CS5560: Knowledge Discovery and Management

First project report

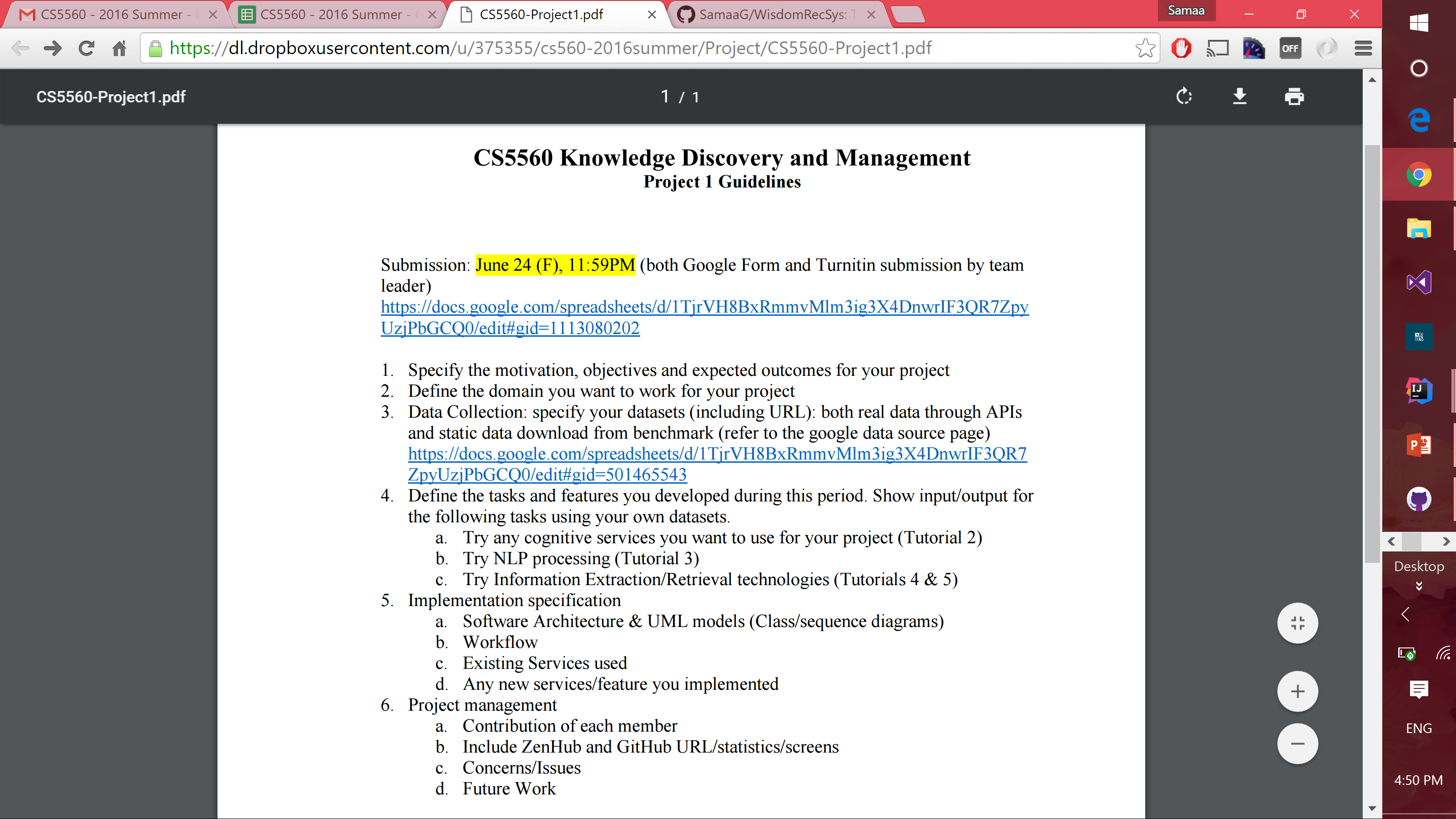
Restaurant recommendation system

Team#5: Wisdom

**Team members**

Samaa Gazzaz (9), Pooja Shekhar (38), Chen Wang (44), Dayu Wang (45)

