

# CIS550 Final Project Report

Erhan Hu, Jiali Han, Yanwen Jin

## 1 Abstract

The basic motivation for this application is to serve as a travel planner focusing on restaurants and bars, around those seven American cities in the provided Yelp dataset. In our final implementation, major functionality is to search by zip code with preferences on different categories. Our system will return the highest-rated businesses based on user preference, providing business name, corresponding url link, business image, rating stars, and with relevant magic verbal description from Bing search.

This application also supports member registration and login, once you have logged in as a member, it provides more magic features. For instance, as a member, you can add your favorite business entity from results, and after you come back to homepage, you would be able to see all your added businesses labeled in real Google maps.

## 2 Modules and architecture

For the backend part, we used MySQL as our database format hosted it on Amazon RDS. We picked only a subset of tables and columns, they are: Business (including id, name, location and rating) and Tips (including business id and tips), and Categories(business-id and category keywords).

For front-end we chose to use Python and Django web framework, along with Bootstrap UI. Also we added a few of public APIs to retrieve more useful information, including Yelp API, Bing Search and Google Map.

## 3 Data instance use

In our project, we basically uses three database tables to provide information for our application, including: Business, Categories, Tips.

Business table has attributes: id, name, city, state, zip code, latitude, longitude, stars, which is the most important part in our application, since our basic functionality is user searching by zip code and our system return back information of business name, city, state and rating stars. In addition, it is also worth mentioning that latitude and longitude attributes helps us a lot to add color to our searching system. We create an interesting feature that results will also provide nearby businesses within a range of area constrained by longitude and latitude.

Category table contains attributes: business-id and category, which makes it easy to search by category, such as restaurant, shops, bars, etc. It improves our searching functionality by allowing users searching efficiently and conveniently based on their preferences on different category business types.

Tips table has attributes: business-id and tip, which could join in business table, and provide tips information such as user comments and recommendations for each business entity. These tip information will be presented with our search result, thus, it is also necessary to make our search result more comprehensive.

## 4 Data cleaning and import mechanism

We sign up an AWS account for our project and create our own database instance on RDS, then we imported the data from the original yelp JSON file to form our own 4 table and populate to our database by using python. The data for the Business table (involves id, name, street-address, city, state, zip code, latitude, longitude, stars) and Hours table (involves business-id and each day in a week) are retrieved from the original business table based on the requiring elements in each business entry. And the Category table which involves business-id and category is formed by getting the business-id from the original business table and its corresponding category keyword. The tips table (involves business-id, tips and likes) are formed by selecting element from each entry in the original tips JSON file. The last user table are created by ourself, that involves user-id, friend-id and First-name, Last-name and Plan-text. The user table is used to store information for the user registration on our website.

## 5 Use cases

### 5.1 About page

When you clicked the “About” button, it direct you to the about page that gives a brief introduction of our website. Our main usage is to search all kinds of business based on

your zip code. If you are not satisfied with the results find in your place, we provide a nearby suggestion zip code lists that are close to your area. If you have further questions, we provide the contact methods that displayed at the bottom of the page.

## 5.2 Home page

When you clicked the “Home” button, it will direct you to our main page. Here, we show a list of popular cities that have the most number of businesses based on the yelp database. If you click “Las Vegas”, it will show you the results directly in another page.



Search

**These areas are popular :**

Phoenix, AZ 85050	21927 places to visit
Las Vegas, NV 89102	14435 places to visit
Charlotte, NC 28211	4368 places to visit
Pittsburgh, PA 15234	2670 places to visit
Madison, WI 53703	1989 places to visit
Urbana, IL 61801	532 places to visit
Fort Mill, SC 29715	157 places to visit

Figure 1: Popular cities displayed in the Home page

## 5.3 Zip code search

The most basic functionality is searching for businesses given a zip code and desired category which can be restaurants, bars and hotel. You can choose “Search for all categories” in the checkbox to return all kinds of businesses. When you click the “search” button in the home page, it returns the highest-rated businesses based on your preference. If you want to get further information for a specific restaurants, just click the name of it, and it will direct you to its yelp page In addition to what is in the database we also retrieve additional stuff from public APIs. We used Yelp API to get an image for every returned business in order to provide a first impression for businesses so users can make better and easier choices for their plans. Also Bing API is used to provide a short verbal description for each business. Besides, It’s worth to mention that at the bottom of the page, we give a list of zip codes that are close enough to the zip code you initially entered. This is to avoid the case when we might miss some businesses that are actually very close to us but have different zip codes. The picture below shows the results of zip code 89104 and its corresponding nearby zip code list.



