratings etc.  <a href="https://maps.googleapis.com/maps/api/place/details/json?placeid=ChIJN1t_tDeuEmsRUsoyG83frY4&amp;key=AddYourOwnKeyHere">https://maps.googleapis.com/maps/api/place/details/json?placeid=ChIJN1t_tDeuEmsRUsoyG83frY4&amp;key=AddYourOwnKeyHere</a>	key placeid or reference  Optional: extensions language	json/xml
	Place photo  <a href="https://maps.googleapis.com/maps/api/place/photo?maxwidth=400&amp;photoreference=CoQBegAAAFg5U0y">https://maps.googleapis.com/maps/api/place/photo?maxwidth=400&amp;photoreference=CoQBegAAAFg5U0y</a>	key photoreference maxheight or maxwidth	json

	- iQEtuVMfqw4KpXYe60 QwJC- wl59NZlcaxSQZNgAhGrij mUKD2NkXatfQF1QRap- PQCx3kMfsKQCcxkZqQ &key=AddYourOwnKeyH ere		
--	--	--	--

### 3.2. Hardware Interfaces & Software Interfaces

The system is intended as a web application and hence is compatible to any browser such as Internet Explorer, Mozilla, Chrome or Netscape Navigator by which user can access the system. Since the application must run over the internet, all the hardware shall require, connection to internet which comprises the hardware interface for the system. As for e.g. Modem, WAN – LAN, Ethernet Cross-Cable.

The business logic for the application will be developed using the Java JDK (Java Development Kit) and the following interfaces are established to retrieve the data from multiple sources,

- The system shall communicate with Uber API to get the list of available Uber taxis at a particular location based on geometric co-ordinates
- The system shall communicate with Yelp API to retrieve restaurant list based on user's query. Ex: "Restaurants in buffalo".
- The system shall communicate with Google Maps API to provide directions from start to end locations for each query result.
- The system shall communicate with Google Places API to get nearby suggestions based on the selected query result

### 3.3. Results

The Web application retrieves the list of restaurant from Google Places and Yelp, ranks according to custom ranking algorithm and populates the result on the front-end. The below results are obtained while trying to find restaurant/taxi service for free text query,

#### Query:

Mexican Restaurant in Buffalo, NY, US

#### Result:

1. ZIGGY'S TACO & SUB
2. ADOBE GRILL BUFFALO
3. EL RANCHITO CLARENCE
4. LONE STAR GRILLE
5. CHIPOTLE MEXICAN GRILL
6. EL AGAVE CHEEKTOWAGA
7. DON TEQUILA MEXICAN RESTAURANT BUFFALO
8. LLOYD TACO TRUCK BUFFALO 2
9. MIGHTY TACO
10. VALLE OF MEXICO BUFFALO
11. WOODYS BEACH CLUB AND TAQUERIA BLASDELL
12. CANTINA LOCO
13. CHILI'S
14. CHIPOTLE MEXICAN GRILL

15. MIGHTY TACO
16. MOE'S SOUTHWEST GRILL
17. LA TOLTECA
18. EL C ANELO
19. SALSARITA'S FRESH CANTINA
20. SALSARITAS BUFFALO
21. BURRITO BAY GETZVILLE
22. AGAVE BUFFALO
23. EL PALENQUE MEXICAN GRILL
24. ELMWOOD TACO & SUBS
25. TACO BELL
26. COYOTE CAFE HAMBURG
27. ROCKO'S TACOS
28. GRAMMA MORA'S MEXICAN RESTAURANT
29. ULTIMA TACO
30. TIJUANA TACO INC
31. AMIGOS RESTAURANT

### **3.4.Interpretation of The Results**

The application returns custom ranked list of restaurant along with a visual representation of location on a Map for the user-entered query.

When a user selects a particular result, the application populates,

- About: Details about the restaurant such as address, rating
- Route: Transit (Public transit), Driving, Walking, Bicycling directions between the user's location and the selected restaurant
- Uber Service: Uber taxi along with seating capacity, time and price estimates
- Uber Promotions: Uber promotions available at the User's location (New Uber User)
- Near by Places: Near by attractions available near the selected restaurant

## **4. Conclusion**

### **4.1.Summary**

We have created a novel web service for providing a comprehensive dinning service for the user. The system integrates data from multiple sources such as Google & Yelp to provide custom ranked list of restaurants matching the user query. Each selected restaurant is integrated with Uber & Google to display Uber taxi details and directions including public transit, walking, driving on Map.

This project provided an opportunity to work with multiple data sources and apply Data Integration concepts studied in CSE636 course. Primarily, we integrated JSON objects with different schema from Google, Yelp & Uber into a standard target schema on the fly and displayed on the User Interface.

### **4.2.Limitations**

- The list of restaurants returned by the search is limited.
- There is no way to directly book a taxi through the site.

### 4.3.Future Work

- The application can be extended to Mobile platform.
- Integrate more data sources to provide a more accurate and comprehensive list of restaurant.
- Include Hoppit API to enable the user to search for restaurants based on his/her mood.

## 5. Bibliography

- Yelp - <http://www.yelp.com/developers/documentation>
- Maps - <https://developers.google.com/maps/documentation/embed/guide>
- Uber - <https://developer.uber.com/v1/endpoints/>
- Google places - <https://developers.google.com/places/documentation/>