**This page is just for reference and will be deleted before submission.**



# Motivation

The difficulty of a recommendation system in artificial intelligence is the observation that even though the behavior of the user has been regarded as the most crucial clue and requirement to find out the best-fitting results, the actual results recommended may still not be satisfactory to the user, and sometimes the user itself does have a vague picture of what he/she really wants the results to be. Therefore, recommendation system is actually a complex human activity imitating system, including natural language processing, human interpretation (e.g. what females would probably like, where an artist would like to go to, etc.), and big data treatment.

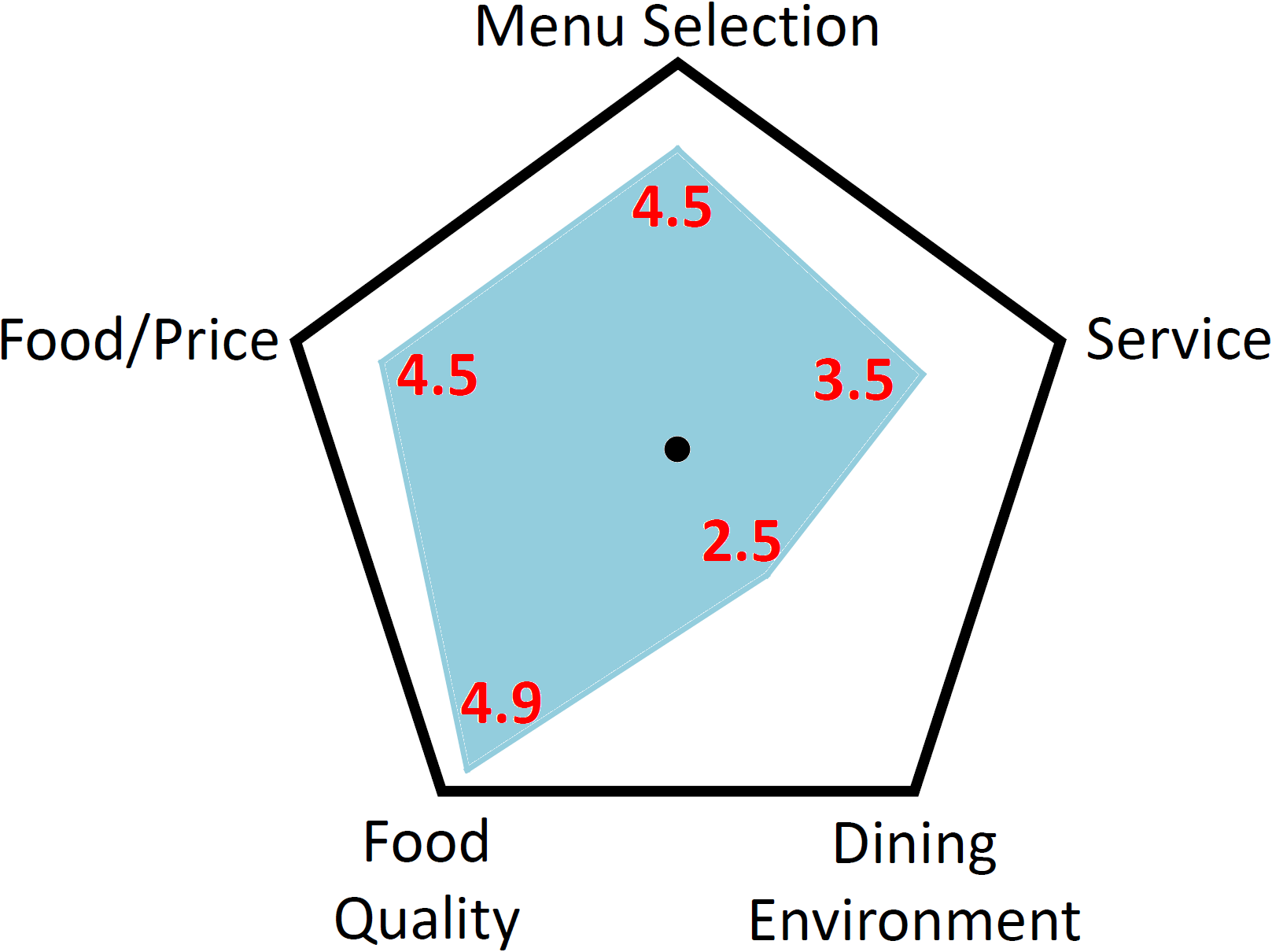
In this class, we decided to build up a small recommendation system from scratch, with the application of those existing services that smartly parse natural language input, and the existing database of targeted items that can be used for us to understand and analyze.

