Project Proposal

A. Team

Fred Chang (fpc@seas.upenn.edu)
Lucy Kuo (yujuko@seas.upenn.edu)
Alicia Teo (alteo@wharton.upenn.edu)
Christina Chen (chenchri@pennlaw.upenn.edu)

B. Website Idea: Travel Helper

Overview

Our webapp is a travel helper that when given a list of landmarks a user wishes to visit, will provide hotel and dining recommendations that are nearby.

After the user selects a city (e.g. New York), the user will be able to create a list of sights they wish to visit (e.g. Times Square, WTC, Central Park), and our query will return Marriott hotels located within a user-selected distance from any or all the landmarks. The user can additionally select the price range ($-$$$$) of the Marriott hotels for their stay. After the user chooses a hotel, the app will then return recommendations for top restaurants to check out around the area which can also be filtered based on price range and ratings.

**** The order of any of these can be swapped, e.g. we can have users add landmarks AND restaurants to their wish-list first, then the hotel recommendations pop up, OR a user can pick a hotel of their choice and we can recommend landmarks and restaurants nearby based on popularity and distance rankings.*

General Flow of User Interaction:

Where do you want to travel? New York
↓
What landmarks do you want to see?
Data about city-specific landmarks will come from Foursquare. We will likely have a drop down listing the city's top landmarks. Some examples from the Foursquare data for New York are:

- Times Square
- Rockefeller Center
- Hudson Yards
- Statue of Liberty

↓

User can filter hotels based on distance from landmarks and price category.
Filter hotels located __**10**__ km of (any/**all**) landmarks
Show only hotels from category __1__ to category __10__ (price) .
↓
User will be shown hotel recommendations based on their filtering above:

- The Westin New York at Times Square
- The St. Regis New York
- Sheraton Brooklyn New York Hotel

↓
Based on user's choice of hotel, popular restaurants will be listed. They can be sorted by price and cuisine.

You have chosen: The St. Regis New York
Here are some popular restaurants around you:

Filter by: ($$$), Cuisine

- Clement ($$)
- The Polo Bar ($$$$)
- TAO Uptown ($$$)

Sort by: $, Avg Star Rating, Number of Reviews

C. Datasets

The three large data sets we will be using are from Foursquare, Marriott Hotels, and Yelp.

1. **Landmarks -** *Foursquare Landmarks (API)*
   We will get information on city landmarks from Foursquare via API requests. Once the user selects a city, we will send an API request to Foursquare to return the list of landmarks located within the city.

   Foursquare Places API: https://developer.foursquare.com/docs/api

2. **Hotels -** *Marriott (via Web Scraping)*
   We will obtain data from Marriott hotels from scraping hotel data from Marriott's website. This data will list all hotels, price category, and hotel address for hotels in the United States.

   List of Marriott properties: https://www.marriott.com/hotel-search.mi

Scraper:
https://chrome.google.com/webstore/detail/data-scraper-easy-web-scr/nndknepjnldbdbepjfgmncbggmopgden?hl=en-US

- **Size:** 5264 rows
- **Information Available:** Hotel Name, Hotel Category (Price Range), Address, Hotel Image URL, Avg Rating, Number of Reviews

Based on the address, we will also use the Google Geocoding API to get latitude / longitude information for each hotel to populate our database.
Google Geocoding API:
https://developers.google.com/maps/documentation/geocoding/intro

3. **Restaurants -** *Yelp (downloadable CSV)*
   Our restaurant data will come from a downloadable dataset provided by Yelp. Based on our review of the data so far, there are 174,000 businesses listed for 11 major cities.
   Kaggle Yelp Data: https://www.kaggle.com/yelp-dataset/yelp-dataset/download

   - **Size:** 8.62 GB
   - **Information Available:** Business Name, Address, Latitude, Longitude, Stars, Review_Count, Is_Open, Attributes, Categories, Hours

D. Queries

Here are some queries that we would write for our dataset:

1) **Find the hotels within Y (user selected) distance from the landmark(s) selected**
   For each of the landmarks selected in city X, calculate the distance between the landmark and the hotel using the lat-long information returned by the foursquare API, and the lat-long information in the hotel table.

   Return all the hotels in city X that are within Y km/miles from each of the landmarks, sort the hotels returned by distance to the landmarks (from lowest distance)

2) **Find restaurants within Y (user selected distance) from the landmark(s) selected**
   For each of the landmarks selected in city X, find the restaurants in nearby zip codes. For these restaurants, calculate the distance between the landmark and the restaurant using the lat-long information returned by the foursquare API, and the lat-long information in the Restaurants table.

   Return all the restaurants in city X that are within Y km/miles from each of the landmarks, sort the restaurants returned by distance to the landmarks (from lowest

distance)

3) **Find hotels within Y, filter hotels by price, and sort the hotels returned by rating (or # reviews)**
   For each of the landmarks selected in city X, calculate the distance between the landmark and the hotel using the lat-long information returned by the foursquare API, and the lat-long information in the hotel table.

   Return all the hotels in city X that are within Y km/miles from each of the landmarks, showing only hotels that below the price range selected by the user, and sort the hotels first by average rating (or # reviews), and then by distance to landmark.

4) **Show restaurants near selected hotel**
   For the hotel selected, filter the Restaurants dataset to restaurants in zipcodes that are near the hotel. For the filtered list of restaurants, calculate the distance between the hotel and the restaurants using the Lat-Long in Hotels and Lat-Long in Restaurants.

5) **For the selected hotel, show the top 5 cuisine types of nearby restaurants**
   For the selected hotel, find the Restaurants nearby (using same logic as #4). After getting the list of nearby restaurants, group by Cuisine, find the number of restaurants in each Cuisine type, and return the top 5 counts.

   E.g. if the user selects Four Seasons Hotel New York, he/she will see:
   - Japanese (15 Restaurants)
   - American (10 Restaurants)
   - Italian (10 Restaurants)
   - French (5 Restaurants)
   - Thai (3 Restaurants)

6) **Filter restaurants based on selected price range or cuisine type. Sort restaurants based on average ratings**
   For the selected hotel, find the restaurants nearby (same logic as #4). Filter the list to only restaurants within the specified range and/or cuisine type. Return the list sorted by average ratings in descending order.