Restaurants found at 89109 (Las Vegas)			
Name	Picture	Rating	Words from Bing
<a href="#">The Capital Grille</a>		4.5	346 Reviews of The Capital Grille "Awesome! Pretty much the word "awesome" sums up our entire dining experience! Hubby and I came here for our anniversary dinner.
<a href="#">Project Pie</a>		4.5	Project Pie is the next generation; the culmination of James' experience and learnings. ... Project Pie MGM 3799 South Las Vegas Blvd Las Vegas, NV 89109 (702) 463-6200

Figure 2: Restaurants found near zip code 89109

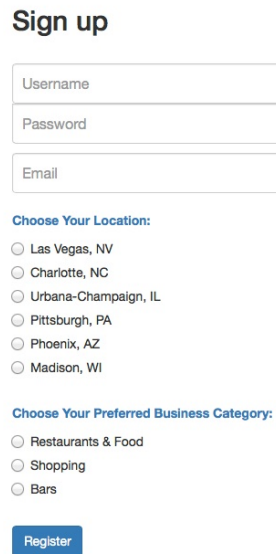
#### Also try these nearby zipcodes:

[89102 \(Las Vegas, NV\)](#)  
[89119 \(Las Vegas, NV\)](#)  
[89103 \(Las Vegas, NV\)](#)  
[89169 \(Las Vegas, NV\)](#)  
[89120 \(Las Vegas, NV\)](#)  
[89118 \(Las Vegas, NV\)](#)  
[89139 \(Las Vegas, NV\)](#)  
[89109 \(Las Vegas, NV\)](#)  
[89104 \(Las Vegas, NV\)](#)  
[89146 \(Las Vegas, NV\)](#)  
[89121 \(Las Vegas, NV\)](#)  
[89074 \(Henderson, NV\)](#)  
[89101 \(Las Vegas, NV\)](#)  
[89108 \(Las Vegas, NV\)](#)  
[89106 \(Las Vegas, NV\)](#)

Figure 3: Nearby zipcode lists

## 5.4 Registered as User

Our website is usable for people who have their own accounts. The first thing you need to do is click the “Register” button on the Home page, and fulfill your personal information in its corresponding area. And your location and preferred business type in the checkboxes, this is used to give the user a more precise result. The picture below displays the sign up page.



**Sign up**

Username

Password

Email

**Choose Your Location:**

- ☐ Las Vegas, NV
- ☐ Charlotte, NC
- ☐ Urbana-Champaign, IL
- ☐ Pittsburgh, PA
- ☐ Phoenix, AZ
- ☐ Madison, WI

**Choose Your Preferred Business Category:**

- ☐ Restaurants & Food
- ☐ Shopping
- ☐ Bars

**Register**

Figure 4:

After you sign up, you can login in your account by clicking the “Login” button on the home page, then it will show you the home page that based on your registered location and present the most popular business places that are close to your location. The fancy thing for user is that, after your search for a business type, there is a column called “Add” on the right. If you make a plan for your trip and want to select this kind of business, just click the “Add”, and then on your own profile page, a list of your selection will display under the “Plan”. The picture below shows an example of plan on a user profile page.

# Plan

## The Capital Grille

- All I can say is...lobster mac and cheese
- The best Calamari in Las Vegas
- Steak tartar is killer and an amazing wine list
- 2014 Generous Pour Event
- Way better service than SW restaurant in Wynn.

## Earl of Sandwich

- Best sandwiches eveeeeeer
- Anxious to see what all the hypes about...
- Amazing. Best sandwich place you will ever go.
- Best sandwiches hands down.
- Cheapest quick lunch of decent quality on the strip.

Figure 5: An example for user plan

## 6 Optimization techniques employed

We implemented indexing on Latitude and Longitude in the Business table. The motivation of doing this is that one of the major functionality of our web application is to search a zip code's nearby zip codes. In real world, it is very possible that two very close zip codes are actually numerically very far from each other, the effect is that two businesses may have different zip codes even if they are steps away from each other. So in order not to miss any businesses that are close enough, we searched for nearby zip codes when a user is search a particular zip code. This is done by comparing latitude/longitude value and returning all zip codes whose latitude/longitude values are close enough to the given zip code's. Since a comparison must be executed for every latitude/longitude value in the Business table, it might take considerable time. So we added a indexing on the latitude/longitude so the comparison can run faster. Another optimization we did is multi-threading. We found that API calls take too much time due to lengthy authorizations and url visitings. The "search results" page usually contains 15 results and each of them requires an API call and takes almost a minute to display. Therefore we implemented multi-threading in Python so that the 15 API calls start together. This enabled each page to return in less than three seconds.

## 7 Technical specifications

In this project we employed Python programming language. For the backend database part we used MySQL for it is better supported in both Ubuntu and Mac OS X platform. The Python packages/libraries we used include Django web framework (a Python library for website/server construction). Also we used APIs including Yelp API, Bing Search API from Azure marketplace and Google Map API. It is worth noting that in order to run this program locally, Django needs to be installed first, along with other small Python packages including oauth2, django-bootstrap-toolkit and MySQL connector for Python. When the project starts running, it first generates relevant SQL queries in format of python strings then connects to the MySQL database instance hosted on RDS. After query results returned, it transforms the results into Python arrays and display them on the html webpages.

## 8 Special Features

As mentioned in the above chapter in this project we embedded Yelp API, Bing Search API and Google Map API. Yelp API is used to return a business picture to give user a more intuitive way to view and compare businesses and thus make better choices. We tried to return multiple pictures for each business to give more informative description but found that Yelp API only supports one picture per business currently. Therefore in order to provide more information we embedded Bing Search API into our program to get the first search result returned as a companion to the image return by Yelp API. After adding these two APIs we found that they provide relevant description in the most cases to help users make choices. Lastly Google Map API is embedded as part of JavaScript code. A map with all selected businesses as markers will be displayed when a user wants to view his/her travel plan. It is intended to provide a more visualized way for users to see where his/her destinations will be located.

## 9 Division of labors

Yanwen Jin took part in the design of functionalities and use cases.

Jiali Han took part in UI design and HTML.

Erhan Hu took part in the coding.