Finding a good restaurant to try new cuisines or even for a fun night out can be a really hard decision to make, especially with all the new restaurants competing to acquire new customers. Using restaurant recommendation apps or systems is very useful, especially when you are starving and would like to taste some local food in a top local restaurant in a city that you had never been there before. Some important features to current restaurant recommendation systems, however, are still missing. For example, when you are using most of the existing restaurant recommendation systems, recommended restaurants almost never change when searching from the same location─usually your home. This can be a problem inasmuch as the same user would want to find new restaurants when searching for a restaurant in two consecutive days. In addition, users usually have to read a lot of reviews in order to decide whether they found a restaurant that meets their needs, because in most cases, just looking at the star rating of a restaurant is an insufficient evidence to believe that the restaurant is on the top level.

Based on such observations mentioned above, we would like to build up a novel restaurant recommendation system that includes the following new features which can distinguish our system from the current existing restaurant recommendation systems.

First, natural language processing is smartly applied in our system. Let’s take the Yelp system as an example. In Yelp, all users are encouraged to input a star-rate (1 to be the worst and 5 to be the best with 0.5 increment), and a text review with photographs. The problem is that it is not very scientific if the star-rating is used as the description of the level of the restaurant in all facets. Probably a user gives a 5 to a restaurant because the food was outstanding, but if another user really cares about the service quality of the restaurant, then the 5-star is misleading and meaningless to the second user. Therefore, all people have reached a consensus that customers’ reviews are very significant. In our system, we use natural language processing approaches to correctly understand customers’ review and put the results into the overall rating of the restaurant. If most customers said that the service is not as splendid as the food in a restaurant, then such information will be summarized and negative evaluation will be applied to the “service” part of the overall restaurant recommendation score.

Second, we are inspired from the way that how social media describes a soccer player, they draw a pentagon and use each corner to represent the ability of the player in a specific facet, e.g. speed, attack, defense, shooting, and stability. Similarly, we can use this very awesome method to present our recommendation result (See Figure 1). This not only summarizes and visualizes the tremendous amount of big data, but also provides a friendly way to let the user know about the restaurant without looking at tons of tedious previous customers’ reviews.



**Figure 1**. Proposed pentagonal representation of a recommended restaurant. The blue-shaded area graphically describes the restaurant from all selected facets.

Third, behavior history of the user is important to our system, primarily because restaurant is special that no one would like to eat in the same restaurant every day. Therefore, not only do we keep the behavior history of a user with one week, but also we take comparison between the systematic recommendation list and the user’s history, in order to decide whether recommendation index (which restaurant comes out first) should be altered.

Finally, based on the time limit of the entire project, we have decided to generate unidirectional server-client architecture for our system, in which two recommendation engines are built in the server and client sides. The engine in the server is the data training engine, which keeps the data being real-time once it is turned on. Also, it trains the formatted data into several relating indicators that represents the recommendation list of the restaurants. The server in the client part is the similarity analyzing engine. It compares the information sent from the server with the user’s requirements of the recommendation, and with the user’s history of behavior. The engine thus rearranges the weighing score of those indicators and generates the final recommendation list of restaurants.

