**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. **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. **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. **Previous related work**

**1.3.1** A Chinese Restaurant Recommendation System Based on Mobile Context-Aware Services, 2013 IEEE 14th 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

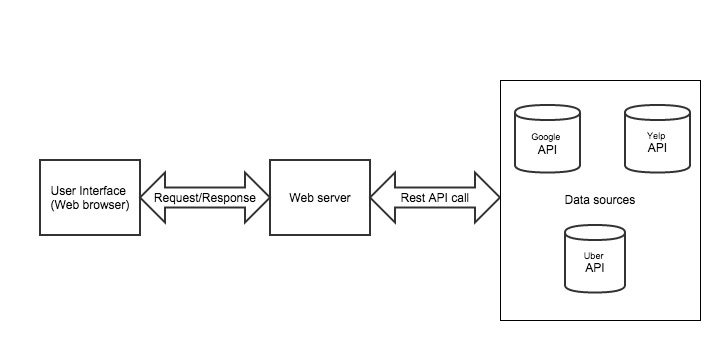
Author: Qiaosha Han, Vivian Lin, Wenqing Dai, Year: 2012

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

1. **Technical Contribution**
   1. **Basic Ideas**

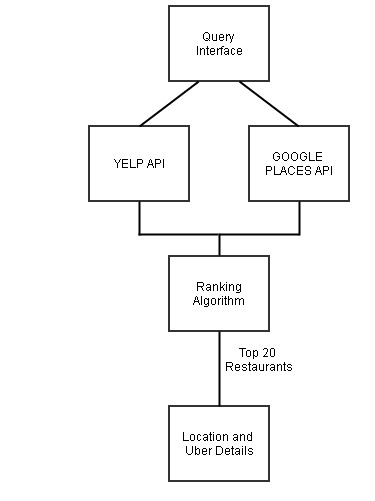
The basic idea of the System is provide restaurant and taxi/public transport recommendations based on user's query and be able to dynamically modify the results based on filter criteria’s. The system is a web application that takes input from User in the form of free text and returns a list of restaurants 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,



* 1. **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.



* 1. **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.

1. **Experimental Evaluation**
   1. **Data Sources**

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

* + 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 base on the geometric co-ordinates of start and end location
    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. 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
    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** |
| **Uber** 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 | Lattitude 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 |
| **YELP** 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 |
| **Google Maps - Embed API** 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|Lexington,MA&key=API\_KEY | origin destination  Optional: mode (driving/walking/bicycling/transit) waypoints alternatives avoid language units region departure\_time (Either departure\_time or arrival\_time is mandatory while choosing mode as transit) arrival\_time | json |
| **Google Maps - Javascript API v3** | Javascript API functions to populate map with multiple markers based on the search result | | |
| **GooglePlaces** 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  https://maps.googleapis.com/maps/api/place/nearbysearch/json?location=-33.8670522,151.1957362&radius=500&types=food&name=cruise&key=AddYourOwnKeyHere | key location radius rankby  Optional: keyword language minprice and maxprice name opennow rankby (prominence/distance) types pagetoken zagatselected | json/xml |
| Google place search  https://maps.googleapis.com/maps/api/place/textsearch/xml?query=restaurants+in+Sydney&key=AddYourOwnKeyHere | query key  Optional: location radius language minprice and maxprice opennow types zagatselected | json/xml |
| Place details such as reviews/ratings etc.  https://maps.googleapis.com/maps/api/place/details/json?placeid=ChIJN1t\_tDeuEmsRUsoyG83frY4&key=AddYourOwnKeyHere | key placeid or reference  Optional: extensions language | json/xml |
| Place photo  https://maps.googleapis.com/maps/api/place/photo?maxwidth=400&photoreference=CoQBegAAAFg5U0y-iQEtUVMfqw4KpXYe60QwJC-wl59NZlcaxSQZNgAhGrjmUKD2NkXatfQF1QRap-PQCx3kMfsKQCcxtkZqQ&key=AddYourOwnKeyHere | key photoreference maxheight or maxwidth | json |

* 1. **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

https://ssl.gstatic.com/ui/v1/icons/mail/images/cleardot.gif

1. **Conclusion**
   1. **Summary**
   2. **Limitations**
   3. **Future Work**
2. **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/>