**CS122 Project Proposal**

**Title: Take Me Away!**

**Objectives**

We want to develop a website that suggests affordable and well rated venues and attractions in the City of Chicago. This project is motivated by the need to include fast access to data on places that match your preferences and have a good price-quality relationship. Sometimes people will take a lot of time reading over reviews or end up going to places they don’t enjoy because information is spread out and overwhelming. We intend to make the choice easier by providing a tool that will select for you the top venues in a category and neighborhood of your choice.

From the front-end side, we will ask the user for information to refine the search criteria and return a map and a list with our main suggestions given the filters. The information we will be asking is:

* Neighborhood
* Type of establishment (restaurant, museum, shop)
* Key words to describe the sort of experience they want
* Target distance
* Time constraints

From the back-end side we will feed our website with information from YELP. We used YELP because it is the most comprehensive data source on venues with an available API (TripAdvisor did not allow us to access their API) We will connect to the YELP API and also scrape their website to obtain the complementary information the API does not provide.

We are aiming to obtain the following data:

* Name and location
* Type of establishment
* Price (YELP provides a $ scale to determine price, i.e. $, $$, $$$)\*
* Ratings (YELP provides a star scale to determine ratings)
* Reviews from users\*
* Hours of operations

\*Means scrapped data.

To prevent scrolling through the entire website we will use the search feature on YELP and crawl through the venues that match the criteria. Since the structure of a YELP search url follows the same pattern this task becomes easier to implement:

<http://www.yelp.com/search?find_loc=Hyde+Park,+Chicago,+IL&start=0&attrs=RestaurantsPriceRange2.1>

The previous link already has a structure that allows to crawl through the establishments in a neighborhood (find\_loc=Hyde+Park,+Chicago,+IL ) for a type of

establishment (attrs=Restaurants) with a given price range (PriceRange2.1) . This way we will make more efficient our crawling through the website.

This information will be used to do a matching with the key words the user provided and the constraints on time, distance and neighborhood. The matching will be done through regular expressions and the findall function.

**Innovative tools and databases**

The project will rely on previous concepts learned in class such as crawling through a website, data structures and regular expressions. However, we will be using 2 concepts not covered in class:

1. Connecting to the API of YELP and querying information
2. Using Google Maps to present information and obtain distance estimations.

If time permits, we will be interested in adding the following features (in order of priority):

1. Scrapping from other websites such as Timeout, Trip Advisor and tourist or restaurant blogs.
2. Recommend which neighborhoods to go based on a high price-quality concentration of establishments
3. Connect to the Foursquare of the user to obtain information on their preferences.

**Deliverables**

1. **Week 3 (Feb 9-12):** Database of venues, reviews, and events
2. **Week 5 (Feb 23-26):** Mapping function and natural language processing matching
3. **March 15:** Complete webpage

**Pendientes**

1. **Hacer función para conectarte al YELP api** 
   1. **Hacer lista de barrios**
   2. **Hacer lista de tipo de establecimiento**
   3. **Vincularse a información de establecimientos**
   4. **Return JSON file**

**Como pasar de un API a un crawler – desde el JSON vínculo al sitio web.**

1. **Crear crawler para YELP**
   1. **Name of restaurant**
   2. **Rating value**
   3. **Reviews**
   4. **Hours of operations**
   5. **Location**
   6. **Return CSV**

**Como generar una base que le sirva a SQL - usar csv en sql**

1. **Programar queries de SQL**
   1. **Hacer diccionario de palabras**
   2. **Sentiment analysis**
   3. **Matching de palabras**
   4. **Informacion geográfica**
   5. **Return query**
2. **Mostrar mapa con ubicación de lugares** 
   1. **Conectarse a google maps**
   2. **Mapear lugares con un código de colores especifico**
   3. **Filtrar por horario de operación**

**Resources**

[**http://engineeringblog.yelp.com/2013/10/yelp-dataset-challenge-winners-round-two-now-live.html**](http://engineeringblog.yelp.com/2013/10/yelp-dataset-challenge-winners-round-two-now-live.html)

[**https://www.yelp.com/dataset\_challenge**](https://www.yelp.com/dataset_challenge)

[**https://github.com/Yelp/dataset-examples**](https://github.com/Yelp/dataset-examples)

[**http://engineeringblog.yelp.com/2014/11/yelp-dataset-challenge-round-3-winners-and-dataset-tools-for-round-4.html**](http://engineeringblog.yelp.com/2014/11/yelp-dataset-challenge-round-3-winners-and-dataset-tools-for-round-4.html)

[**https://www.statwing.com/demos/yelp-dataset-challenge#workspaces/21456**](https://www.statwing.com/demos/yelp-dataset-challenge#workspaces/21456)