Based on the distinguishable designation mentioned above, we are quite confident about our restaurant recommendation system to be successful and practical, and useful. More details will be given in the following sections.

# Objective

There are multiple necessary features that are still very much needed in currently existing restaurant recommendation systems. The objective of creating our own restaurant recommendation system is to try and fill the gap by providing those missing features. This system should be able to recommend new restaurants to users that match their needs even if users ask for recommendation from the same location and preferences as a previous search; recommendations should be different. In addition, this system will categories and rank different features of restaurant using sentiment analysis of user reviews. This way, new users don’t have to read all the reviews. Instead, they can just check a visual summary of the features and their ranking for each restaurant.

Our main objective in this class project is to learn, study, and understand the basic concepts of natural language processing, as well as all the implementation details of the usage of Apache Spark NLP APIs. We are not setting our goal to beat the current existing restaurant recommendation system (e.g. Yelp), since those mature system is build up by a large group of people and by a long-term research focused on this area. Instead, we would like people to distinguish our recommendation system by the novel features we are developing in our system. First, there is not any restaurant recommendation system can be easily found through the internet which applies Spark NLP. This could be a very proceeding research topic in the area of application of natural language processing. Second, it is very novel to use the “polygonal representation” to describe restaurant and present it to the user (see Figure 1). It is definitely clearer, faster, more readable, and more understandable to a user by reading pictures, instead of looking at tedious, bazaar text. Finally, server-client programming is another important thing that all of us would like to study in this project. Since restaurant recommendation is a service, so it is inevitable to apply server-client method of programming.

# Expected Outcome

When we finish developing this system, it should be able to provide restaurant recommendation to users. Users should be able to specify different kind of preferences when asking for a recommendation such as location, type, closing time…etc. Recommended systems should be updated if the user asks for recommendation two consecutive days; not showing the same recommendations. In addition, ranking of different ranking of each recommended restaurant should be provided depending on analyzing the customer reviews. For example, in addition to five-star ratings, a restaurant should also have a ranking of cleanliness, noise level, friendliness and other features users might have mentioned in the reviews.

# Project Domain

This project is going to be a recommendation system. Namely, it will be specialized in recommending restaurants to users. In addition, the specific domain is involved in restaurants and food chains. Not only will it provide the recommendation, our system will make it easy for users to decide where to eat by summarizing existing reviews by categorizing and raking different features mentioned in the reviews.

# Data Collection

## Dataset Used:

For this project, we will be using Yelp restaurant information. Yelp is a hybrid app that provides users with the ability to search for restaurant by specifying their preferences. Then, Yelp provides a list of restaurants which could be reorganized and sorted by rating, nearest location or other preferences.

## Collection Process:

Yelp provides developers with valuable resources and accessibility to static and dynamic datasets. For the static dataset, we will be using the challenge dataset provided by Yelp for developers that want to use their information in research and join a challenge at the same time. That data set is 2.2GB in size and includes 2.2M reviews and 591K tips by 552K users for 77K businesses. It can be accessed and downloaded through the following URL: <https://www.yelp.com/dataset_challenge>

Moreover, Yelp also provides a well-documented API for developers to access the real-time data featured on Yelp. This API provides access to search over 50 million local businesses from 32 countries. The API URL is: <https://www.yelp.com/developers/>

# Tasks and Features

# Input/output

## Cognitive Services:

## NLP Processing:

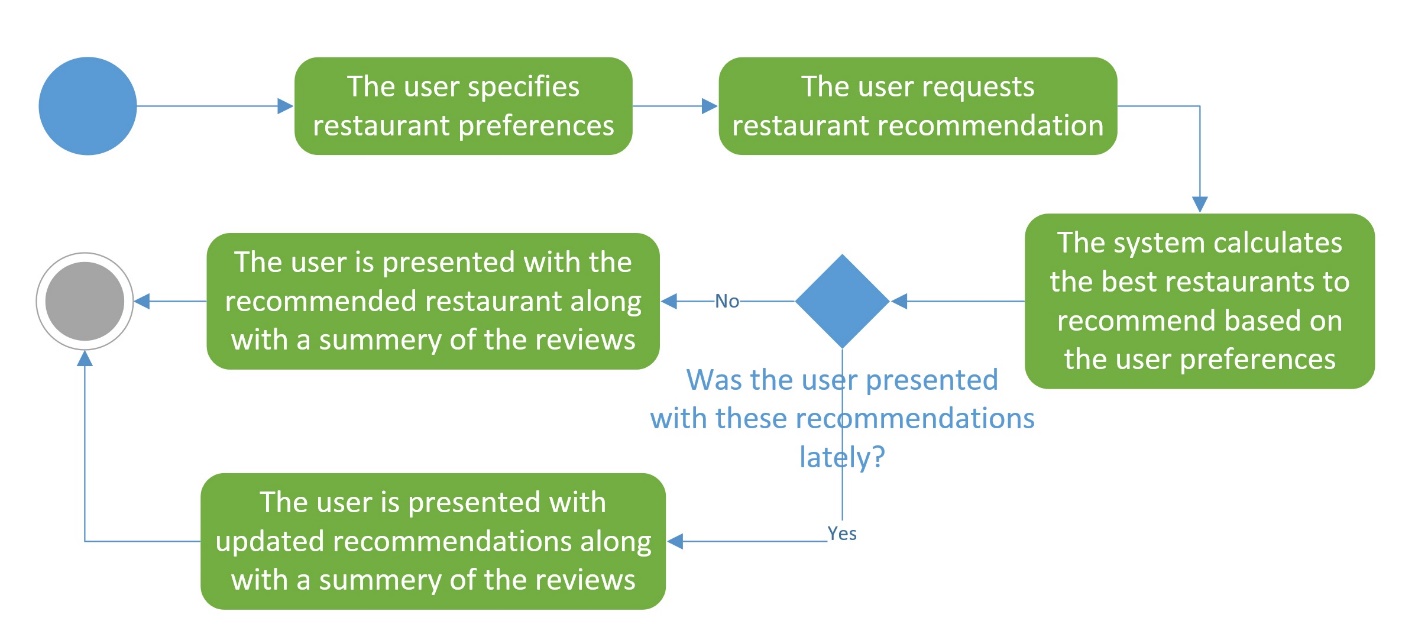
## Information Extraction/Retrieval Technology:

# Implementation Specification

## Software Architecture:

## UML Models:

## Workflow:



## Existing Services:

## New Services/Features Implemented:

# Project Management

## Contribution of Each Member:

|  |  |  |
| --- | --- | --- |
| Member | Contribution | |
| Samaa Gazzaz | Documentation:   * Motivation * Objective * Expected Outcome * Project Domain * Project Management * Workflow Diagram\* | Dataset collection:   * Collecting Static Datasets   Input/Output:   * Information Extraction/Retrieval Technology\*   + Code is included in the repository |
| Pooja Shekhar |  | |
| Chen Wang |  | |
| Dayu Wang |  | |

## Version Control/Screenshots:

For this project, we used GitHub as the main version control tool. The whole project, in addition to documentation, is up on: <https://github.com/SamaaG/WisdomRecSys>

## Concerns/Issues:

Although there are plenty of currently available recommendation systems that could serve as reference for this project, the time restrain imposes a huge concern on whether our system could be implemented on time. Since this is the first exposure for all our team members with recommendation systems, the time needed to get up to speed is going to take away from actual implementation time.

## Future Work:

For the next iteration, we plan to implement at least one feature for the system. In addition, preparing the dataset should be done by the next report.